Consultative Committee for Space Data Systems

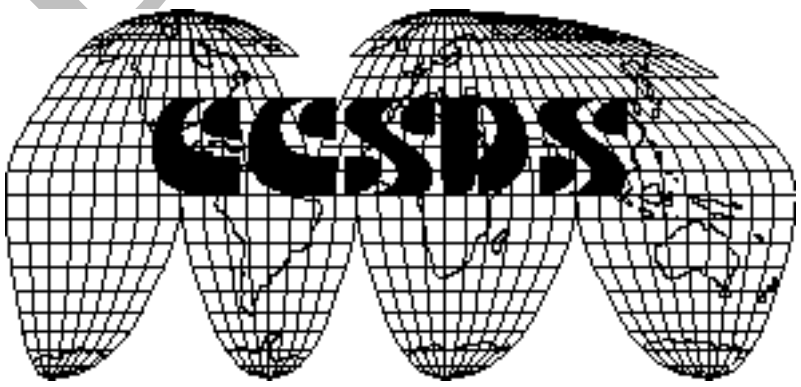
RECOMMENDATION FOR SPACE DATA SYSTEM STANDARDS

Monitored Data Cross Support Transfer Service

CCSDS 922.1-W-0.24

White Book

May 2014



NOTE TO READERS

This preview draft is being made available to assist reviewers of the *Cross Support Transfer Service – Specification Framework* Draft Recommended Standard (Red Book, CCSDS 921.1-R-2) in reviewing that document. Since the Monitored Data CSTS is built using the CSTS Specification Framework, this MD CSTS specification provides a full example of the use of the CSTS Specification Framework to develop a CSTS specification.

This MD-CSTS specification has been submitted to the CCSDS Secretariat for the first Red Book review by CCSDS Agencies. The Secretariat will correct any typographical or formatting errors that may exist in this preview draft prior to releasing the document for Agency review. The Agency review of the Issue 1 MD-CSTS Red Book is expected to occur in late summer 2014.

This note will not appear in the version of the draft specification that is made available for Agency review.

AUTHORITY

Issue:	Red Book, Issue 1
Date:	May 2014
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 the *Procedures Manual for the Consultative Committee for Space Data Systems*, and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This document is published and maintained by:

CCSDS Secretariat
Space Communications and Navigation Office, 7L70
Space Operations Mission Directorate
NASA Headquarters
Washington, DC 20546-0001, USA

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:

- 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.
- 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.
- 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

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS shall not be held responsible for identifying any or all such patent rights.

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 *Procedures Manual 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 addressed to the CCSDS Secretariat at the 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 e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- Russian Federal Space Agency (RFSA)/Russian Federation.
- 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.
- CSIR Satellite Applications Centre (CSIR)/Republic of South Africa.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- 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.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- 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.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.

– United States Geological Survey (USGS)/USA.

CONTENTS

Table Page

TBS.

1 INTRODUCTION

1.1 PURPOSE OF THIS RECOMMENDATION

This Recommendation defines the Monitored Data Cross Support Transfer Service (CSTS), in conformance with the Cross Support Transfer Services Specification Framework Recommended Standard [1]. The Monitored Data CSTS (MD-CSTS) is a service that allows a spaceflight mission to receive cyclic reports on, and to query the current values of, the parameters that are pertinent to cross support services being provided by a Cross Support Complex. The Monitored Data service also allows a spaceflight mission to receive notifications of the occurrence of events of interest associated with the services that are being provided by a Cross Support Complex.

1.2 SCOPE

This Recommendation defines the Monitored Data service in terms of:

a) the CSTS procedures that comprise the service;

b) the extensions and refinements of the behavior of those CSTS procedure necessary to provide the transfer service;

c) the extensions and refinements of standard CSTS operations associated with each of the procedures;

d) the relationships among the procedures that comprise the service.

It does not specify:

a) individual implementations or products;

b) the implementation of entities or interfaces within real systems;

c) the methods or technologies required to measure the values of monitored parameters and to detect the occurrence of events of interest;

d) the methods or technologies required for communication;

e) the management activities necessary to schedule, configure, and control the Monitored Data service;

f) the specific parameters that are to be reported and events that are to be notified by the MD-CSTS.

1.3 APPLICABILITY

The applicability and limits of applicability of Cross Support Transfer Services in general, as described in [1], pertain to the Monitored Data service, with the addition of the conditions described in section 1.3.1, below.

1.3.1 APPLICABILITY OF THIS RECOMMENDED STANDARD

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.

1.4 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.

1.5 DOCUMENT STRUCTURE

1.5.1 DOCUMENT ORGANIZATION

Section 2 describes the Monitored Data Cross Support Transfer service in terms of:

- the role of Service Management with respect to the MD-CSTS;
- the allocation of production and provision of the MD-CSTS to functional resources;
- the cross support view of the MD-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 composition of the MD-CSTS. The service type identifier is declared, the procedures that comprise the service are identified, and the CSTS state machine that applies to the MD-CSTS is specified. Because the MD-CSTS is composed of procedures that are directly adopted from the CSTS Framework without extension, no further specification of the MD-CSTS is required.

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

Section 5 defines the monitored parameters and notifiable events that report the status of the MD-CSTS itself.

ANNEX A provides the formal specification of the object identifiers for the Monitored Data transfer service.

ANNEX B is the Implementation Conformance Statement Proforma for the MD-CSTS.

ANNEX C defines the monitored data production process. In particular, it specifies how monitored data values are to be labelled so that, when transferred by MD-CSTS instances, the sources of the measurements are unambiguous.

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

ANNEX E describes an example set of monitored parameters, notifiable events, and their associated Functional Resource types. These examples are taken from the SANA registry (which is the normative repository of all such definitions) at the time of publication of this Recommended Standard.

ANNEX F provides the list of informative references.

ANNEX G lists the acronyms used in this document.

1.5.2 CROSS SUPPORT TRANSFER SERVICES DOCUMENTATION

The basic organization of the Cross Support Services documentation and the relationship to the CSTS documentation is shown in Figure 1-1.

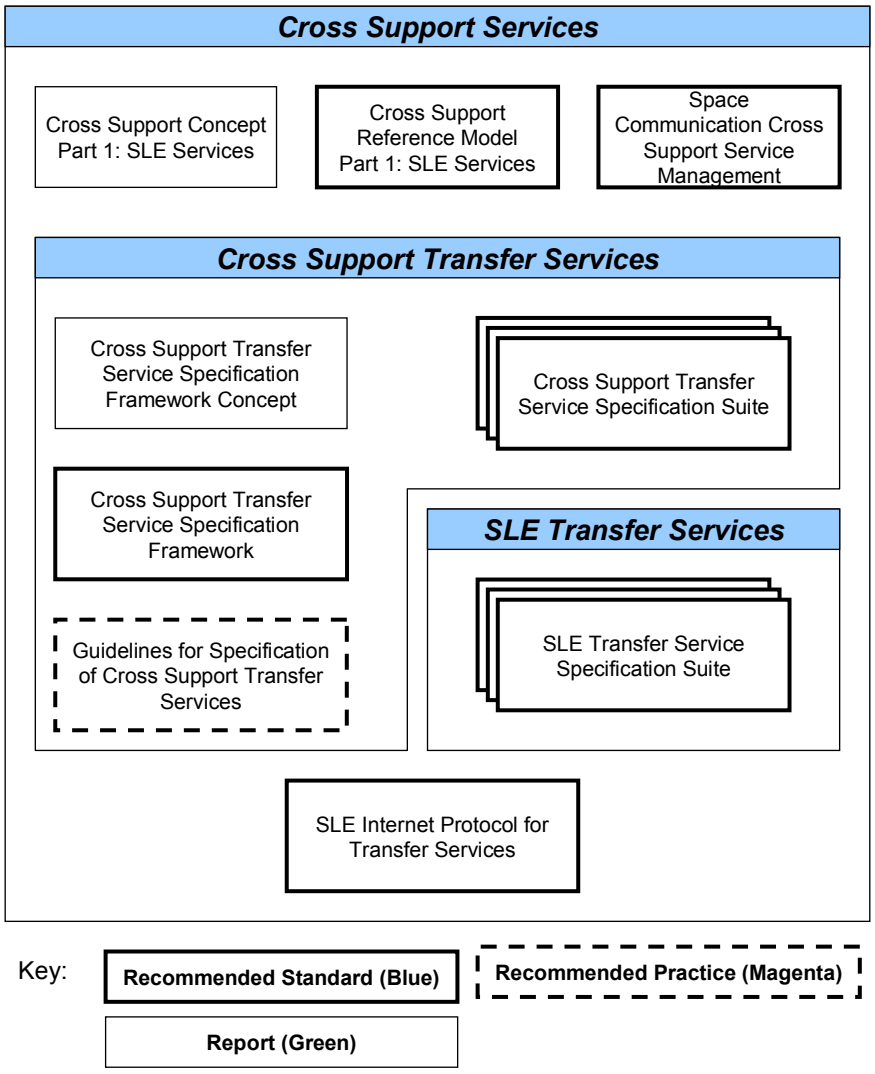


Figure 1-1: Cross Support Service Documentation

The CSTS Documentation is:

a) *Cross Support Concept—Part 1: Space Link Extension Services* (reference [E1] a report introducing the concepts of cross support and the Space Link Extension (SLE) services. Many of the concepts for the SLE transfer services have been adopted for the CSTSes as explained in section 2;

b) *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.

c) *Space Communication Cross Support - Service Management – Service Specification* (informative reference [E6]): a Recommended Standard that specifies the

management framework and services through which CSTSes are configured and scheduled;

d) The SLE Transfer Services suite: The SLE Transfer Services comprise a set of cross support transfer services that are used to transfer specific telecommand and telemetry protocol data units. 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 Framework. However, due to history (the SLE Transfer Services were already specified and implemented prior to development of the CSTS Framework) the SLE Transfer Services are separated from CSTSes;

e) *Space Link Extension - Internet Protocol for Transfer Services* (reference [E2]) a Recommended Standard that defines a protocol for transfer of Protocol Data Units (PDU) defined in the Cross Support Transfer Services. This Recommended Standard was originally developed to support SLE transfer services (hence the title), but it is also applicable to (and specified for) use by cross support transfer services.

The documents specific to Cross Support Transfer Services are:

f) *Cross Support Transfer Services Specification Framework* (reference [1]): a Recommended Standard that defines basic building blocks for the specification of Cross Support Transfer Services;

g) *Guideline for Specification of Cross-Support Transfer Services*: a Recommended Practice that, when published, will define the guidelines for construction of a Cross Support Transfer Service based on the CSTS Specification Framework;

NOTE - As of the publication of this Recommended Standard, the Guidelines Magenta Book is in-progress. It is not required to understand this Recommended Standard.

h) *Cross Support Transfer Services Specification Framework Concepts* (reference [E3]): A Report that provides tutorial material on the objectives and concepts of the CSTS Specification Framework;

i) Cross Support Transfer Services Specification Suite: The set of specifications for actual CSTSes built from the procedures in the CSTS Framework and in accordance with the CSTS Guidelines.

1.5.2.1 Terms Defined in the Cross Support Transfer Services Specification Framework (CSTS SFW) [1]

a) Association Control procedure;

b) Buffered Data Delivery procedure;

c) complete data delivery mode;

d) cross support service production;

e) cross support transfer service;

f) Event Identifier;

g) Event Label;

h) Event Name;

i) Functional Resource Instance;

j) Functional Resource Instance Number;

k) Functional Resource Name;

l) Functional Resource Type;

m) Label List;

n) Label List Set;

o) Parameter Identifier;

p) Parameter Label;

q) Parameter Name;

r) prime procedure instance;

s) Procedure Type;

t) Procedure Instance Identifier;

u) publish;

v) Published Identifier;

w) qualified parameter;

x) secondary procedure instance;

y) service-user-responding-timer;

z) subscription.

1.5.2.2 Terms Defined in the Cross Support Reference Model [2]

a) Complex Management (CM) (called *SLE Complex Management* in [2]);

b) Mission User Entity (MUE);

c) Service Agreement (called *SLE Service Agreement* in [2]);

d) Service Package (called *SLE Service Package* in [2]);

e) Space link session;

f) Transfer service production;

g) Transfer service provision;

h) Utilization Management (UM);

i) Utilization phase (called SLE Service Package Utilization phase in [2]).

1.5.3 NOMENCLATURE

The following conventions apply throughout this Recommendation:

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

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

c) the word ‘may’ implies an optional specification;

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

1.5.4 CONVENTIONS

1.5.4.1 The conventions defined in the CSTS Specification Framework (CSTS SFW) Recommended Standard (reference [1]) are applicable to this Monitored Data service specification.

1.5.4.2 Object Identifier Representation

The MD 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 the CSTS SFW (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.

1.5.4.2.1 Functional Resource Type. As specified in reference [1], a Functional Resource Type is a Published Identifier (i.e., an ISO OID). In the descriptions and examples in this Recommended Standard, a Functional Resource Type is represented using the notation *{published identifier descriptor}*, which is a textual description of the Published Identifier. Thus {Antenna} represents the Published Identifier for the Antenna Functional Resource Type.

1.5.4.2.2 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 descriptor}*, which is a textual description of the Object Identifier. Thus {Cyclic Report} represents the Object Identifier for the Cyclic Report Procedure Type.

1.5.4.2.3 Functional Resource Name. As specified in reference [1], a Functional Resource Name is composed of a Functional Resource Type published identifier and an integer Functional Resource Instance Number (FRIN). In the descriptions and examples in this Recommended Standard, a Functional Resource Name is represented using the notation *[{functional resource type published identifier descriptor}:FRIN]*. [{Antenna}:1] represents name of the Antenna Functional Resource Type that is assigned FRIN = 1 in the scheduled Service Package.

1.5.4.2.4 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 descriptor}:'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 it 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 descriptor}:'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 descriptor}:SPIN]*. [{Notification}:1] represents the Procedure Instance Identifier of the Notification instance that serves as the first Notification secondary procedure instance a CSTS.

1.5.4.2.5 Parameter Name. As specified in reference [1], a Parameter Name can be either the name of a Functional Resource parameter or the name of a CSTS procedure parameter. A

Functional Resource parameter name is composed of (1) the Functional Resource Name of the Functional Resource Instance that reports the parameter, and (2) the Parameter Identifier for the parameter. In the descriptions and examples in this Recommended Standard, a Parameter Name is represented using the notation *[[{functional resource type published identifier descriptor}:FRIN] : {parameter published identifier descriptor}]*. Thus [[{Antenna}:1] : {`actualAzimuth`}] represents the actual-azimuth parameter of the Antenna Functional Resource Type that is assigned FRIN 1 in the scheduled Service Package.

A CSTS procedure parameter name is composed of (1) the Procedure Instance Identifier of the procedure instance that reports the parameter, and (2) the Parameter Identifier for the parameter. In the descriptions and examples in this Recommended Standard, a CSTS parameter name is represented using the notation *[[{procedure type object identifier descriptor}:SPIN] : {parameter published identifier descriptor}]*. Thus [[{Cyclic Report}:'prime procedure instance'] : {`pCRlabelLists`}] represents the label lists parameter of the prime instance of the Cyclic Report procedure of the CSTS.

1.5.4.2.6 Event Name. As specified in reference [1], an Event Name can be either the name of a Functional Resource event or the name of a CSTS procedure event. A Functional Resource event name is composed of (1) the Functional Resource Name of the Functional Resource Instance that reports the event, and (2) the Event Identifier for the event. In the descriptions and examples in this Recommended Standard, an Event Name is represented using the notation *[[{functional resource type published identifier descriptor}:FRIN] : {event published identifier descriptor}]*.

NOTE - As of the publication of this Recommended Standard, there are no procedure events for the three procedures used in the Monitored Data service (Cyclic Report, Information Query, and Notification).

1.5.4.2.7 Parameter Label. As specified in reference [1], a Parameter Label is composed of (1) the Functional Resource Type that reports the parameter and (2) the Parameter Identifier for the parameter. In the descriptions and examples in this Recommended Standard, a Parameter Label is represented using the notation *[{functional resource type published identifier descriptor}: {parameter published identifier descriptor}]*. Thus [{Antenna}:{`actualAzimuth`}] represents the actual-azimuth parameter of the Antenna Functional Resource Type.

1.5.4.2.8 Event Label. As specified in reference [1], an Event Label is composed of (1) the Functional Resource Type that reports the parameter and (2) the Event Identifier for the event. In the descriptions and examples in this Recommended Standard, an EventLabel is represented using the notation *[{functional resource type published identifier descriptor}: {event published identifier descriptor}]*.

1.5.4.2.9 Label List. As specified in reference [1], a Label List is a data structure that specifies the name of a list of Parameter Labels or Events Labels, indicates if the list is the default list, and contains all Parameter Labels or Event Labels represented by that Label List name. In the descriptions and examples in this Recommended Standard, a Label List is represented using the notation:

{ labelListName;
defaultFlag;
<Parameter Label or Event Label>;
.
.
.
<Parameter Label or Event Label>
}

where *<Parameter Label or Event Label>* is as described in 1.5.4.2.7 or 1.5.4.2.8.

1.5.4.2.10 Label List Set. As specified in reference [1], a Label List Set is the set of Label Lists accessible by the user of the given service instance. In the descriptions and examples in this Recommended Standard, a Label List Set is represented using the notation:

{ <Label List>;
.
.
.
<Label List>
}

where *<Label List>* is as described in 1.5.4.2.9.

1.6 REFERENCES

[1] *Cross Support Transfer Service - Specification Framework*. Draft Recommended Standard, CCSDS 921.1-R-2. Red Book. XXX 2014.

[2] *Cross Support Reference Model, Part 1: Space Link Extension*. Recommended Standard, CCSDS 910.4-B-2. Blue Book. October 2005.

2 OVERVIEW OF THE MONITORED DATA SERVICE

2.1 SERVICE SUMMARY

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

a) Obtain cyclic reports of the values of monitored parameters of interest during the MD-CSTS service instance provision period, which occurs within the utilization phase of a cross support service package;

b) Obtain notification of the occurrence of events of interest during the MD-CSTS service instance provision period;

c) Query the current values of monitored parameters of interest during the MD-CSTS service instance provision period.

An instance of the Monitored Data service is realized through the following activities:

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

b) The user selects which monitored parameters are to be reported cyclically and at which reporting periods, and starts (enables) the reporting of those parameters;

c) The user selects which events are to be notified and enables the transmission of those event notifications;

d) The service provider reports the selected monitored parameter values at the specified periodicity;

e) The service provider transmits the selected notifications upon occurrence of their associated events;

f) The user queries the current values of monitored parameters;

g) The user stops (disables) periodic reporting and event notification; and

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

The Monitored Data service delivers only current parameter values: there is no capability in this service to deliver parameter values stored from previous times (that is, there is no “complete” mode).

2.2 FUNCTIONAL DESCRIPTION

As defined in the Cross Support Reference Model: Part 1 (reference [2]), related cross support services are bundled into Service Packages for the purposes of ensuring that the required relationships among those cross support services are preserved during their production and provision. For example, multiple cross support transfer services may be related to the operation of the same radio frequency (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 defined in reference [E6].

A Service Package may contain multiple instances of the Monitored Data service. Each instance of the Monitored Data service in a Service Package is capable of reporting all monitored parameters and event notifications 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 Monitored Data service, the Monitored Data service instance has access to the monitored parameters and event notifications for those RF links and SLE transfer service instances (in addition to the Monitored Data service instance's own monitored parameters and event notifications).

In accordance with reference [2], 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 Monitored Data service, respectively.

2.2.1 SERVICE PRODUCTION

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

- instances of Functional Resource types that realize that Service Package; and
- a Data Collection function that collects the monitored parameter measurements and event notifications emitted by those Functional Resource instances in order to make them available to the Monitored Data service instances.

Figure 2-1 is a notional representation of the relationship between the functional resource instances of a Service Package and instances of monitored data service that report on that Service Package.

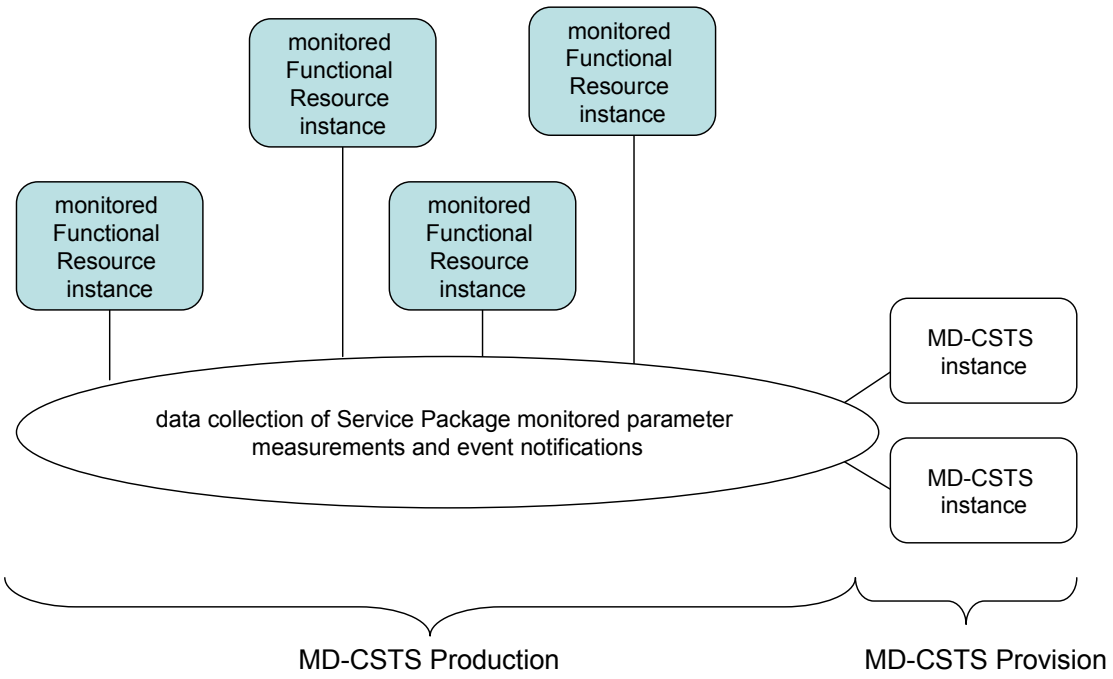


Figure 2-1: Production and Provision of Monitored Data Services (Notional)

As defined in [1], a *Functional Resource Type* is a logical function or related set of functions that provides space communication capability. A Functional Resource Type may have configuration parameters, monitored parameters, notifiable events, and/or directives associated with it. A *monitored* Functional Resource type is a Functional Resource type for which monitored parameters and/or notifiable events are specified. For the purposes of the Monitored Data CSTS, only monitored Functional Resource types are of interest.

In general, a Functional Resource type may be instantiated multiple times in the same Service Package, and therefore it is necessary to distinguish the individual instances of those Functional resource types, known as *Functional Resource instances*.

A standard common set of space communication Functional Resource types and their associated monitored parameters, notifiable events, and directives have been defined by CCSDS and registered with SANA in accordance with the process specified in the CSTS SFW [1]. Annex C identifies an example set of Functional Resource types and the monitored parameters and notifiable events associated with those Functional Resource types. These example types, parameters, and events are consistent with the SANA registry at the time of publication of this Recommended Standard, but may change over time without Annex C being updated. Annex C should therefore be considered informative, and implementers of the MD-CSTS must use the SANA registry as the normative source for definitions of the applicable Functional Resource types, parameters, and events.

NOTES

1 The operational scenario in section 2.5 provides hypothetical examples of functional resource types.

2 Cross Support Complexes that provide the MD-CSTS will be expected to report as many of the standard monitored parameters and notifiable events as possible. Individual Cross Support Complexes may also use the MD-CSTS to report non-standard monitored parameters and event notifications that are available from those Complexes.

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) and even multiple instances of the same functional resource type(s) (e.g., a symbol stream associated with an X-Band return space link and a symbol stream associated with an S-Band return space link). Conceptually, instrumentation of the functional resources produces the raw measurements that are the basis for the monitored parameter values and the event notifications associated with those functional resources.

The MD-CSTS is not constrained to operate with any particular set of functional resource types, but rather can report the monitored parameter values and event notifications for any set of functional resources as long as the monitored data and event notifications emitted by those functional resources are defined in the SANA registry.

Functional resources may occur in different combinations and multiplicities within a Service Package. For example, a single antenna may be used by both forward and return space links, and more than one return space links may use the same antenna. If quadrature phase shift key (QPSK) modulation is used on the carrier, there may be two symbol streams multiplexed onto the single carrier. It is therefore possible that a single Service Package may have multiple instances of the same types of monitored parameters or notifiable events associated with the different instances of the functional resources (e..g., bit synchronizer lock status for a Service Package that encompasses multiple return symbol streams).

To provide unambiguous identification of each instance of a monitored parameter or notifiable event within the context of the Service Package, the MD-CSTS uses the Parameter Name of each monitored parameter that is reported and the Event Name of each event that is notified. The Parameter Name of a parameter represents both (a) the type of monitored parameter (e.g., `carrierFrequency`) and (b) the name of the Functional Resource instance with which the parameter is associated. Similarly, the Event Name of a notifiable event represents both (a) the type of the notifiable event (e.g., ‘bit sync lock lost’) and (b) the name of the Functional Resource instance with which the event is associated.

As specified in the CSTS Specification Framework (reference [1]):

a) the monitored parameter type component of the Parameter Name is called the Parameter Identifier and has the syntax of a Published Identifier, that is, a special type of Object Identifier on the CCSDS Registration Tree;

b) the notifiable event type component of the Event Name is called the Event Identifier and has the syntax of a Published Identifier;

c) the name of the Functional Resource instance has the syntax of a Functional Resource Name which has two components:

   1) a Functional Resource Type, which has the syntax of a Published Identifier, and

   2) a Functional Resource Instance Number, which is a positive integer.

NOTES

3 The Functional Resource Instance Number is used to differentiate between multiple instances of the same Functional Resource Type that are configured in the same Service Package. Whereas the Parameter Identifiers, Event Identifiers, and Functional Resource Types are all statically defined, the assignment of Functional Resource Instance Numbers is a dynamic result of the scheduling of Service Packages, which may contain different combinations and multiplicities of Functional Resource Types.

4 The abstract syntaxes of the Parameter Name, Event Name, Functional Resource Type, Functional Resource Instance Number, and Functional Resource Name are formally specified in reference [1].

The Data Collection function of MD-CSTS production makes all of the monitored parameters and event notifications available to all instances of MD-CSTS for the Service Package in conformance with the naming, syntax, type, and units defined in the specification for those monitored parameters and event notifications. In cases where the real implementations of functional resources do not present their monitored data measurements and/or event notifications in conformance with the naming, syntax, and/or units required for transfer via the MD-CSTS, the Data Collection function performs the appropriate format conversion.

During the execution of a Service Package, the production collectively supplies two kinds of data for use in the provision of MD-CSTS instances:

a) The current values of all instances of functional resource monitored parameters that are collected and made available by the Complex for either cyclic reporting to, or query by, that spaceflight mission; and

b) Functional resource event notifications for all notifiable events that are made available by the Complex for that spaceflight mission.

2.2.2 SERVICE PROVISION

Multiple instances of the MD-CSTS may exist in the same Service Package, with each instance providing selected subsets of the functional resource monitored parameter values and event notifications that have been aggregated from the monitored functional resources.

Each instance of the Monitored Data service allows the user to subscribe to any of the sets of monitored parameters supported by the Complex and have that set of parameters be cyclically reported via that service instance. Optionally, each instance of the Monitored Data service allows the user to query the current value of any monitored parameter. Also optionally, each instance of the Monitored Data service allows the user to subscribe to any of the supported set of notifiable events to have those specific events to be notified via that MD-CSTS instance.

NOTE - The capabilities of real implementations of Functional Resources or the Data Collection function of the production affect the freshness and/or accuracy of the data being reported. For example, a particular piece of equipment may output a new measurement value only every 5 seconds. An implementation of the data collection function might hold this value as a constant in the database for the duration of the 5 seconds. In this case, a resulting query that is invoked half-way between samples would return the 2.5-second-old value as the "current" value. Agreements between spaceflight mission and cross support complexes should specify any such timing relationships.

2.2.2.1 Cyclic Reporting of Parameters

The MD-CSTS Cyclic Report procedure (see section 4) allows the service user to subscribe to a *list of parameters* to be periodically reported and the *delivery cycle* (period) for those reports in the START invocation message that activates the Cyclic Report procedure. Multiple instances of the Cyclic Report procedure can be active concurrently, each with a different delivery cycle, which allows the service user to fine tune the reporting based on relative importance and expected rate of change in the values of the monitored parameters.

The Cyclic Report procedure is the prime procedure for the MD service. That is, at least one instance of the Cyclic Report procedure must be active in order for the MD service itself to be considered active.

Once a Cyclic Report procedure instance is activated, it reports the values of the subscribed parameters at the specified delivery cycle until a subsequent STOP invocation message for that procedure instance is received.

The list of parameters in the START invocation may contain:

a) the names of individual monitored parameters that the Cross Support Complex has previously agreed to support, for all instances of all Functional Resources in the Service Package;

b) the labels of monitored parameter types. Labels provide a form of wildcard selection – when a parameter label is used, the values of all instances of that parameter type are reported for all instances of all Functional Resources in the Service Package;

c) the name of a list of parameter labels that has been agreed between the Cross Support Complex and the Mission. Such lists can be used to bundle monitored parameters that are routinely used as a group by the service user. A special case of the label list is the *default list*, which is a label list that is used by default when MD-CSTS user leaves the list of parameters empty;

d) the name of a specific Functional Resource instance, which is another wildcard selection mechanism, this one causing all parameter values for the named Functional Resource instance to be reported; and

e) the identifier of a Functional Resource type, yet another wildcard selection mechanism, this one causing all parameter values for all instances of that Functional Resource type that occur in the Service Package to be reported.

Optionally (by implementation), the list of parameters may also contain the names of procedure configuration parameters, procedure types, or procedure instance identifiers in order to receive periodic reports of the values of the configuration parameters of the procedures that comprise the MD service.

NOTE - The configuration parameters of the procedures that comprise the MD service are all static (that is, they do no change during the execution of the MD service). Some of these configuration parameters may be voluminous and cyclic reporting may have adverse operational effects. Therefore, cyclic reporting of procedure configuration parameters is optional.

If the list of parameters in the START invocation contains any parameter names, parameter labels, list names, resource types, or resource names that are unknown to or not supported by the Cross Support Complex, the START operation fails and a list of the unknown parameter names, labels, etc., are returned to the service user.

At least one instance of the MD-CSTS Cyclic Report procedure is mandatory for all implementations, but each implementation sets the maximum number of instances that can be active during the service instance provision period of the MD-CSTS.

2.2.2.2 Event Notifications

The MD-CSTS Notification procedure (see section 6) allows the service user to subscribe to a *list of events* in the START invocation message that activates the Notification procedure. Once a Notification procedure instance is activated, it notifies the service user upon the occurrence of any of the subscribed events a subsequent STOP invocation message for that procedure instance is received.

The START invocation may subscribe to notifiable notifiable events by using the names of events names, event labels, an event label list (named or default), a Functional Resource type, or a Functional Resource Name.

The Notification procedure is optional for MD-CSTS implementations. If an implementation supports the Notification procedure, multiple instances of the Notification procedure can be active concurrently, with each implementation setting the maximum number of instances that can be active during the service instance provision period of the MD-CSTS.

2.2.2.3 Query of Parameters

The Information Query procedure (see section 5) allows the service user to query the current values of functional resource monitored parameters and MD-CSTS procedure configuration parameters. The Information Query procedure consists of a single GET operation that can be used to query the values of different parameters every time that it is invoked.

The GET invocation message can be used to query Functional Resource parameter values using Functional Resource parameter names, parameter labels, a parameter label list (named and default), a Functional Resource Typs, or a Functional Resource Name.

The GET invocation message can also use procedure configuration parameter names, a Procedure Type, or a Procedure Instance Identifier to query the values of configuration parameter of the procedures that comprise the MD-CSTS instance.

The Information Query procedure is optional for MD-CSTS implementations. At most one instance of the Information Query procedure can exist during the service instance provision period of an MD-CSTS.

2.3 SERVICE MANAGEMENT

As defined in the Cross Support Reference Model: Part 1 (reference [2]), 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).

With regard to the production and provision of Monitored Data service instances, cross support service management:

a) May establish one or more named lists of Parameter Labels to be used to request groups of monitored parameters to be cyclically reported (and queried if the optional query capability is supported) during the MD-CSTS service instance provision period. One of these lists may be designated the default list of Parameter Labels, which means that it will be applied unless Parameter Names, Parameters Labels, etc., are explicitly requested. As specified in reference [1], each Parameter Label is composed of (a) the

Parameter Identifier of that parameter and (b) the Functional Resource Type with which that Parameter Identifier is associated;

b) May (if the Notification procedure is supported by the Complex) establish one or more named lists of Event Labels to be used to request groups of notifiable events to be reported during the MD-CSTS service instance provision period. One of these lists may be designated the default list of Event Labels, which means that it will be applied unless Event Names, Event Labels, etc., are explicitly requested. As specified in reference [1], each Event Label is composed of (a) the Event Identifier of that event and (b) the Functional Resource Type with which that Event Identifier is associated; and

c) Schedules the Service Packages which specify both the cross support services that are to be monitored and the Monitored Data service instances that transfer the monitored parameters to the users of the Monitored Data service instances.

The means by which service management performs these functions is outside the scope of this Recommendation.

2.4 CROSS SUPPORT VIEW

Figure 2-2 shows an example configuration of a Cross Support Complex providing instances of Monitored Data service to a Mission Data Operations System (MDOS). In this example, the functional resources that are configured to support the Service Package (and therefore available for monitoring via the MD-CSTS) are the Space Link Extension (SLE) Forward Communications Link Transmission Unit (FCLTU) transfer service (reference [E4]), the functional resource instances that are associated with FCLTU production (antenna, forward space link carrier transmission, and forward symbol stream transmission), the SLE Return All Frames (RAF) transfer service (reference [E5]), and the functional resource instances that are associated with RAF production (antenna, return space link carrier reception, and return symbol stream reception). The SLE transfer service instances and the production functional resources associated with them are not part of the Monitored Data service, but they are monitored through the Monitored data service.

NOTE 1 - For the purposes of this description, the functional resources named above and shown in the figure are names of functional resource types registered with SANA at the time of publication of this Recommended Standard. Subsequent changes to the SANA registry may have resulted in different names.

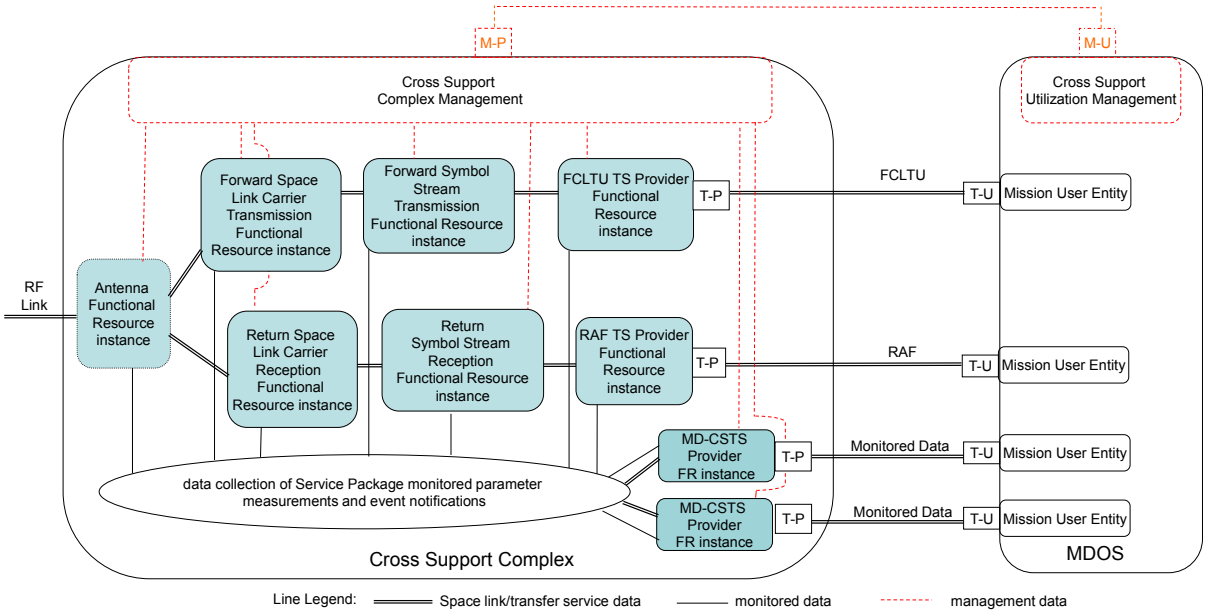


Figure 2-2: Example of the Management and Provision of Monitored Data Service Instances for a Service Package

As shown in Figure 2-2, a single Service Package may include multiple instances of Monitored Data service. All Monitored Data service instances (i.e., all MD-CSTS users) have access to all monitored parameters and notifiable events associated with the Service Package.

As also shown in the figure, the MD-CSTS instances are themselves monitored functional resources. Any managed parameters or notifiable events associated with the MD-CSTS instances are made available via those MD-CSTS instances in the same way that such information about any Functional Resource instance is made available.

NOTE 2- According to the Cross Support Reference Model (reference [2]), the responsibility for monitoring the overall state of 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 Space Communication Cross Support Service Management infrastructure (reference [E6]). Although Figure 2-2 shows the Mission User Entity (MUE) for the MD-CSTS as being separate from the UM role, it should be understood that in the nominal case the use of the MD-CSTS will be in support of a Service Management responsibility. However, there may be other uses of monitored data that are not linked to UM. For example, the principal investigator for a particular instrument may need to correlate her instrument data (returned via the RAF SLE service) with communication service status that is not available via the RAF service. Such a principal investigator could be a user of an MD-CSTS instance and subscribe to the monitored parameters of interest.

2.5 OPERATIONAL SCENARIO

This section presents an operational scenario of the MD service. This scenario is written using the Functional Resource Name, Parameter Name, Parameter Label, Event Name, Event Label, Procedure Type, and Procedure Instance Identifier representation conventions described in 1.5.4.2.

NOTE - The representation conventions in 1.5.4.2 represent the OID components as text descriptors enclosed in braces, e.g., {Association Control} represents the OID assigned as the Procedure Identifier for the Association Control procedure. The OIDs associated with Functional Resources (Functional Resource Types, functional resource parameter identifiers, and functional resource event identifiers) are formally registered with SANA, but the subset of those OIDs that are used in this operational scenario are identified in ANNEX E. The OIDs associated with CSTS Framework procedures (Procedure Types, procedure configuration parameter identifiers, and procedure event identifiers) are specified in the CSTS Specification Framework (reference [1]).

The following scenario is success-oriented. That is, no error conditions are addressed.

2.5.1 SERVICE AGREEMENT/SERVICE PLANNING ACTIVITIES

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

For the purpose of this scenario, the cross support services include a single S-Band forward space link and associated single instance of FCLTU SLE transfer service [E4], an S-Band return space link and associated single instance of RAF SLE transfer service [E5], and an X-Band return space link and associated single instance of RAF transfer service. Multinet supports both the optional Notification and Information Query procedures of the MD-CSTS standard in addition to the mandatory Cyclic Report procedure: the Service Agreement also includes a single instance of the MD-CSTS, which supports four instances of each of the Cyclic Report and Notification procedures, and one instance of the Information Query procedure.

The functional resource instances associated with these cross support services are:

a. Two instances of the Antenna functional resource type, one for X-Band and one for S-Band. The functional resource name for the X-Band Antenna is “[{Antenna}:1]”, and the functional resource name for the S-Band Antenna is “[{Antenna}:2]”;

b. One instance of the Forward Space Link Carrier Transmission functional resource type, with functional resource name “[{Forward Space Link Carrier Transmission}:1]” for the Forward S-Band;

c. One instance of the Forward Symbol Stream Transmission functional resource type for the Forward S-Band link, with functional resource name "[{Forward Symbol Stream Transmission}:1]";

d. One instance of the TC MC Multiplexing, Channel Synchronization, and Encoding functional resource type for Forward S-Band link, with functional resource name "[{TC MC Mux, Channel Sync and Encoding}:1]"

e. One instance of the FCLTU Transfer Service Provider functional resource type, with functional resource name "[{FCLTU TS Provider}:1]";

f. Two instances of the Return Space Link Carrier Reception functional resource type, one for X-Band and one for S-Band. The functional resource name for the X-Band Return Space Link Carrier Reception functional resource instance is "[{Return Space Link Carrier Reception}:1]" and the functional resource name for the S-Band Return Space Link Carrier Reception functional resource instance is "[{Return Space Link Carrier Reception}:2]";

g. Two instances of the Return Space Link Subcarrier Reception functional resource type, one for X-Band return space link and one for S-Band return space link. The functional resource name for the X-Band Return Space Link Subcarrier Reception functional resource instance is "[{Return Space Link Subcarrier Reception}:1]" and the functional resource name for the S-Band Return Space Link Subcarrier Reception functional resource instance is "[{Return Space Link Subcarrier Reception}:2];

h. Two instances of the Return Symbol Stream Reception functional resource type, one for X-Band return space link subcarrier and one for S-Band return space link subcarrier. The functional resource name for the X-Band Return Symbol Stream functional resource instance is "[{Return Symbol Stream Reception}:1]" and the functional resource name for the S-Band Return Symbol Stream functional resource instance is "[{Return Symbol Stream Reception}:2]";

NOTE - In this example the subcarriers are bi-phase shift key (BPSK)-modulated, so that only one symbol stream is carried by each subcarrier. If the data were instead QPSK-modulated onto the carrier with independent symbol streams on the I and Q channels, the functional resource name would also have to distinguish the channel, e.g., the I-channel might be designated "[{Return Symbol Stream Reception}:1] and the Q-channel might be designated "[{Return Symbol Stream Reception}:2]".

i. Two instances of the Return TM Synchronization and Decoding functional resource type, one for the X-Band return space link subcarrier symbol stream and one for then S-Band return space link subcarrier symbol stream. The functional resource name for the X-Band Return TM Synchronization and Decoding resource instance is "[{Return TM Synchronization and Decoding}:1]" and the functional resource name for the S-

Band Return TM Synchronization and Decoding resource instance is "[{Return TM Synchronization and Decoding}:2]";

j. Two instances of the RAF Transfer Service Provider functional resource type, one for the X-Band return space link subcarrier symbol stream and one for the S-Band return space link subcarrier symbol stream. The functional resource name for the X-Band RAF Transfer Service Provider resource instance is "[{RAF TS Provider}:1]" and the functional resource name for the S-Band RAF Transfer Service Provider resource instance is "[{RAF TS Provider}:2]";

k. One instance of the Monitored Data CSTS Provider functional resource type for the Service Package. The functional resource name for the MD-CSTS Provider resource instance is "[{MD CSTS Provider}:1]".

Among the Functional Resource monitored parameters collected and made available by the Multinet Complex are:

- the `pointingMode` for the Antenna FR Type. The Parameter Label for this parameter is [{Antenna}:{`pointingMode`}] (see E2for the actual OID values),
- the `actualTransmitFrequency` for the Forward Space Link Carrier Transmission FR Type. The Parameter Label for this parameter is [{Forward Space Link Carrier Transmission}:{`actualTransmitFrequency`}] (see E3 for the actual OID values),
- the `serviceInstanceState` and `numberOfCltusRadiated` (for the FCLTU Transfer Service Provider FR Type. The Parameter Labels for these parameters are [{FCLTU Transfer Service Provider}:{`serviceInstanceState`}] and [{FCLTU Transfer Service Provider}:{`numberOfCltusRadiated`}], respectively (see E6 for the actual OID values);
- the `actualReceiveFrequency` for the Return Space Link Carrier Reception FR Type. The Parameter Label for this parameter is [{Return Space Link Carrier Reception}:{`actualReceiveFrequency`}]. (see E7 for the actual OID values),
- the `subcarrierLockStatus` and `actualSubcarrierFrequency` for the Return Space Link Subcarrier Reception FR Type. The Parameter Labels for these parameters are [{Return Space Link Subcarrier Reception}:{`subcarrierLockStatus`}] and [{Return Space Link Subcarrier Reception}:{`actualSubcarrierFrequency`}], respectively (see E8 for the actual OID values),
- the `symbolSynchronizerLockStatus` for the Return Symbol Stream Reception FR Type. The Parameter Label for this parameter is [{Return Symbol Stream Reception}:{`symbolSynchronizerLockStatus`}] (see E9 for the actual OID values), and

- the numberOfFramesDelivered and serviceInstanceState for the RAF Transfer Service Provider FR Type. The Parameter Labels for these parameters are [{RAF Transfer Service Provider}:{numberOfFramesDelivered}] and [{RAF Transfer Service Provider}:{serviceInstanceState}], respectively (see E11 for the actual OID values).

Among the notifiable events made available by the Multinet Complex are:

- frameLockAcquired and lossOfFrameLock for the Return TM Synchronizaton and Decoding FR Type. The Parameter Labels for these events are [{Return TM Synchronization and Decoding}:{frameLockAcquired}] and [{Return TM Synchronization and Decoding}:{lossOfFrameLock}], respectively (see E10 for the actual OID values);

- svcProductionConfigured, svcProductionInterrupted, svcProductionHalted, and svcProductionOperational for the MD CSTS Provider FR Type. The Parameter Labels for these events are [{MD CSTS Provider}:{svcProductionConfigured}], [{MD CSTS Provider}:{svcProductionInterrupted}], [{MD CSTS Provider}:{svcProductionHalted}] and [{MD CSTS Provider}:{svcProductionOperational}], respectively (see E12 for the actual OID values),

- productionInterrupted, productionHalted, and productionRunning for the RAF TS Provider FR Type. The Parameter Labels for these events are [{RAF TS Provider}:{svcProductionConfigured}], [{RAF TS Provider}:{svcProductionInterrupted}], [{RAF TS Provider}:{svcProductionHalted}], and [{RAF TS Provider}:{svcProductionOperational}], respectively (see E11 for the actual OID values), and

- productionInterrupted, productionHalted, and productionOperational for the FCLTU TS Provider FR Type. The Parameter Labels for these events are [{FCLTU TS Provider}:{productionInterrupted}], [{FCLTU TS Provider}:{productionHalted}] and [{FCLTU TS Provider}: {productionOperational}], respectively (see E6 for the actual OID values).

As part of the Service Management activities that establish the relationship between the mission and the Complex, UM and CM negotiate a single monitored Parameter Label List that will serve as the default Label List for both the Cyclic Report and Information Query procedures. In this scenario, the default Label List is given the name “defaultLabelList” and contains the Parameter Labels for the FCLTU TS Provider numberOfCltusRadiated parameter and the RAF TS Provider numberOfFramesDelivered parameter. The

`defaultList` parameter of the Label List is set to TRUE to indicate that it is the default Label List.

{ "defaultLabelList";
TRUE;
[{FCLTU TS Provider} : {`numberOfCltusRadiated`}];
[{RAF TS Provider} : {`numberOfFramesDelivered`}]
}

The default Label List (and any other Label Lists that might be created) can be used by any Cyclic Report or Information Query procedure instance of any MD-CSTS instance in any Service Package that is established in the context of the Service Agreement.

Prior to the time at which cross support is desired, Utilization Management causes Complex Management to create a Service Package that specifies a Space Link Session with the S-Band forward link and both S-Band and X-Band return links to be provided and the start and stop times for each of the associated service provisions and productions. Included in the Service Package is an instance of a Monitored Data cross support transfer service.

2.5.2 SERVICE PACKAGE EXECUTION

2.5.2.1 Binding the MD-CSTS Instance

As of the scheduled beginning of the service instance provision period of the Monitored Data service instance, the service instance exists in the 'unbound' state. At the scheduled start time of the space link services and the production of the associated transfer services, the Complex establishes the space links with the spacecraft and begins processing of the signals to and from the spacecraft. Any time after the scheduled start of the service instance provisions periods of the RAF and FCLTU service instances, the users of those services may bind to those services and use them.

At any time following scheduled beginning of the service instance provision period of the Monitored Data service, the user of that service invokes the BIND operation of the Association Control procedure instance to bind to the service provider, transition the service instance to the 'bound.ready' state, and place each of the other procedure instances in the 'inactive' state. In this scenario, the scheduled beginning of the service instance provision period of the Monitored Data service is prior to the time at which the associated production process becomes operational, and the binding to the service instance occurs while the Production Status of the service is still merely configured.

2.5.2.2 Prime Instance of Cyclic Report Procedure

Following the successful binding of the service instance, the user of the Monitored Data service invokes the START operation for the prime instance of the Cyclic Report Procedure, which places the prime procedure in the 'active' state and the service instance in the 'bound.active' state. The parameter of the START invocation that is used to subscribe to monitored parameters is empty, thus indicating that the parameters of the default Label List

are to be reported. The START invocation also specifies that the reporting is to occur at two-second intervals.

The MD-CSTS provider invokes the TRANSFER-DATA operation of the prime Cyclic Report procedure instance to report the values of the monitored parameters at the specified two-second-interval. Each invocation of the TRANSFER-DATA operation for this instance of the Cyclic Report procedure transfers the values for the monitored parameters with the following Parameter Names:

- [[{FCLTU TS Provider}:1] : {numberOfCltusRadiated}];
- [[{RAF TS Provider}:1] : {numberOfFramesDelivered}], and
- [[{RAF TS Provider}:2)] : {numberOfFramesDelivered}].

Note that the use of a Label List causes every instance of a parameter with a Parameter Label that is in the list to be reported. Thus numberOfFramesDelivered is reported for both instances of the RAF TS Provider because both have the same Parameter Label.

Because the prime instance of the Cyclic Report Procedure is started when the production of the MD CSTS is simply 'configured' and not yet 'operational' values for the three parameters are reported as 'unavailable' until the MD service production becomes operational, at which time the actual values of the parameters begin to be reported.

2.5.2.3 Second Instance of Cyclic Report Procedure

The user of the MD-CSTS starts the second Cyclic Report Procedure instance to cyclically report a group of parameters every ten seconds. The START invocation lists the following Parameter Names:

- [[{Forward Space Link Carrier Transmission}:1] : {actualTransmitFrequency}];
- [[{Return Space Link Subcarrier Reception}:1] : {subcarrierLockStatus}];
- [[{Return Space Link Subcarrier Reception}:2] : {subcarrierLockStatus}];
- [[{FCLTU TS Provider}:1] : {serviceInstanceState}];
- [[{RAF TS Provider}:1] : {serviceInstanceState}] and
- [[{RAF TS Provider}:2] : {serviceInstanceState}].

The MD-CSTS provider invokes the TRANSFER-DATA operation of the second Cyclic Report procedure instance to report the values of the monitored parameters at the specified ten-second-interval. Each invocation of the TRANSFER-DATA operation for this instance of the Cyclic Report procedure transfers the values for the monitored parameters with the following Parameter Names:

- [[{Forward Space Link Carrier Transmission}:1] : {actualTransmitFrequency}];

- [[{Return Space Link Subcarrier Reception}:1] : {subcarrierLockStatus}];
- [[{Return Space Link Subcarrier Reception}:2] : {subcarrierLockStatus}];
- [[{FCLTU TS Provider}:1] : {serviceInstanceState}];
- [[{RAF TS Provider}:1] : {serviceInstanceState}] and
- [[{RAF TS Provider}:2] : {serviceInstanceState}].

As with the prime procedure instance, the values for these parameters are reported as 'unavailable' while the Production Status of the Monitored Data service is 'configured'.

2.5.2.4 Third Instance of Cyclic Report Procedure

The user of the MD-CSTS starts the third Cyclic Report Procedure instance to cyclically report (at ten-second intervals) all parameters of the S-Band Return Space Link Subcarrier Reception Functional Resource by using Functional Resource Name in the list-of-parameters:

- [{Return Space Link Subcarrier Reception}:2].

The MD-CSTS provider invokes the TRANSFER-DATA operation of the third Cyclic Report procedure instance to report the values of the monitored parameters at the specified ten-second-interval. Each invocation of the TRANSFER-DATA operation for this instance of the Cyclic Report procedure transfers the values for the monitored parameters with the following Parameter Names:

- [[{Return Space Link Subcarrier Reception}:2] : {subcarrierLockStatus}];
- [[{Return Space Link Subcarrier Reception}:2] : {actualSubcarrierFrequency}].

The values for these parameters are reported as 'unavailable' while the Production Status of the Monitored Data service is 'configured'.

2.5.2.5 First Instance of Notification Procedure

The user of the MD-CSTS starts the first Notification procedure instance to subscribe to the following Event Labels on occurrences of changes in the state of the frame synchronizer:

- [{Return TM Synchronization and Decoding } : {frameLockAcquired}];
- [{Return TM Synchronization and Decoding } : {lossOfFrameLock}].

By using Event Labels instead of Event Names in the subscription, the user subscribes to the frameLockAcquired' and 'lossOfFrameLock' event notifications for all instances of Return TM Synchronization and Decoding that are configured as part of the Service Package.

When the Return S-Band subcarrier locks, the MD-CSTS provider invokes the NOTIFY operation of the Notification procedure instance to signal the occurrence of the event with the Event Name
[[{Return TM Synchronization and Decoding }:2] : {frameLockAcquired}] .

When the Return X-Band subcarrier locks, the MD-CSTS provider invokes the NOTIFY operation of the Notification procedure instance to signal the occurrence of the event with the Event Name
[[{Return TM Synchronization and Decoding }:1] : {`frameLockAcquired`}].

2.5.2.6 Second Instance of Notification Procedure

The user of the MD-CSTS starts the second Notification procedure instance to subscribe to the notifiable events of the following Functional Resource Type on occurrences of changes in the Production Status of the production processes for the MD CSTS Provider:

- {MD CSTS Provider}.

By using Functional Resource Type in the subscription, the user subscribes to all event notifications for the MD CSTS Provider instance that is configured as part of the Service Package. This includes all Production Status-related notifications for the instance of that Functional Resource Type.

When the Monitored Data Production Status transitions from 'configured' to 'operational', the MD-CSTS provider invokes the NOTIFY operation of the Notification procedure instance to signal the occurrence of the event with the Event Name
[{MD CSTS Provider}:1] : {svcProductionOperational}].

2.5.2.7 Third Instance of Notification Procedure

The user of the MD-CSTS starts the third Notification procedure instance to subscribe to the notifiable events of the following Functional Resource Type on occurrences of changes in the Production Status of the production processes for the RAF TS Providers:

- {RAF TS Provider}.

By using the Functional Resource Type in the subscription, the user subscribes to all event notifications for all instances of the RAF TS Provider that are configured as part of the Service Package. This includes all Production Status-related notifications for that Functional Resource Type.

When the Production Status of the RAF service associated with the return X-Band symbol stream transitions from 'configured' to 'operational', the MD-CSTS provider invokes the NOTIFY operation of the Notification procedure instance to signal the occurrence of the event with the Event Name [{RAF TS Provider}:1] : {svcProductionOperational}].

When the Production Status of the RAF service associated with the return S-Band symbol stream transitions from 'configured' to 'operational', the MD-CSTS provider invokes the NOTIFY operation of the Notification procedure instance to signal the occurrence of the event with the Event Name [{RAF TS Provider}:2] : {svcProductionOperational}].

2.5.2.8 Fourth Instance of Notification Procedure

The user of the MD-CSTS starts the fourth Notification procedure instance to subscribe to the notifiable events of the following Functional Resource Type on occurrences of changes in the Production Status of the production processes for the FCLTU TS Provider:

- {FCLTU TS Provider}.

By using the Functional Resource Type in the subscription, the user subscribes to all event notifications for the FCLTU TS Provider instance that is configured as part of the Service Package. This includes all Production Status-related notifications for that Functional Resource Type.

When the Production Status of the FCLTU service associated with the forward S-Band symbol stream transitions from 'configured' to 'operational', the MD-CSTS provider invokes the NOTIFY operation of the Notification procedure instance to signal the occurrence of the event with the Event Name [{FCLTU TS Provider}:1] : {svcProductionOperational}].

2.5.2.9 Information Query Procedure

The Information Query procedure of the MD-CSTS can be used to retrieve the current values of parameters of Functional Resources that are configured as part of the Service Package. The Information Query procedure of the MD-CSTS can also be used to query the procedure configuration parameters of the Association Control, Cyclic Report, Information Query, and Notification procedures of the MD-CSTS instance executing those procedures.

2.5.2.9.1 Query of Functional Resource Parameters

At some time during the execution of the Service Package, the user of the MD-CSTS invokes the GET operation of the Information Query Procedure instance to request the current values of the pointingMode parameter for both the S-Band and X-Band antennas. The GET invocation contains the Parameter Label [{Antenna}:{`pointingMode`}], which causes the MD service instance to return the parameter values for both configured Antennas:

- [[{Antenna}:1] : {`pointingMode`}] for the X-Band antenna and

- [[{Antenna}:2] : {`pointingMode`}] for the S-Band antenna.

2.5.2.9.2 Query of Procedure Parameters

At some time during the execution of the Service Package, the user of the MD-CSTS invokes the GET operation of the Information Query procedure to retrieve the contents of the Label List Set used by the Cyclic Report procedure. The GET invocation contains the Parameter Name [[{Cyclic Report}:'prime procedure instance'] : {`pCRlistNames`}], which causes the MD service instance to return the Label List Set containing the single Label List that has been defined:

```
{   {   “defaultLabelList”;
        TRUE;
        [{FCLTU TS Provider} : {numberOfCltusRadiated}];
        [{RAF TS Provider} : {numberOfFramesDelivered}]
    }
}
```

NOTES

1 The OID for the Cyclic Report procedure configuration parameter identifier `pCRlistNames` is specified in reference [1].

2 There is only one set of parameter label lists for all instances of the Cyclic Report and Information Query procedure for all instances of the MD-CSTS operating in the context of a Service Agreement. Thus the use of the Cyclic Report prime procedure instance `pCRistNames` parameter is not uniquely required to retrieve the Label List Set. The `pCRlistNames` parameter of any instance of the Cyclic Report procedure could be used. Furthermore, the `pIQlistNames` parameter of the Information Query procedure could also be used to retrieve the same Label List Set.

3 The user of the MD service may also query label list contents by Procedure Instance Identifier. If the query is made on the Procedure Instance Identifier, all procedure parameters for the specified instance of the specified procedure type are reported. In the case of this scenario, the GET invocation on the prime procedure instance of the Cyclic Report procedure would return the all procedure parameters of that procedure instance. The Cyclic Report procedure has only one parameter (`pCRlistNames`), so the contents of `pCRlistNames` for the CR prime procedure instance would be reported. However, since the label lists are common to all instances of the procedure type, the same Label List Set would be returned regardless of which procedure instance is specified in the GET invocation.

4 The user of the MD service may also query label list contents by Procedure Type. If the query is made on the Procedure Type, all procedure parameters for all configured instances of the specified procedure type are reported. In the case of this scenario, where four instances of Cyclic Report procedure are configured (even though only three are activated), the GET invocation on the Cyclic Report procedure would return four sets of procedure parameters, so the contents of `pCRlistNames` for each of the four CR procedure instances would be reported. However, since the label lists are common to all instances of the procedure type, four copies of the same Label List Set would be returned.

2.5.2.10 Stopping the MD-CSTS Procedures and Unbinding the MD-CSTS Instance

At some times prior to the end of the service instance provision period the Monitored Data service instance, the user invokes (in no particular order) the STOP operations for the second and third Cyclic Report Procedure instances and the four instances of the Notification procedure.

The user then invokes the STOP operation for the prime Cyclic Report Procedure instance, which transitions the service instance to the ‘bound.ready’ state.

After all Cyclic Report Procedure instances and all Notification procedure instances have been stopped, but still prior to the end of the service instance provision period of the Monitored Data service instance, the user invokes the UNBIND operation of the Association Control procedure instance to transition the Monitored Data service instance to the ‘unbound’ state.

3 COMPOSITION OF THE MONITORED DATA CROSS SUPPORT TRANSFER SERVICE

3.1 GENERAL

The Monitored Data service may be implemented as defined herein without need for further extension or refinement.

The object identifiers for the Monitored Data service are specified in ANNEX A.

3.2 PROCEDURES OF THE MONITORED DATA CROSS SUPPORT TRANSFER SERVICE

3.2.1 The Monitored Data transfer service shall be composed of the Association Control, Cyclic Report, Notification, and Information Query procedures.

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

3.2.3 The Association Control procedure shall be adopted directly from the CSTS SFW (reference [1]).

3.2.4 The Version of the Association Control procedure shall be the same as the Version of the Association Control procedure from reference [1].

3.2.5 The Cyclic Report procedure shall be the primary procedure for the Monitored Data transfer service.

NOTE - Being the primary procedure implies that at least one instance of the procedure is mandatory.

3.2.6 The Cyclic Report procedure shall be refined from the Cyclic Report procedure defined in reference [1].

3.2.7 The Version of the Cyclic Report procedure shall be "1".

3.2.8 There shall be zero or more secondary instances of the Cyclic Report procedure.

NOTE - The number of secondary instances of the Cyclic Report procedure is unconstrained by this service specification. However, each real implementation sets the number of instances that will be instantiated.

3.2.9 The Information Query procedure shall be refined from the Information Query procedure defined in reference [1].

3.2.10 The Version of the Information Query procedure shall be "1".

3.2.11 There shall be zero or one secondary procedure instance of the Information Query procedure.

NOTE - The Information Query procedure is optional; an implementation is not required to include it.

3.2.12 The Notification procedure shall be refined from the Notification procedure defined in reference [1].

3.2.13 The Version of the Notification procedure shall be "1".

3.2.14 There shall be zero or more secondary procedure instances of the Notification procedure.

NOTES

1 The Notification procedure is optional; an implementation is not required to include it. But if an implementation does include the Notification procedure, it may include one or more instances of them. The number of secondary instances of the Notification procedure is unconstrained by this service specification. However, each real implementation sets the number of instances that will be instantiated.

2 Table 3-1 summarizes the procedures that comprise the Monitored Data transfer service, where (a) the "[P]" in the *Procedure* row designates Cyclic Report as the primary procedure; (b) *Version* = "-" indicates that the version of the service procedure is the same as that of the CSTS SFW procedure for those procedures that are directly adopted, and *Version* = "1" indicates the version of the refined and extended service procedures (Cyclic Report, Information Query, and Notification); (c) *No. of Instances* indicates the minimum and maximum number of allowed instances of each procedure type; (d) *Specification Approach* indicates which procedures are directly adopted or refined and extended; and (e) *Source* indicates the CSTS SFW procedure from which the service procedure is adopted or refined and extended.

Table 3-1: Monitored Data Transfer Service Procedures

Procedure	Association Control	Cyclic Report [P]	Information Query	Notification
Version	-	1	1	1
No. of Instances	1..1	1..*	0..1	0..*
Specification Approach	adopted	refined	refined	refined
Source	CSTS SFW [1]: Association Control	CSTS SFW [1]: Cyclic Report	CSTS SFW [1]: Information Query	CSTS SFW [1]: Notification

3.3 MONITORED DATA CROSS SUPPORT TRANSFER SERVICE STATE MACHINE

The Monitored Data Cross Support Transfer Service state machine shall conform to the state machine for a CSTS with a stateful prime procedure, as defined in the CSTS Framework (reference [1]).

4 CYCLIC REPORT PROCEDURE

4.1 DISCUSSION

4.1.1 PURPOSE

The Cyclic Report procedure of the Monitored Data service is used to cyclically report monitored parameter values for all Functional Resources configured as part of a Service Package.

The Cyclic Report procedure of the Monitored Data service may be used to cyclically report the procedure configuration parameters of the Association Control procedure, Cyclic Report procedure, the Information Query procedure, and the Notification procedure of the MD-CSTS instance that executes those procedures. Support for cyclic reporting of procedure configuration parameters is optional.

4.1.2 CONCEPT

The concept of the Cyclic Report procedure is the same as that of the parent CSTS SFW Cyclic Report procedure, with the exception of what is reported when Parameter Labels or Functional Resource Types are used to subscribe to functional resource monitored parameters.

When a Parameter Label is used to subscribe to functional resource monitored parameters, the Cyclic Report procedure reports the values of all functional resource monitored parameters that have that Parameter Label for all instances of the associated Functional Resource Type that are configured as part of the Service Package. This includes Parameter Labels that are represented by named and default lists.

When a Functional Resource Type is used to subscribe to functional resource monitored parameters, the Cyclic Report procedure reports the values of all functional resource monitored parameters of all instances of that Functional Resource Type that are configured as part of the Service Package.

As described in reference [1], the Cyclic Report procedure has access to all configuration parameters of the procedures that comprise a CSTS that implements the Cyclic Report. In the case of the MD-CSTS, the Association Control, Cyclic Report, Information Query, and Notification procedures each has one or more configuration parameters that can be cyclically reported by the Cyclic Report procedure.

NOTE - The configuration parameters of the Association Control, Cyclic Report, Information Query, and Notification procedures are all static for the duration of the MD service instance. Therefore cyclic reporting of these parameters would result in the same information being reported every cycle. Implementers should consider the utility of repeatedly reporting the same values when deciding whether or not to implement the optional cyclic reporting of configuration parameters.

4.2 PROCEDURE TYPE IDENTIFIER

The procedure type identifier for the MD-CSTS Cyclic Report procedure shall be the same as the CSTS Framework Cyclic Report procedure.

4.3 REFINEMENT

The Cyclic Report procedure refines the behavior of the CSTS SFW Cyclic Report procedure by modifying the behavior of the procedure when Parameter Labels or Functional Resource Types are subscribed.

4.4 BEHAVIOR

4.4.1 The Stopping and Terminating behaviors of the Cyclic Report procedure shall be the same as those of the CSTS SFW Cyclic Report procedure as specified in reference [1].

4.4.2 The Starting behavior of the Cyclic Report procedure shall be the same as that of the CSTS SFW Cyclic Report procedure as specified in reference [1], except that the requirement that specifies the conditions for determining that the START invocation is valid shall be replaced with the requirement as specified in 4.4.2.1.

4.4.2.1 Upon invocation of the START operation, the service provider shall confirm that the invocation is valid. A START invocation for the Cyclic Delivery procedure is valid if it meets any one of the following conditions:

a) if the `list-of-parameters` parameter is left empty (signifying subscription to default list of Functional Resource parameter labels) and the default list has been established;

b) if the `list-of-parameters` parameter contains one or more parameter names of parameters of Functional Resource instances that are configured as part of the Service Package;

c) if the implementation of the service supports cyclic reporting of procedure configuration parameters and if the `list-of-parameters` parameter contains one or more parameter names of procedure configuration parameters for instances of the Association Control, Cyclic Report, Information Query, or Notification procedures of the MD-CSTS instance that executes the Cyclic Report procedure;

d) if the `list-of-parameters` parameter contains one or more parameter labels of parameters of Functional Resource Types of which instances are configured as part of the Service Package;

e) if the `list-of-parameters` parameter contains one parameter list name representing a list of default Functional Resource parameter labels that has been established;

f) if the `list-of-parameters` parameter contains one Functional Resource Name of a Functional Resource instance that is configured as part of the Service Package;

g) if the `list-of-parameters` parameter contains one Functional Resource Type of which at least one instance is configured as part of the Service Package;

h) if the implementation of the service supports cyclic reporting of procedure configuration parameters and if the `list-of-parameters` parameter contains the Procedure Type of the Association Control, Cyclic Report, Information Query, or Notification procedure;

i) if the implementation of the service supports cyclic reporting of procedure configuration parameters and if the `list-of-parameters` parameter contains one Procedure Instance Identifier of an active instance of the Association Control, Cyclic Report, Information Query, or Notification procedures of the MD-CSTS instance that executes the Cyclic Report procedure.

4.4.3 The Transferring Data behavior of the Cyclic Report procedure shall be the same as that of the CSTS SFW Cyclic Report procedure as specified in reference [1], except that the requirement that specifies which qualified parameters are to be reported as a function of the content of the `list-of-parameters` parameters of the START invocation shall be replaced with the requirement as specified in 4.4.3.1.

NOTE - The nominal SFW Cyclic Report procedure operates properly only when a service that uses the procedure has at most only one instance of any monitored parameter type. This constraint does not apply to the MD-CSTS; there can be multiple instances of the same monitored parameter type.

4.4.3.1 The Service Provider shall deliver the qualified parameters (parameter name, the value, the type, and the qualifier of the parameters) using the `qualified-parameters` parameter. If `list-of-parameters`:

a) is left empty, then for each parameter label represented by the default list, deliver a qualified parameter for every instance of the parameter identified by that Parameter Label for all instances of its associated Functional Resource Type that are configured as part of the Service Package.

b) contains one or more parameter names of monitored parameters of instances of Functional Resources that are configured as part of the Service Package, then deliver the qualified parameters for each of the listed parameters;

c) contains one or more parameter names of procedure configuration parameters of instances of Association Control procedure, Cyclic Report procedure, Information Query procedure, or the Notification procedure of the MD-CSTS instance, then deliver the qualified parameters for each of the listed parameters;

d) contains one or more individual parameter labels, then for each label deliver a qualified parameter for every instance of the parameter identified by that Parameter Label for all instances of its associated Functional Resource Type that are configured as part of the Service Package;

e) contains the name of a list of parameter labels, then for each parameter label represented by the named list, deliver a qualified parameter for every instance of the parameter identified by that Parameter Label for all instances of its associated Functional Resource Type that are configured as part of the Service Package;

f) contains a Functional Resource Name, then return the qualified parameters for all of the parameters for the named instance of the Functional Resource;

NOTE - This behavior is the same as that of the CSTS SFW Cyclic Report procedure.

g) contains a Functional Resource Type, then for each parameter label associated with that Functional Resource Type, deliver a qualified parameter for every instance of each parameter for that Functional Resource Type for all instances of that Functional Resource Type that are configured as part of the Service Package;

h) contains the Procedure Type for the Association Control procedure, Cyclic Report procedure, Information Query procedure, or the Notification procedure, then deliver a qualified parameter for every instance of each configuration parameter for that procedure type for all instances of that procedure type that are active in the MD-CSTS instance. Support for reporting procedure configuration parameters by the Cyclic Report procedure is optional; or

i) contains the Procedure Instance Identifier for an instance of the Association Control procedure, Cyclic Report procedure, Information Query procedure, or the Notification procedure, then deliver a qualified parameter for every instance of each configuration parameter for the named instance of that procedure type for the MD-CSTS instance. Support for reporting procedure configuration parameters by the Cyclic Report procedure is optional.

4.5 REQUIRED OPERATIONS

4.5.1 The Cyclic Report procedure shall use the STOP and TRANSFER-DATA operations of the CSTS SFW Cyclic Report procedure without extension or refinement.

4.5.2 The Cyclic Report procedure shall use the START operation of the CSTS SFW Cyclic Report procedure as refined in 4.5.3.

NOTE - Table 4-1 summarizes the operations of the Cyclic Report procedure of the Monitored Data service.

Table 4-1: Cyclic Report Procedure Required Operations

Operations	Extended	Refined	Procedure Blocking/Non-Blocking
START	N	Y	Blocking
STOP	N	N	Blocking
TRANSFER-DATA	N	N	Non Blocking

4.5.3 START (CONFIRMED)

4.5.3.1 Invocation, Return, and Parameters

The START invocation of the MD-CSTS Cyclic Report procedure uses the parameters of the Cyclic Report START invocation as defined in reference [1].

The START return of the MD-CSTS Cyclic Report procedure uses the parameters of the Cyclic Report START return as defined in reference [1], except for the definitions of the values of the `diagnostic` parameter, which are refined as specified in 4.5.3.1.1.

4.5.3.1.1 `diagnostic` refinement

4.5.3.1.1.1 The START return of the Cyclic Report procedure of the MD-CSTS shall use the definitions of the 'unable to comply', 'out of service', 'unknown list name', 'default not defined', 'out of range', 'unknown procedure type', and 'unknown procedure instance identifier' diagnostic values as defined for the CSTS SFW Cyclic Report START return in reference [1].

4.5.3.1.1.2 The definition of the 'unknown Functional Resource Type' diagnostic shall be refined as follows:

> "'unknown Functional Resource Type'—the Functional Resource Type contained in the `list-of-parameters` parameter is unknown to the service provider or the Functional Resource Type is not associated with any Functional Resource that is configured as part of the Service Package. The unknown Functional Resource Type shall be returned with the diagnostic."

4.5.3.1.1.3 The definition of the 'unknown Functional Resource Name' diagnostic shall be refined as follows:

> "'unknown Functional Resource Name'—while the Functional Resource Type is known, the Functional Resource Name contained in the `list-of-parameters` is unknown to the service provider or the selected Functional Resource instance is not associated with any Functional Resource that is configured as part of the Service Package. The unknown Functional Resource Name shall be returned with the diagnostic."

4.5.3.1.1.4 The definition of the 'unknown parameter identifier' diagnostic shall be refined as follows:

"'unknown parameter identifier' - one or more Parameter Identifiers contained in the `list-of-parameters` parameter are unknown to the service provider for one of the following reasons:

1) the Functional Resource specified as part of the parameter name is not associated with any Functional Resource that is configured as part of the Service Package;

2) the Functional Resource type as part of the parameter label is not associated with any Functional Resource that is configured as part of the Service Package;

3) a parameter with the given Published Identifier does not exist for the specified Functional Resource instance or type.

The list of unknown Parameter Names or Parameter Labels shall be returned with the `diagnostic`. For each unknown Parameter Identifier that is contained in a Parameter Name in the `list-of-parameters`, the Parameter Name shall be returned. For each unknown Parameter Identifier that is contained in a Parameter Label in the `list-of-parameters`, the Parameter Label shall be returned."

4.6 CONFIGURATION PARAMETERS

The Cyclic Report procedure adopts the configuration parameters of the CSTS SFW Cyclic Report procedure without addition or modification.

4.7 PROCEDURE STATE TABLE

The Cyclic Report procedure adopts the state table of the CSTS SFW Cyclic Report procedure without addition or modification.

`

5 INFORMATION QUERY PROCEDURE

5.1 DISCUSSION

5.1.1 PURPOSE

The Information Query procedure of the Monitored Data service is used to query the current values of monitored parameters for any Functional Resource configured as part of a Service Package.

The Information Query procedure of the Monitored Data service is used to query the procedure configuration parameters of the Association Control procedure, Cyclic Report procedure, the Information Query procedure, and the Notification procedure of the MD-CSTS instance that executes those procedures.

5.1.2 CONCEPT

The concept of the Information Query procedure is the same as that of the parent CSTS SFW Information Query procedure, with the exception of what is reported when Parameter Labels or Functional Resource Types are used to request the values of monitored parameters.

When a Parameter Label is used to request values of monitored parameters, the Information Query procedure returns the values of all monitored parameters that have that Parameter Label for all instances of the associated Functional Resource Type that are configured as part of the Service Package. This includes Parameter Labels that are represented by named and default lists.

When a Functional Resource Type is used to request values of monitored parameters, the Information Query procedure returns the values of all monitored parameters of all instances of that Functional Resource Type that are configured as part of the Service Package.

As described in reference [1], the GET operation of the Information Query procedure has access to all configuration parameters of the procedures that comprise a CSTS that implements the Information Query procedure. In the case of the MD-CSTS, the Association Control, Cyclic Report, Information Query, and Notification procedures each has one or more configuration parameters that can be queried by the Information Query procedure.

5.2 PROCEDURE TYPE IDENTIFIER

The procedure type identifier for the MD-CSTS Information Query procedure shall be the same as the CSTS Framework Information Query procedure.

5.3 REFINEMENT

The Information Query procedure refines the behavior of the CSTS SFW Information Query procedure by specifying the behavior of the procedure when Parameter Labels or Functional Resource Types are used in queries.

5.4 BEHAVIOR

NOTE - The Getting Parameters behavior of the CSTS SFW Information Query procedure is the same as the behavior of the CSTS SFW GET operation. The refinement of the behavior of the MD-CSTS Information Query procedure is specified as a refinement of the behavior of the CSTS SFW GET operation.

5.4.1 The Getting Parameters behavior of the Information Query GET operation shall be the same as that of the CSTS SFW GET operation shall be the same as specified in reference [1], with exception of (a) the requirement that specifies the conditions for determining that the GET invocation is valid and (b) the requirement that specifies which qualified parameters are to be returned as a function of the content of the `list-of-parameters` parameters of the GET invocation, which shall be replaced with the requirements as specified in 5.4.1.1 and 5.4.1.2, respectively.

5.4.1.1 The GET invocation is valid if it meets any one of the following conditions:

a) if the `list-of-parameters` parameter is left empty (signifying the default list of Functional Resource monitored parameter labels) and the default list has been established;

b) if the `list-of-parameters` parameter contains one or more parameter names of parameters of Functional Resource instances that are configured as part of the Service Package;

c) if the `list-of-parameters` parameter contains one or more parameter names of configuration parameters for instances of the Association Control, Cyclic Report, Information Query, or Notification procedures of the MD-CSTS instance that executes the Notification procedure;

d) if the `list-of-parameters` parameter contains one or more parameter labels of parameters of Functional Resource Types of which instances are configured as part of the Service Package;

e) if the `list-of-parameters` parameter contains one parameter list name representing a list of default Functional Resource parameter labels that has been established;

f) if the `list-of-parameters` parameter contains one Functional Resource Name of a Functional Resource instance that is configured as part of the Service Package;

g) if the `list-of-parameters` parameter contains one Functional Resource Type of which at least one instance is configured as part of the Service Package;

h) if the `list-of-parameters` parameter contains the Procedure Type of the Association Control, Cyclic Report, Information Query, or Notification procedure; or

i) if the `list-of-parameters` parameter contains one Procedure Instance Identifier of an instance of the Association Control, Cyclic Report, Information Query, or Notification procedures of the MD-CSTS instance that executes the Information Query procedure.

5.4.1.2 If the GET invocation is valid, the service provider shall return the qualified parameters (parameter name, the value, the type, and the qualifier of the parameters) using the `qualified-parameters` parameter. If `list-of-parameters`:

a) is left empty, then for each parameter label represented by the default list, return a qualified parameter for every instance of the parameter identified by that Parameter Label for all instances of its associated Functional Resource Type that are configured as part of the Service Package;

b) contains one or more parameter names of monitored parameters of instances of Functional Resources that are configured as part of the Service Package, then return the qualified parameters for each of the listed parameters;

c) contains one or more parameter names of procedure configuration parameters of instances of Association Control procedure, Cyclic Report procedure, Information Query procedure, or the Notification procedure of the MD-CSTS instance, then return the qualified parameters for each of the listed parameters;

d) contains one or more parameter labels, then for each label, return a qualified parameter for every instance of the parameter identified by that Parameter Label for all instances of its associated Functional Resource Type that are configured as part of the Service Package;

e) contains the name of a list of parameter labels, then for each parameter label represented by the named list return a qualified parameter for every instance of the parameter identified by that Parameter Label for all instances of its associated Functional Resource Type that are configured as part of the Service Package;

f) contains a Functional Resource Name, then return the qualified parameters for all of the parameters for the named instance of the Functional Resource ;

NOTE - This behavior is the same as that of the CSTS SFW Information Query procedure.

g) contains a Functional Resource Type, then for each parameter label associated with that Functional Resource Type, return a qualified parameter for every instance of each

parameter for that Functional Resource Type for all instances of that Functional Resource Type that are configured as part of the Service Package;

h) contains the Procedure Type for the Association Control procedure, Cyclic Report procedure, Information Query procedure, or the Notification procedure, then return a qualified parameter for every instance of each configuration parameter for that procedure type for all instances of that procedure type that are active in the MD-CSTS instance; or

i) contains the Procedure Instance Identifier for an instance of the Association Control procedure, Cyclic Report procedure, Information Query procedure, or the Notification procedure, then return a qualified parameter for every instance of each configuration parameter for the named instance of that procedure type for the MD-CSTS instance.

NOTE - The nominal SFW Information Query procedure operates properly only when a service that uses the procedure has at most only one instance of any monitored parameter type. This constraint does not apply to the MD-CSTS; there can be multiple instances of the same monitored parameter type.

5.4.2 The Terminating behaviors of the Information Query procedure shall be the same as that of the CSTS SFW Information Query procedure as specified in reference [1].

5.5 REQUIRED OPERATIONS

5.5.1 The Information Query procedure shall use the GET operation of the CSTS SFW Information Query procedure as refined in 5.5.2.

Note - Table 5-1 summarizes the operations of the Information Query procedure of the Monitored Data service.

Table 5-1: Information Query Procedure Required Operations

Operations	Extended	Refined	Procedure Blocking/Non-Blocking
GET	N	Y	Non Blocking

5.5.2 GET (CONFIRMED)

5.5.2.1 Invocation, Return, and Parameters

The GET invocation of the MD-CSTS Information Query procedure uses the parameters of the Information Query GET invocation as defined in reference [1].

The GET return of the MD-CSTS Information Query procedure uses the parameters of the Information Query GET return as defined in reference [1], except for the definitions of the values of the `diagnostic` parameter, which are refined as specified in 5.5.2.1.1.

5.5.2.1.1 `diagnostic` refinement

5.5.2.1.1.1 The GET return of the Information Query procedure of the MD-CSTS shall use the definitions of the 'unknown list name', 'default not defined', 'unknown procedure type' and 'unknown procedure instance identifier' diagnostic values as defined for the CSTS SFW Information Query GET return in reference [1].

5.5.2.1.1.2 The definition of the 'unknown Functional Resource Type' diagnostic shall be refined as follows:

> "'unknown Functional Resource Type'—the Functional Resource Type contained in the `list-of-parameters` parameter is unknown to the service provider or the Functional Resource Type is not associated with any Functional Resource that is configured as part of the Service Package. The unknown Functional Resource Type shall be returned with the diagnostic."

5.5.2.1.1.3 The definition of the 'unknown Functional Resource Name' diagnostic shall be refined as follows:

> "'unknown Functional Resource Name'—while the Functional Resource Type is known, the Functional Resource Name contained in the `list-of-parameters` is unknown to the service provider or the selected Functional Resource instance is not associated with any Functional Resource that is configured as part of the Service Package. The unknown Functional Resource Name shall be returned with the diagnostic."

5.5.2.1.1.4 The definition of the 'unknown parameter identifier' diagnostic shall be refined as follows:

> "'unknown parameter identifier' - one or more Parameter Identifiers contained in the `list-of-parameters` parameter are unknown to the service provider for one of the following reasons:
>
> 1) the Functional Resource specified as part of the parameter name is not associated with any Functional Resource that is configured as part of the Service Package;
> 2) the Functional Resource type as part of the parameter label is not associated with any Functional Resource that is configured as part of the Service Package;
> 3) a parameter with the given Published Identifier does not exist for the specified Functional Resource instance or type.
>
> The list of unknown Parameter Names or Parameter Labels shall be returned with the `diagnostic`. For each unknown Parameter Identifier that is contained in a Parameter Name in the `list-of-parameters`, the Parameter Name shall be returned. For each

unknown Parameter Identifier that is contained in a Parameter Label in the `list-of-parameters`, the Parameter Label shall be returned."

5.6 CONFIGURATION PARAMETERS

The Information Query procedure adopts the configuration parameters of the CSTS SFW Information Query procedure without addition or modification.

5.7 PROCEDURE STATE TABLE

The Information Query procedure adopts the state table of the CSTS SFW Information Query procedure without addition or modification.

6 NOTIFICATION PROCEDURE

6.1 DISCUSSION

6.1.1 PURPOSE

The Notification procedure of the Monitored Data service is used to report event notifications for all Functional Resources configured as part of a Service Package.

NOTE - As described in reference [1], the Notification procedure has access to all procedure-specific notifiable events of the procedures that comprise a CSTS that implements the Notification procedure. In the case of the MD-CSTS, none of the procedures that comprise the service (Association Control, Cyclic Report, Information Query, and Notification) has any procedure-specific events that could be reported by the Notification procedure.

6.1.2 CONCEPT

The concept of the Notification procedure is the same as that of the parent CSTS SFW Notification procedure, with the exception of what is reported when Event Labels or Functional Resource Types are used to subscribe to functional resource notifiable events.

When an Event Label is used to subscribe to notifiable events, the Notification procedure notifies the user of the occurrences of all events that have that Event Label for all instances of the associated Functional Resource Type that are configured as part of the Service Package. This includes Event Labels that are represented by named and default lists.

When a Functional Resource Type is used to subscribe to notifiable events, the Notification procedure notifies the user of the occurrences of all events of all instances of that Functional Resource Type that are configured as part of the Service Package.

6.2 PROCEDURE TYPE IDENTIFIER

The procedure type identifier for the MD-CSTS Notification procedure shall be the same as for the CSTS Framework Notification procedure.

6.3 REFINEMENT

6.3.1 The Notification procedure refines the behavior of the CSTS SFW Notification procedure by modifying the behavior of the procedure when Event Labels or Functional Resource Types are subscribed.

6.3.2 The Notification procedure refines the definitions of the standard ‘production configured’, production operational’, production interrupted’, and ‘production halted’ event notifications.

6.4 BEHAVIOR

6.4.1 The Stopping and Terminating behaviors of the Notification procedure shall be the same as those of the CSTS SFW Notification procedure as specified in reference [1].

6.4.2 The Starting behavior of the Notification procedure shall be the same as that of the CSTS SFW Notification procedure as specified in reference [1], except that the requirement that specifies the conditions for determining that the START invocation is valid shall be replaced with the requirement as specified in6.4.2.1.

6.4.2.1 Upon invocation of the START operation, the service provider shall confirm that the invocation is valid. A START invocation for the Notification procedure is valid if it meets any one of the following conditions:

a) if the `list-of-events` parameter is left empty (signifying subscription to default list of Functional Resource parameter labels) and the default list has been established;

b) if the `list-of-events` parameter contains one or more event names of notifiable events of Functional Resource instances that are configured as part of the Service Package;

c) if the `list-of-events` parameter contains one or more event labels of notifiable events of Functional Resource Types of which instances are configured as part of the Service Package;

d) if the `list-of-events` parameter contains one event label list name representing a list of default Functional Resource event labels that has been established;

e) if the `list-of-parameters` parameter contains one Functional Resource Name of a Functional Resource instance that is configured as part of the Service Package.

NOTE - The CSTS SFW Notification procedure also supports subscription to procedure events of the procedures of the CSTS that implements the Notification procedure. However, none of the procedures of the MD-CSTS have procedure events, so any request to subscribe to procedure events using Event Names, Procedure Type or Procedure Instance Identifier would result in a negative return with the corresponding diagnostic for unknown event identifier, unknown procedure type, or unknown procedure instance identifier, respectively.

6.4.3 The Transferring Data behavior of the Notification procedure shall be the same as that of the CSTS SFW Notification procedure as specified in reference [1], except that the requirement that specifies which events are to be notified as a function of the content of the `list-of-events` parameters of the START invocation shall be replaced with the requirement as specified in 6.4.3.1.

6.4.3.1 Upon the occurrence of any of the notifiable events to which the procedure instance has been subscribed, the Service Provider shall invoke the NOTIFY operation to inform the Service User of the occurrence of the event. If `list-of-events`:

a) is left empty, then for each event label represented by the default list, the procedure shall invoke the NOTIFY operation upon the occurrence of any of the events identified by any of the Event Labels represented by the default list for all instances of its associated Functional Resource Type that are configured as part of the Service Package;

b) contains one or more individual functional resource event names, the procedure shall invoke the NOTIFY operation upon the occurrence of any of the specifically-names events;

NOTE - This behavior is the same as that of the CSTS SFW Notification procedure.

c) contains one or more individual event labels, the procedure shall invoke the NOTIFY operation upon the occurrence of any of the events identified by any of the Event Labels for any instances of their associated Functional Resource Types that are configured as part of the Service Package;

d) contains the name of a list of event labels, the procedure shall invoke the NOTIFY operation upon the occurrence of any of the events identified by any of the Event Labels represented by the named list for all instances of its associated Functional Resource Type that are configured as part of the Service Package;

e) contains a Functional Resource Name, then return all events for that instance of the Functional Resource Type named by that Functional Resource Name; or

NOTE - This behavior is the same as that of the CSTS SFW Notification procedure.

f) contains a Functional Resource Type, then for each event associated with that Functional Resource Type, return the events for any instance of that Functional Resource Type that is configured as part of the Service Package.

NOTE - The nominal SFW Notification procedure operates properly only when a service that uses the procedure has at most only one instance of any notifiable event type. This constraint does not apply to the MD-CSTS; there can be multiple instances of the same notifiable event type.

6.5 REQUIRED OPERATIONS

6.5.1 The Notification procedure shall refine the START operation as specified in 6.5.4.

6.5.2 The Notification procedure shall refine the NOTIFY operation of the CSTS SFW Notification procedure as specified in 6.5.4.

6.5.3 The Notification procedure shall use the STOP operation of the CSTS SFW Notification procedure without extension or refinement.

Note - Table 6-1 summarizes the operations of the Notification procedure.

Table 6-1: Notification Procedure Required Operations

Operations	Extended	Refined	Procedure Blocking/Non-Blocking
START	N	Y	Blocking
STOP	N	N	Blocking
NOTIFY	N	Y	Non Blocking

6.5.4 START (CONFIRMED)

6.5.4.1 Invocation, Return, and Parameters

The START invocation of the MD-CSTS Notification procedure uses the parameters of the Information Query GET invocation as defined in reference [1].

The START return of the MD-CSTS Notification procedure uses the parameters of the Notification START return as defined in reference [1], except for the definitions of the values of the `diagnostic` parameter, which are refined as specified in 6.5.4.1.1.

6.5.4.1.1 `diagnostic` refinement

6.5.4.1.1.1 The START return of the Notification procedure of the MD-CSTS shall use the definitions of the 'unable to comply', 'out of service', 'unknown list name', 'default not defined', 'unknown procedure type' and 'unknown procedure instance identifier' diagnostic values as defined for the CSTS SFW Notification START return in reference [1].

6.5.4.1.1.2 The definition of the 'unknown Functional Resource Type' diagnostic shall be refined as follows:

> "'unknown Functional Resource Type'—the Functional Resource Type contained in the `list-of-events` is unknown to the service provider or the Functional Resource Type is not associated with any Functional Resource that is configured as part of the Service Package. The unknown Functional Resource Type shall be returned with the diagnostic."

6.5.4.1.1.3 The definition of the 'unknown Functional Resource Name' diagnostic shall be refined as follows:

'unknown Functional Resource Name'—while the Functional Resource Type is known, the Functional Resource Name contained in the `list-of-events` is unknown to the service provider or the selected Functional Resource instance is not associated with any Functional Resource that is configured as part of the Service Package. The unknown Functional Resource Name shall be returned with the diagnostic

6.5.4.1.1.4 The definition of the 'unknown event identifier' diagnostic shall be refined as follows:

"'unknown event identifier'—one or more Event Identifiers contained in the `list-of-events` parameter are unknown to the service provider for one of the following reasons:

1) the Functional Resource specified as part of the event name is not associated with any Functional Resource that is configured as part of the Service Package;

2) the Functional Resource type as part of the event label is not associated with any Functional Resource that is configured as part of the Service Package;

3) an event with the given Published Identifier does not exist for the specified Functional Resource instance or type.

The list of unknown Event Names or Event Labels shall be returned with the `diagnostic`. For each unknown Event Identifier that is contained in an Event Name in the `list-of-events`, the Event Name shall be returned. For each unknown Event Identifier that is contained in an Event Label in the `list-of-events`, the Event Label shall be returned."

6.5.5 NOTIFY (UNCONFIRMED)

6.5.5.1 Invocation and Parameters

The NOTIFY invocation of the MD-CSTS Notification procedure uses the parameters of the Notification NOTIFY invocation as defined in reference [1], except for the definition of the `notification-type` parameter, which is refined as specified in 6.5.5.1.1.

6.5.5.1.1 `notification-type` refinement

6.5.5.1.1.1 The value of the **`notification-type`** shall be any one of the types specified by the common NOTIFY operation in the CSTS SFW, with the following refined definitions:

a) 'production configured': With respect to the production status of the Monitored Data service, the definition of the 'production configured' event shall be refined to mean that configuration of the Monitored Data Collection Functional Resource has been completed

NOTE - This refined definition applies to the event with the Published Identifier `svcProductionConfigured` in the `CCSDS-CSTS-OBJECT-IDENTIFIERS` module specified in the CSTS SFW (reference [1]).

b) production interrupted': With respect to the production status of the Monitored Data service, the definition of the 'production interrupted' event shall be refined to mean that the Monitored Data Collection Functional Resource has been stopped because of a condition that may be temporary.

NOTE - This refined definition applies to the event with the Published Identifier `svcProductionInterrupted` in the `CCSDS-CSTS-OBJECT-IDENTIFIERS` module specified in the CSTS SFW.

c) 'production halted': With respect to the production status of the Monitored Data service, the definition of the 'production halted' event shall be refined to mean that the Monitored Data Collection Functional Resource has been stopped by management action.

NOTE - This refined definition applies to the event with the Published Identifier `svcProductionHalted` in the `CCSDS-CSTS-OBJECT-IDENTIFIERS` module specified in the CSTS SFW.

d) 'production operational': With respect to the production status of the Monitored Data service, the definition of the 'production operational' event shall be refined to mean that the Monitored Data Collection Functional Resource has changed to 'operational'.

NOTE - This refined definition applies to the event with the Published Identifier `svcProductionOperational` in the `CCSDS-CSTS-OBJECT-IDENTIFIERS` module specified in the CSTS SFW.

NOTE - With respect to the production status of the Monitored Data service, the definition of the 'production configuration change' event is not applicable because there are no configuration parameters of the Monitored Data Collection Functional Resource that can change dynamically.

6.6 CONFIGURATION PARAMETERS

The Notification procedure adopts the configuration parameters of the CSTS SFW Notification procedure without addition or modification.

6.7 PROCEDURE STATE TABLE

The Notification procedure adopts the state table of the CSTS SFW Notification procedure without addition or modification.

7 CONFIGURATION PARAMETERS

7.1 GENERAL

The configuration and operation of an instance of the Monitored Data transfer service requires that the configuration parameters specified in this section be established at the service provider prior to the start of the service instance provision period for that Monitored Data service instance. This section specifies how the configuration parameters are to be set.

7.2 ASSOCIATION CONTROL PROCEDURE MANAGED INFORMATION

NOTE - The CSTS SFW defers the setting of the service-user-responding-timer, `initiator-identifier`, `responder-identifier`, `responder-port-identifier`, and `service-instance-identifier` configuration parameters to the service that uses the Association Control procedure.

7.2.1 The service-user-responding-timer (reference [1]) shall be a managed parameter.

7.2.2 The `initiator-identifier` (reference [1]) shall be a managed parameter.

7.2.3 The `responder-identifier` (reference [1]) shall be a managed parameter.

7.2.4 The `responder-port-identifier` (reference [1]) shall be a managed parameter.

7.2.5 The `service-instance-identifier` (reference [1]) shall be a managed parameter.

7.3 CYCLIC REPORT PROCEDURE MANAGED INFORMATION

7.3.1 Service Management may establish one or more named lists of functional resource monitored parameter labels in the context of each Service Agreement.

7.3.2 Each named label list of functional resource monitored parameters shall be available to all instances of Cyclic Report and MD Information Query procedures of all MD-CSTS instances in the context of the same Service Agreement.

7.3.3 For each named parameter label list, service management shall establish the name of the list and the set of Parameter Labels that is represented by that list name.

7.3.4 The syntax of the parameter label list names shall be as specified for parameter lists in the CSTS SFW (reference [1]).

7.3.5 Service Management may designate one of the named parameter label lists as the default list of parameter labels in the context of each Service Agreement.

7.3.6 The default list of parameter labels shall be made available for use as the default list by all instances of Cyclic Report and Information Query procedures of all CSTS instances in the context of the same Service Agreement.

7.4 NOTIFICATION PROCEDURE MANAGED INFORMATION

NOTE - The Notification procedure is optional. If the MD Notification procedure is not available in an instance of the Monitored Data service, the values of the managed information associated with the MD Notification procedure are undefined.

7.4.1 Service Management may establish one or more named lists of notifiable event labels in the context of each Service Agreement.

7.4.2 Each named list of notifiable events shall be available for use by all Monitored Data transfer service instances operating in the context of the same Service Agreement.

7.4.3 For each named notifiable event list, service management shall establish the name of the list and the set of Event Labels that is represented by that list name

7.4.4 The syntax of the monitored parameter label list names shall be as specified for parameter lists in the CSTS SFW (reference [1]).

7.4.5 Service Management may designate one of the named event label lists as the default list of notifiable event labels in the context of each Service Agreement.

7.5 INFORMATION QUERY PROCEDURE MANAGED INFORMATION

NOTE 1 - The Information Query procedure uses the same named lists of parameter labels and default list of parameter labels as the Cyclic Report Procedure (see 7.4.1 and 7.4.5).

NOTE 2 - There are no other configuration parameters of the Information Query procedure that the CSTS SFW defers to the service that uses the procedure.

.

ANNEX A

SERVICE OBJECT IDENTIFIERS MODULE

(NORMATIVE)

```
CCSDS-MONITORED-DATA-OBJECT-IDENTIFIERS
    {iso identified-organization (3) standards-producing-
    organization(112) ccsds(4) css (4) csts(1)  services (2)
    serviceIdentifiers(2)   monitoredData (1)
    monitoredDataModules (4) identifiers (1)
}
DEFINITIONS
IMPLICIT TAGS
::= BEGIN

EXPORTS  monitoredDataDerivedServices
,        monitoredDataExtServiceParameters
,        monitoredDataServiceProcedures
,        mdCstsProvider
,        mdCstsProviderParametersId
,        mdCstsProviderEventsId
;

IMPORTS  servicesIdentifiers
,        crossSupportFunctionalities
    FROM CCSDS-CSTS-OBJECT-IDENTIFIERS

         svcProductionStatus,
,        svcProductionConfigured
,        svcProductionOperational
,        svcProductionInterrupted
,        svcProductionHalted
    FROM  CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS

         pCRlistNames
,        pIQlistNames
,        pNlistNames
    FROM  CCSDS-CSTS-FW-FUNCTIONAL-RESOURCES-OBJECT-
          IDENTIFIERS
;

-- CCSDS-CSTS-OBJECT-IDENTIFIERS
-- CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS and
-- CCSDS-CSTS-FW-FUNCTIONAL-RESOURCES-OBJECT-IDENTIFIERS
-- are defined in the CSTS Framework [1]

 -- ************************************************************
```

```
-- Root Object Identifiers of the Service
monitoredData OBJECT IDENTIFIER   ::= {servicesIdentifiers 1}
monitoredDataDerivedServices OBJECT IDENTIFIER   ::=
                  {monitoredData 1}
monitoredDataExtServiceParameters OBJECT IDENTIFIER   ::=
                  {monitoredData 2}
monitoredDataServiceProcedures    OBJECT IDENTIFIER   ::=
                  {monitoredData 3}
-- ************************************************************
-- Root Object Identifiers of the MD-CSTS Provider
-- Functional Resource Type
mdCstsProvider     OBJECT IDENTIFIER   ::=                    {
crossSupportFunctionalities 17}
mdCstsProviderParametersId   OBJECT IDENTIFIER   ::=
                  {mdCstsProvider 1}
mdCstsProviderEventsId OBJECT IDENTIFIER ::=
                  {mdCstsProvider 2}
mdCstsProviderDirectivesId OBJECT IDENTIFIER ::=
                  {mdCstsProvider 3}


-- ************************************************************
-- Generic Service Object Identifiers used by the
-- Monitored Data service
-- The Cyclic Report and Information Query procedures may
-- request the status of the production process using the OID
-- svcProductionStatus


-- The Notification procedure may receive event
-- notifications regarding change of production
-- process status using the following OIDs:
-- svcProductionConfigured
-- svcProductionOperational
-- svcProductionInterrupted, or
-- svcProductionHalted


-- ************************************************************
-- Framework Procedure Object Identifiers used by the
-- Monitored Data service
-- The Cyclic Report and Information Query procedures may
-- request the values of procedure configuration parameters
-- with the OIDs
```

```
-- pCRlistNames
-- pIQlistNames
-- pNlistNames

-- ************************************************************

-- Root Object Identifiers of the Monitored Data Collection
-- Functional Resource Type

mdCollection                OBJECT IDENTIFIER   ::=         {
                            crossSupportFunctionalities 18}

mdCollectionParametersId    OBJECT IDENTIFIER   ::=
                            {mdCollection 1}

mdCollection                OBJECT IDENTIFIER ::=
                            {mdCollection 2}

mdCollectionDirectivesId    OBJECT IDENTIFIER ::=
                            {mdCstsCollection 3}


END
```

ANNEX B

IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA

(NORMATIVE)

B1 INTRODUCTION

B1.1 OVERVIEW

This annex provides the Implementation Conformance Statement (ICS) Requirements List (RL) for an implmentation of the *Monitored Data Cross Support Transfer Service*, CCSDS 922.1-R-1, January 2014. CCSDS 922.1 specifies the requirements on the provider of the Monitored Data Cross Support Transfer Service.

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

a) the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;

b) 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;

c) 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);

d) a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

B1.2 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 a 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:

a) M mandatory;

b) O optional;

c) O.<n> optional, but support of at least one of the group of options labeled by the same numeral <n> is required;

d) C<n> conditional as defined in corresponding expression below table;

e) X prohibited;

f) 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:

a) Y Yes, supported by the implementation;

b) N No, not supported by the implementation;

c) 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 E of the CSTS SFW (reference [1].using ASN.1. The ASN.1 data type specifications constrain 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 in annex E, 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>" not 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:

a) Y — Yes, the range/set defined in the Recommended Specification is fully supported by the implementation;

b) SV<n> — The range/set defined in the Recommended Specification is not fully supported by the implementation. The supported subset is documented below the table.

B1.3 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 B1.2. 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.

B2 PICS PROFORMA FOR THE MONITORED DATA CSTS PROTOCOL (CCSDS 922.1-B-1)

B2.1 GENERAL INFORMATION

The PICS for an MD-CSTS implementation shall encompass the filled in Table B-1toTable B-4.

Table B-1: Identification of PICS

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

Table B-2: Identification of Implementation Under Test

Implementation name	
Implementation version	
Special Configuration	
Other Information	

Table B-3: 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 B-4: Identification of Specification

CCSDS 922.1-B-1	
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 [ ]

A1.1 REQUIREMENTS LIST

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

Table B-5: Required Procedures

Procedures				
Item	**Description**	**Reference**	**Status**	**Support**
proc-1	Association Control	CSTS SFW 4.3	M	
proc-2	Unbuffered Data Delivery	CSTS SFW 4.4	M	
proc-3	Information Query	CSTS SFW 4.9	O	
proc-4	Cyclic Report	CSTS SFW 4.10	M	
proc-5	Notification	CSTS SFW 4.11	O	

The Unbuffered Data Delivery procedure is mandatory in the sense that the Cyclic Report procedure (which is mandatory) is derived from the Unbuffered Data Delivery procedure. In this MD-CSTS ICS, all requirements for the Unbuffered Data Delivery procedure are covered by the requirements for the Cyclic Report procedure.

Table B-6: Required PDUs

PDUs								
Item	PDU	Ref.	Service-Provider-System		Service-User-System		Relay	
			Status	Support	Status	Support	Status	Support
pdu-1	BindInvocation	CSTS SFW E3.5	M		M		N/A	
pdu-2	BindReturn	CSTS SFW E3.5	M		X		N/A	
pdu-3	PeerAbortInvocation	CSTS SFW E3.5	M		M		N/A	
pdu-4	UnbindInvocation	CSTS SFW E3.5	M		M		N/A	
pdu-5	UnbindReturn	CSTS SFW E3.5	M		M		N/A	
pdu-6	GetInvocation	CSTS SFW E3.4	C1		C1		N/A	
pdu-7	GetReturn	CSTS SFW E3.4	C1		C1		N/A	
pdu-8	NotifyInvocation	CSTS SFW E3.4	C2		C2		N/A	
pdu-9	StartInvocation	CSTS SFW E3.4	C3		C3		N/A	
pdu-10	StartReturn	CSTS SFW E3.4	C3		C3		N/A	
pdu-11	StopInvocation	CSTS SFW E3.4	C3		C3		N/A	
pdu-12	StopReturn	CSTS SFW E3.4	C3		C3		N/A	
pdu-13	TransferData-Invocation	CSTS SFW E3.4	C4		C4		N/A	

C1: IF proc-3 THEN M ELSE N/A
C2: IF proc-5 THEN M ELSE N/A
C3: IF proc-4 OR proc-5 THEN M ELSE N/A
C4: IF proc-4 THEN M ELSE N/A

Table B-7: BIND Invocation Parameters

Parameters of the BindInvocation PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
bindInv-1	invokerCredentials	CSTS SFW E3.3	M			
bindInv-2	invokeId	CSTS SFW E3.3	M			
bindInv-3	procedureInstanceId	CSTS SFW E3.3	M		AV1	
bindInv-4	initiatorIdentifier	CSTS SFW E3.3	M			
bindInv-5	responder-PortIdentifier	CSTS SFW E3.3	M			
bindInv-6	serviceType	CSTS SFW E3.3	M			
bindInv-7	versionNumber	CSTS SFW E3.3	M			
bindInv-8	serviceInstance-Identifier	CSTS SFW E3.3	M			
bindInv-9	extensionParameter	CSTS SFW E3.5	M			

AV1: procedureInstanceId (bindInv-3) is constrained to the value 'association control'.

The parameters in Table B-7 that reference CSTS SFW E3.5 are contained in the BindInvocation type defined in CSTS SFW E3.5.

The parameters in Table B-7 that reference CSTS SFW E3.3 are contained in the complex standardInvocationHeader parameter of type StandardInvocationHeader in the BindInvocation type defined in CSTS SFW E3.5. The StandardInvocationHeader type is defined in CSTS SFW E3.3.

Table B-8: BIND Return Parameters

Parameters of the BindReturn PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
bindRet-1	performerCredentials	CSTS SFW E3.3	M			
bindRet-2	invokeId	CSTS SFW E3.3	M			
bindRet-3	result	CSTS SFW E3.3	M			
bindRet-4	responderIdentifier	CSTS SFW E3.5	M			
bindRet-5	positive	CSTS SFW E3.3	C5		'not used'	
bindRet-6	diagnostics	CSTS SFW E3.3	C6			
bindRet-7	extended	CSTS SFW E3.3	C6			
bindRet-8	extensionParameter	CSTS SFW E3.5	M		'not used'	
bindRet-9	extensionDiagnostics	CSTS SFW E3.5	C6		'not used'	

C5: IF bindRet-3 = 'positive result' THEN M ELSE N/A
C6: IF bindRet-3 = 'negative result' THEN M ELSE N/A

If result (bindRet-3) is 'positive result' then (a) the responderIdentifier (bindRet-4) and extensionParameter (bindRet-8) parameters are defined in the type AssocBindPosReturnExt (see CSTS SFW E3.5).

If result (bindRet-3) is 'negative result', then (a) the diagnostics (bindRet-6) parameter carries the standard diagnostic parameter values defined in CSTS SFW E3.3 and the additional diagnostic parameter values defined by the type AssocBindDiagnosticsExt in CSTS SFW E3.5, (b) the extensionDiagnostics (bindRet-9) parameter is defined by the type AssocBindDiagnosticsExt in CSTS SFW E3.5, and (c) the responderIdentifier (bindRet-4) and extensionParameter (bindRet-8) parameters are defined in the type AssocBindNegReturnExt in CSTS SFW E3.5.

NOTE - The parameters in Table B-8 that reference CSTS SFW E3.3 are contained in the complex StandardReturnHeader type is defined in CSTS SFW E3.3. The BindReturn type defined in CSTS SFW E3.5 is of type StandardReturnHeader.

Table B-9: PEER-ABORT Invocation Parameters

Parameters of the PeerAbortInvocation PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
peerAbortInv-1	diagnostic	CSTS SFW E3.5	M		40..51, 127	

Table B-10: UNBIND Invocation Parameters

Parameters of the UnbindInvocation PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
unbindInv-1	invokerCredentials	CSTS SFW E3.3	M			
unbindInv-2	invokeId	CSTS SFW E3.3	M			
unbindInv-3	procedureInstanceId	CSTS SFW E3.3	M		AV2	
unbindInv-4	extensionParameter	CSTS SFW E3.3	M		'not used'	

AV2: procedureInstanceId (unbindInv-3) is constrained to the value 'association control'.

Table B-11: UNBIND Return Parameters

Parameters of the UnbindReturn PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
unbindRet-1	performerCredentials	CSTS SFW E3.3	M			
unbindRet-2	invokeId	CSTS SFW E3.3	M			
unbindRet-3	result	CSTS SFW E3.3	M		AV3	
unbindRet-4	positive	CSTS SFW E3.3	M		'not used'	
unbindRet-5	diagnostics	CSTS SFW E3.3	X			
unbindRet-6	extended	CSTS SFW E3.3	X			
unbindRet-7	extensionParameter	CSTS SFW E3.3	M		'not used'	

AV3: The result (unbindRet-3) parameter for the UnbindReturn is constrained to the value 'positive result'.

Table B-12: GET Invocation Parameters

Parameters of the GetInvocation PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
getInv-1	invokerCredentials	CSTS SFW E3.3	M			
getInv-2	invokeId	CSTS SFW E3.3	M			
getInv-3	procedureInstanceId	CSTS SFW E3.3	M		AV4	
getInv-4	listOfParameters	CSTS SFW E3.4	M			
getInv-5	extensionParameter	CSTS SFW E3.4	M		NULL	

AV4: procedureInstanceId (getInv-3) is constrained to one of the two values 'prime procedure' or 'secondary procedure'.

The parameters in Table B-12 that reference CSTS SFW E3.4 are contained in the GetInvocation type defined in CSTS SFW E3.4.

The parameters in Table B-12 that reference CSTS SFW E3.3 are contained in the complex standardInvocationHeader parameter of type StandardInvocationHeader in the GetInvocation type defined in CSTS SFW E3.4. The StandardInvocationHeader type is defined in CSTS SFW E3.3.

Table B-13: GET Return Parameters

Parameters of the GetReturn PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
getRet-1	performerCredentials	CSTS SFW E3.3	M			
getRet-2	invokeId	CSTS SFW E3.3	M			
getRet-3	result	CSTS SFW E3.3	M			
getRet-4	qualifiedParameters	CSTS SFW E3.3	C7			
getRet-5	diagnostics	CSTS SFW E3.3	C8			
getRet-6	extended	CSTS SFW E3.3	C8		'not used'	
getRet-7	extensionParameter	CSTS SFW E3.3	C7		'not used'	
getRet-8	extensionDiagnostics	CSTS SFW E3.3	C8		'not used'	

C7: IF getRet-3 = 'positive result' THEN M ELSE N/A
C8: IF getRet-3 = 'negative result' THEN M ELSE N/A

The extensionParameter (getRet-7) parameter is defined in the type GetPosReturnExt (see CSTS SFW E3.3).

The diagnostics (getRet-5) parameter carries the standard diagnostic parameter values defined for the StandardReturnHeader type in CSTS SFW E3.3 and the additional diagnostic parameter values defined for the type GetDiagnosticsExt in CSTS SFW E3.3. The extensionDiagnostics (getRet-8) parameter is defined by the type GetDiagnosticsExt in CSTS SFW E3.3.

Table B-14: START Invocation Parameters

Parameters of the StartInvocation PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
startInv-1	invokerCredentials	CSTS SFW E3.3	M			
startInv-2	invokeId	CSTS SFW E3.3	M			
startInv-3	procedureInstanceId	CSTS SFW E3.3	M		AV5	
startInv-4	deliveryCycle	CSTS SFW E3.12	C9			
startInv-5	listOfParameters	CSTS SFW E3.12	C9			
startInv-6	lsitOfEvents	CSTS SFW E3.13	C10			
startInv-7	extensionParameter	CSTS SFW E3.1 2 or CSTS SFW E3.1 3	M		'not used'	

C9: IF proc-4 THEN M ELSE N/A
C10: IF proc-5 THEN M ELSE N/A
AV5: procedureInstanceId (startInv-3) is constrained to one of the two values 'prime procedure' or 'secondary procedure'.

The parameters in Table B-14 that reference CSTS SFW E3.12 are contained in the CyclicReportStartInvocExt type defined in CSTS SFW E3.12.

The parameters in Table B-14 that reference CSTS SFW E3.13 are contained in the NotificationStartInvocExt type defined in CSTS SFW E3.13.

The parameters in Table B-14 that reference CSTS SFW E3.3 are contained in the complex standardInvocationHeader parameter of type StandardInvocationHeader in the StartInvocation type defined in CSTS SFW E3.3. The StandardInvocationHeader type is defined in CSTS SFW E3.3.

The extensionParameter (startInv-7) parameter is defined in different extension types for the START invocations for the Cyclic Report and Notification procedures. For the Cyclic Report procedure, it is defined in the CyclicReportStartInvocExt type defined in CSTS SFW E3.12. For the Notification procedure, it is defined in the NotificationStartInvocExt type defined in CSTS SFW E3.13. In either case, its value is set to 'not used'.

Table B-15: START Return Parameters

Parameters of the StartReturn PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
startRet-1	performerCredentials	CSTS SFW E3.3	M			
startRet-2	invokeId	CSTS SFW E3.3	M			
startRet-3	result	CSTS SFW E3.3	M			
startRet-4	positive	CSTS SFW E3.3	C11		'not used'	
startRet-5	diagnostics	CSTS SFW E3.12 or CSTS SFW E3.13	C12			
startRet-6	extended	CSTS SFW E3.3	C12		'not used'	
startRet-7	extensionDiagnostics	CSTS SFW E3.12 or CSTS SFW E3.13	M		'not used'	

C11: IF getRet-3 = 'positive result' THEN M ELSE N/A
C12: IF getRet-3 = 'negative result' THEN M ELSE N/A

The diagnostics (startRet-5) parameter is defined in different extension types for the START returns for the Cyclic Report and Notification procedures. For both procedures, the parameter carries the standard diagnostic parameter values defined for the StandardReturnHeader type in CSTS SFW E3.3 and the additional diagnostic parameter values defined for the type StartDiagnosticsExt in CSTS SFW E3.3. For the Cyclic Report procedure, the parameter also carries the diagnostic parameter values defined for the CyclicReportStartDiagnosticsExt type defined in CSTS SFW E3.12. For the Notification procedure, the parameter also carries the diagnostic parameter values defined for the CyclicReportStartDiagnosticsExt type defined in NotificationStartNegReturnDiagnosticsExt type defined in CSTS SFW E3.13.

The diagnostics extensionDiagnostics (startRet-7) parameter is defined in different extension types for the START returns for the Cyclic Report and Notification procedures. For the Cyclic Report procedure, the parameter is defined in the CyclicReportStartDiagnosticsExt type defined in CSTS SFW E3.12. For the Notification procedure, the parameter is defined in the NotificationStartNegReturnDiagnosticsExt type defined in CSTS SFW E3.13. In either case, the value of extensionDiagnostics parameter is set to 'not used'.

Table B-16: STOP Invocation Parameters

Parameters of the StopInvocation PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
stopInv-1	invokerCredentials	CSTS SFW E3.3	M			
stopInv-2	invokeId	CSTS SFW E3.3	M			
stopInv-3	procedureInstanceId	CSTS SFW E3.3	M		AV6	
stopInv-4	extensionParameter	CSTS SFW E3.3	M		'not used'	

AV6: procedureInstanceId (stopInv-3) is constrained to one of the two values 'prime procedure' or 'secondary procedure'.

Table B-17: STOP Return Parameters

Parameters of the StopReturn PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
stopRet-1	performerCredentials	CSTS SFW E3.3	M			
stopRet-2	invokeId	CSTS SFW E3.3	M			
stopRet-3	result	CSTS SFW E3.3	M			
stopRet-4	positive	CSTS SFW E3.3	C13		'not used'	
stopRet-5	diagnostics	CSTS SFW E3.3	C14			
stopRet-6	extended	CSTS SFW E3.3	C14			
stopRet-7	extensionDiagnostics	CSTS SFW E3.3	C14		'not used'	

C13: IF getRet-3 = 'positive result' THEN M ELSE N/A
C14: IF getRet-3 = 'negative result' THEN M ELSE N/A

The diagnostics (stopRet-5) parameter carries the standard diagnostic parameter values defined for the StandardReturnHeader type in CSTS SFW E3.3. The extensionDiagnostics (getRet-8) parameter is defined by the type Diagnostics in CSTS SFW E3.3.

Table B-18: NOTIFY Invocation Parameters

Parameters of the NotifyInvocation PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
notifyInv-1	invokerCredentials	CSTS SFW E3.3	M			
notifyInv-2	invokeId	CSTS SFW E3.3	M			

Parameters of the NotifyInvocation PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
notifyInv-3	procedureInstanceId	CSTS SFW E3.3	M		AV7	
notifyInv-4	eventTime	CSTS SFW E3.3				
notifyInv-5	notificationType	CSTS SFW E3.3			AV8	
notifyInv-6	extensionParameter	CSTS SFW E3.3			'not used'	

AV7: procedureInstanceId (notifyInv-3) is constrained to one of the two values 'prime procedure' or 'secondary procedure'.

AV8: notificationType (notifyInv-5) is constrained to one of the four values `svcProductionConfigured`, `svcProductionInterrupted`, `svcProductionHalted`, or `svcProductionOperational`.

Table B-19: TRANSFER-DATA Invocation Parameters

Parameters of the TransferDataInvocation PDU						
Item	Parameter	Ref.	Status	Support	Values	
					Allowed	Supported
transferDataInv-1	invokerCredentials	CSTS SFW E3.3	M			
transferDataInv-2	invokeId	CSTS SFW E3.3	M			
transferDataInv-3	procedureInstanceId	CSTS SFW E3.3	M		AV9	
transferDataInv-4	generationTime	CSTS SFW E3.3	M			
transferDataInv-5	sequenceCounter	CSTS SFW E3.3	M			
transferDataInv-6	qualifiedParameters	CSTS SFW E3.12	M			
transferDataInv-7	extensionParameter	CSTS SFW E3.12	M		'not used'	

AV9: procedureInstanceId (transferDataInv-3) is constrained to one of the two values 'prime procedure' or 'secondary procedure'.

The qualifiedParameters (transferDataInv-6) and extensionParameter (transferDataInv-7) parameters are defined in the CyclicReportTransferDataRef type defined in CSTS SFW E3.12. The CyclicReportTransferDataRef type refines the abstract `data` parameter of the base TransferDataInvocation type defined in CSTS SFW E3.4.

ANNEX C

FUNCTIONAL RESOURCE PARAMETER AND EVENT NOTIFICATION PRODUCTION

(NORMATIVE)

C1 GENERAL

During the execution of a Service Package, the functional resources that are instantiated by that Service Package supply two kinds of data for use by MD-CSTS instances:

a) The current values of all functional resource monitored parameters that are collected and made available for use by that spaceflight mission; and

b) Event notifications for all notifiable events that are made available for use by that spaceflight mission.

During the execution of a Service Package, each active Cyclic Report procedure instance of each MD-CSTS instance retrieves and transfers the current values of the cyclically-reported monitored parameters to which that procedure instance is subscribed, at the polling interval for that procedure instance.

Upon invocation of the GET operation of the Query Information procedure (if implemented and supported), each MD-CSTS instance retrieves and transfers the current values of the queriable monitored parameters named in the GET invocation.

If the Notification procedure is implemented and supported, upon the occurrence of a notifiable event, each MD-CSTS instance that is subscribed to that notifiable event transfers the notification to the user of that service instance.

C2 PRODUCTION OF FUNCTIONAL RESOURCE PARAMETER VALUES

C2.1 During the execution of a Service Package, the production processes associated with that Service Package shall produce and maintain the current values of all functional resource monitored parameters for all production functional resource instances.

C2.2 Each production monitored parameter type shall belong to a specified functional resource type.

C2.3 Each monitored parameter value shall be identified by its Parameter Name.

C2.4 Each monitored parameter shall be of the type (e.g., integer) specified for its parameter type.

C3 PRODUCTION OF EVENT NOTIFICATIONS

C3.1 During the execution of a Service Package, the production processes associated with that Service Package shall emit notifications on the occurrence of all notifiable events for all functional resource instances.

C3.2 Each production notifiable event type shall belong to a specified functional resource type.

C3.3 Each event shall be identifiable by its Event Name.

C3.4 Each notifiable event shall be accompanied by the additional information (if any) that is associated with that event type.

ANNEX D

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

D1 SECURITY CONSIDERATIONS

D1.1 INTRODUCTION

This subsection describes security aspects of the Monitored Data service.

The CSTS Specification Framework (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. The Monitored Data 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 MD-CSTS. Specification of the various underlying communications technologies, and in particular their associated security provisions, are outside the scope of this Recommended Standard.

D1.2 SECURITY CONCERNS WITH RESPECT TO THE MONITORED DATA SERVICE

The Statements of Security Concerns subsection of the CSTS Specification Framework 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.

D1.3 POTENTIAL THREATS AND ATTACK SCENARIOS

As a member of the suite of CSTSes, the Monitored Data service depend on unspecified mechanisms operating in the underlying communications service, or on privacy-ensuring capabilities in the service-specific application processes that interoperate through the Framework procedures, 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 MD-CSTS protocol data units as they traverse the WAN between service user and service provider.

The CSTS Specification Framework 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 monitored parameters such as the azimuth and elevation of the ground station antenna and the actual receive and transmit frequencies could assist an attacker in acquiring return link data or jamming 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 MD-CSTS instances by which they could gain access to the data transferred via the service.

The MD-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 MD-CSTS.

If the service provider of the MD-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.

D1.4 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

The consequences of not applying security to the MD-CSTS are possible degradation and loss of ability to use the service, or the interception of data that provides information that could aid in the acquisition and/or jamming of the space link itself.

D2 SANA CONSIDERATIONS

The MD-CSTS relies on SANA registries:

- To provide the identification and definition of Functional Resource parameters and events;
- To register the object identifiers (OIDs) for the MD-CSTS.

D2.1 IDENTIFICATION AND DEFINITION OF FUNCTIONAL RESOURCE PARAMETERS AND EVENTS

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

```
{iso 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. Under each Functional Resource type OID, the parameters, events, and directives are registered under dedicated subnodes.

Maintenance of the 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.

Maintenance of the registry of the Functional Resource types, parameters, events, and directives under the `agencyFunctionalities` subnode is under the purview of designated Agency-level control authorities. The process and procedure for designating Agency-level control authorities is documented in the CSTS Specification Framework.

D2.2 REGISTRATION OF MD-CSTS OBJECT IDENTIFIERS

The `CCSDS-MONITORED-DATA-OBJECT-IDENTIFIERS` module defined in ANNEX A is registered with SANA. Maintenance of the SANA regisrty of this module and the OIDs contained within occurs as a result of changes in the published version of the Recommended Standard.

D3 PATENT CONSIDERATIONS

There are no patents that are known to apply to the technology used in the Monitored Data service.

ANNEX E

EXAMPLE FUNCTIONAL RESOURCE TYPE OBJECT IDENTIFIER REGISTRY

(INFORMATIVE)

E1 INTRODUCTION

This annex provides an example registry of object identifiers for the Functional Resource Types that are used in the Operational Scenario for the MD-CSTS (see section 2.5) . This set of object identifiers has been extracted from the SANA Functional Resource registry at the time of publication of this Recommended Standard. 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.

These examples are included in this informative annex to illustrate the contents of SANA Functional Resource Type registry entries and to provide explicit object identifiers for the Functional Resource Types, parameters, and events that are cited abstractly in the Operational Scenario.

The example object identifiers documented in this annex may not always correspond to those in the SANA registry. Over time, the Functional Resource object identifiers that are maintained in the SANA registry may evolve as Functional Resource Types, parameters, and events are added and/or deprecated. Real MD-CSTS implementations must always select and support Functional Resource Types, parameters, and notifiable events as specified in the SANA registry.

[Red Book Temporary Note – The SANA Functional Resource registry has not been established as of the Red Book 1 review of the MD-CSTS Draft Recommended Standard. It will be established prior to publication of the MD-CSTS Recommended Standard (Blue Book), and this annex will reflect the content of the SANA registry as of the publication of the Blue Book).]

The Functional Resource Types used in the Operational Scenario section of this Recommended Standard are:

- Antenna;
- Forward Space Link Carrier Transmission;
- Forward Symbol Stream Transmission;
- FCLTU Transfer Service Provider;
- Return Space Link Carrier Reception;
- Return Space Link Subcarrier Reception;

- Return Symbol Stream Reception;
- Return TM Synchronization and Decoding;
- RAF Transfer Service Provider; and
- Monitored Data CSTS Provider.

NOTE - The Forward Symbol Stream Transmission functional group type must be present in the configuration of the forward link, and so it is listed here as being part of the configuration that enables the operational scenario presented in 2.5. However, the operational scenario does not include the querying of any parameters of the Forward Symbol Stream Transmission functional group type, and so this functional group type is not explicitly mentioned in 2.5.

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

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

The `crossSupportFunctionalities` subnode is used to register CCSDS-standard Functional Resource Types, and the `agenciesFunctionalities` subnode is used to register Agency-specific Functional Resource Types. All Functional resource Types listed above are registered under the `crossSupportFunctionalities` subnode:

```
{crossSupportResources  crossSupportFunctionalities (1)}
```

Thus the OID for the `crossSupportFunctionalities` subnode is 3/112/4/4/2/1.

The `crossSupportFunctionalities` subnode is the root node for all Functional Resource Types described in this Annex. Each of the Functional Resource Types described in this Annex are registered as nodes directly under this root node.

As specified in reference [1], the parameters, notifiable events, and directives that are specific to each Functional Resource Type are registered under `parametersId`, `eventsId`, and `directivesId` subnodes under that Functional Resource Type's node. For the Functional Resource Types listed above, all have functional resource type-specific parameters, some have functional resource type-specific events, and (as of this writing) none have functional resource type-specific directives.

The following sections identify the object identifiers associated with each of the Functional Resource Types listed above. Each section has a table identifying the names, descriptions, types and OIDs of the functional resource type-specific parameters for that Functional Resource Type that are used in the Operational Scenario in 2.5. The full set of parameters for these Functional Resource Types are found in the SANA registry

If the Functional Resource Type has functional resource type-specific events that are used in the Operational Scenario, the section contains a table identifying the names, descriptions,

types and OIDs of each functional resource type-specific event for that Functional Resource Type that is used in the scenario.

E2 ANTENNA

The Antenna Functional Resource Type is the first node under the `crossSupportFunctionalities` node:

```
{crossSupportFunctionalities  antenna (1)}
```

Thus the OID for the Antenna Functional Resource Type is (3/112/4/4/2/1/1).

The Antenna `parametersId` node is the first subnode of the `antenna` node. Thus the OID for the Antenna `parametersId` node is (3/112/4/4/2/1/1/1).

Table E-1 identifies the parameters of the Antenna Functional Resource Type used in the Operational Scenario and provides their descriptions, types, and OIDs (registered under the Antenna `parametersId` node).

The Antenna `eventsId` node is the second subnode of the `antenna` node. Thus the OID for the Antenna `eventsId` node is (3/112/4/4/2/1/1/2).

The Antenna Functional Resource Type has no functional resource type-specific events that were used in the Operational Scenario.

Table E-1: Antenna Functional Resource Type Parameters

Parameter	Description	Type	OID
`pointing-mode`	This enumerated parameter reports the pointing mode of the antenna servo system. The values this parameter can take on are: - 'stow' – the antenna is in or is moving to its stow position; - 'halt' – the antenna has been stopped in its current position; - 'slew' – the antenna is moving at commanded angular rates; - 'program-track' – the antenna is pointed in accordance with spacecraft trajectory predicts; - 'conical scan' – the antenna is performing a conical scan around the nominal pointing and applies offsets with respect to the predicts such that the observed signal strength is constant throughout the scan; - 'auto-track' – the antenna pointing is driven by a tracking receiver that by means of a suitable feed (e.g. monopulse) determines an error signal both for azimuth and elevation. Antenna implementations will typically support only a subset of the above listed pointing modes.	SEQUENCE (SIZE (1)) OF IntUnsigned (0 .. 5) { stow (0) , halt (1) , slew (2) , program-track (3) , conical-scan (4) , auto-track (5) }	3/112/4/4/2/1/1/1/9

E3 FORWARD SPACE LINK CARRIER TRANSMISSION

The Forward Space Link Carrier Transmission Functional Resource Type is the second node under the `crossSupportFunctionalities` node:

```
{crossSupportFunctionalities  fwdSpaceLinkCarrierTransmission (2)}
```

Thus the OID for the Forward Space Link Carrier Transmission Functional Resource Type is (3/112/4/4/2/1/2).

The Forward Space Link Carrier Transmission `parametersId` node is the first subnode of the `fwdSpaceLinkCarrierTransmission` node. Thus the OID for the Forward Space Link Carrier Transmission `parametersId` node is (3/112/4/4/2/1/2/1).

Table E-2 identifies the parameters of the Forward Space Link Carrier Transmission Functional Resource Type used in the Operational Scenario and provides their descriptions, types, and OIDs (registered under the Forward Space Link Carrier Transmission `parametersId` node).

The Forward Space Link Carrier Transmission `eventsId` node is the second subnode of the `fwdSpaceLinkCarrierTransmission` node. Thus the OID for the Forward Space Link Carrier Transmission `eventsId` node is (3/112/4/4/2/1/2/2).

The Forward Space Link Carrier Transmission Functional Resource Type has no functional resource type-specific events used in the operational scenario.

Table E-2: Forward Space Link Carrier Transmission Functional Resource Type Parameters

Parameter	Description	Type	OID
`actual-transmit-frequency`	This parameter reports the current forward link frequency in Hz. In general the frequency will be constant, except during the forward link sweep and for Category B missions in case the forward link is being ramped as to compensate for the Doppler shift and rate on the forward link.	SEQUENCE (SIZE (1)) OF IntUnsigned (2025000000 .. 40500000000)	3/112/4/4/2/1/2/1/5

E4 FORWARD SYMBOL STREAM TRANSMISSION

The Forward Symbol Stream Transmission Functional Resource Type is the fifth node under the `crossSupportFunctionalities` node:

```
{crossSupportFunctionalities  fwdSymbolStreamTransmission (5)}
```

Thus the OID for the Forward Symbol Stream Transmission Functional Resource Type is (3/112/4/4/2/1/5).

The Forward Symbol Stream Transmission `parametersId` node is the first subnode of the `fwdSymbolStreamTransmission` node. Thus the OID for the Forward Symbol Stream Transmission `parametersId` node is (3/112/4/4/2/1/5/1).

The Forward Symbol Stream Transmission Functional Resource Type has no monitored parameters that were used in the Operational Scenario.

The Forward Symbol Stream Transmission `eventsId` node is the second subnode of the `fwdSymbolStreamTransmission` node. Thus the OID for the Forward Symbol Stream Transmission `eventsId` node is (3/112/4/4/2/1/5/2).

The Forward Symbol Stream Transmission Functional Resource Type has no functional resource type-specific events that were used in the Operational Scenario.

E5 TC MC MULTIPLEXING, CHANNEL SYNCHRONIZATION, AND ENCODING

The TC MC Multiplexing, Channel Synchronization, and Encoding Functional Resource Type is the fiftieth node under the `crossSupportFunctionalities` node:

```
{crossSupportFunctionalities  fwdSymbolStreamTransmission (50)}
```

Thus the OID for the Forward Symbol Stream Transmission Functional Resource Type is (3/112/4/4/2/1/50).

The Forward Symbol Stream Transmission `parametersId` node is the first subnode of the `fwdSymbolStreamTransmission` node. Thus the OID for the Forward Symbol Stream Transmission `parametersId` node is (3/112/4/4/2/1/5/1).

The Forward Symbol Stream Transmission Functional Resource Type has no monitored parameters that were used in the Operational Scenario.

The Forward Symbol Stream Transmission `eventsId` node is the second subnode of the `fwdSymbolStreamTransmission` node. Thus the OID for the Forward Symbol Stream Transmission `eventsId` node is (3/112/4/4/2/1/5/2).

The Forward Symbol Stream Transmission Functional Resource Type has no functional resource type-specific events that were used in the Operational Scenario.

E6 FORWARD CLTU TRANSFER SERVICE PROVIDER

The Forward CLTU (FCLTU) Transfer Service (TS) Provider Functional Resource Type is the sixth node under the `crossSupportFunctionalities` node:

```
{crossSupportFunctionalities  fCltuTsProvider (6)}
```

Thus the OID for the FCLTU TS Provider Functional Resource Type is (3/112/4/4/2/1/6).

The FCLTU TS Provider `parametersId` node is the first subnode of the `fCltuTsProvider` node. Thus the OID for the FCLTU TS Provider `parametersId` node is (3/112/4/4/2/1/6/1).

Table E-3 identifies the parameters of the FCLTU TS Provider Functional Resource Type used in the Operational Scenario and provides their descriptions, types, and OIDs (registered under the FCLTU TS Provider `parametersId` node).

The FCLTU TS Provider `eventsId` node is the second subnode of the `fCltuTsProvider` node. Thus the OID for the FCLTU TS Provider `eventsId` node is (3/112/4/4/2/1/6/2).

Table E-4 identifies the events of the FCLTU TS Provider Functional Resource Type used in the Operational Scenario and provides their descriptions and OIDs (registered under the FCLTU TS Provider `eventsId` node).

Table E-3: FCLTU TS Provider Functional Resource Type Parameters

Parameter	Description	Type	OID
si-state	This enumerated parameter reports the status of the given instance of the F-CLTU service. It can take on the following values: - 'unbound' – all resources required to enable the provision of the Forward CLTU 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 F-CLTU transfer service provider port is not bound; - 'ready'– an association has been established between the user and the provider, and they may interact by means of the service operations. However, sending of CLTUs from the user to the provider (by means of the CLTU-TRANSFER-DATA operation) is not permitted; the user may enable the delivery of CLTUs by means of the appropriate service operation (CLTU-START), which, in turn, will cause the provider to transition to the state 'active'; - 'active' – this state resembles state 'ready', except that now the user can send CLTUs and the provider is enabled to radiate CLTUs to the spacecraft; the service continues in this state until the user invokes the CLTU-STOP operation to cause the provider to suspend transmission of CLTUs and transition back to state 'ready' or the PEER-ABORT invocation to cause the service to transition back to the 'unbound' state.	SEQUENCE (SIZE (1)) OF IntUnsigned (0 .. 2) { unbound (0) , ready (1) , active (2) }	3/112/4/4/2/1/6/1/8

Parameter	Description	Type	OID
`number-of-cltus-radiated`	This parameter reports the number of CLTUs that the provider successfully radiated completely during the service provision period. A CLTU in the process of being radiated is not included in this count.	SEQUENCE (SIZE (1)) OF IntUnsigned	3/112/4/4/2/1/6/1/23

Table E-4: FCLTU TS Provider Functional Resource Type Events

Parameter	Description	OID
`'production operational'`	The production process is ready to radiate CLTUs and production status has changed to 'operational'.	3/112/4/4/2/1/6/2/1
`'production interrupted'`	The production process has stopped due to a condition that may be temporary.	3/112/4/4/2/1/6/2/3
`'production halted'`	The production process has been stopped by management action.	3/112/4/4/2/1/6/2/3

E7 RETURN SPACE LINK CARRIER RECEPTION

The Return Space Link Carrier Reception Functional Resource Type is the eighth node under the `crossSupportFunctionalities` node:

```
{crossSupportFunctionalities  rtnSpaceLinkCarrierReception (8)}
```

Thus the OID for the Return Space Link Carrier Reception Functional Resource Type is (3/112/4/4/2/1/8).

The Return Space Link Carrier Reception `parametersId` node is the first subnode of the `rtnSpaceLinkCarrierReception` node. Thus the OID for the Return Space Link Carrier Reception `parametersId` node is (3/112/4/4/2/1/8/1).

Table E-5 identifies the parameters of the Return Space Link Carrier Reception Functional Resource Type used in the Operational Scenario and provides their descriptions, types, and OIDs (registered under the Return Space Link Carrier Reception `parametersId` node).

Table E-5: Return Space Link Carrier Reception Functional Resource Type Parameters

Parameter	Description	Type	OID
`actual-receive-frequency`	This parameter reports the observed carrier return link frequency in Hz. This parameter therefore varies with the Doppler shift induced by the radial velocity of the spacecraft relative to the ground based antenna. In 1-way mode, the Doppler shift applies only once, but also the onboard oscillator drift affects the observed return link carrier frequency. In 2-way mode in combination with a constant forward link frequency, the Doppler shift approximately doubles with respect to the 1-way case, but the contribution of the onboard oscillator drift is eliminated.	SEQUENCE (SIZE (1)) OF IntUnsigned (2199700000 .. 38500000000)	3/112/4/4/2/1/8/1/13

The Return Space Link Carrier Reception `eventsId` node is the second subnode of the `rtnSpaceLinkCarrierReception` node: Thus the OID for the Return Space Link Carrier Reception `eventsId` node is (3/112/4/4/2/1/8/2).

The Return Space Link Carrier Reception Functional Resource Type has no functional resource type-specific events that were used in the Operational Scenario.

E8 RETURN SPACE LINK SUBCARRIER RECEPTION

The Return Space Link Subcarrier Reception Functional Resource Type is the ninth node under the `crossSupportFunctionalities` node:

```
{crossSupportFunctionalities  rtnSpaceLinkSubcarrierReception (9)}
```

Thus the OID for the Return Space Link Subcarrier Reception Functional Resource Type is (3/112/4/4/2/1/9).

The Return Space Link Subcarrier Reception `parametersId` node is the first subnode of the `rtnSpaceLinkSubcarrierReception` node. Thus the OID for the Return Space Link Subcarrier Reception `parametersId` node is (3/112/4/4/2/1/9/1).

Table E-6 identifies the parameters of the Return Space Link Subcarrier Reception Functional Resource Type used in the Operational Scenario and provides their descriptions, types, and OIDs (registered under the Return Space Link Subcarrier Reception `parametersId` node).

Table E-6: Return Space Link Subcarrier Reception Functional Resource Type Parameters

Parameter	Description	Type	OID
`subcarrier-lock-status`	This enumerated parameter reports on the subcarrier lock status of the BPSK subcarrier demodulator. The values the parameter may have are: - 'locked' – the demodulator has locked on the return link subcarrier; - 'not-locked' – the demodulator has not locked on the return link subcarrier and therefore cannot deliver telemetry.	SEQUENCE (SIZE (1)) OF IntUnsigned (0 .. 1) { locked (0) , not-locked (1) }	3/112/4/4/2/1/9/1/3

Parameter	Description	Type	OID
actual-subcarrier-frequency	This parameter reports the actually received subcarrier frequency in 1/1000 Hz, i.e. this parameter reflects the Doppler shift of the subcarrier frequency. If the applicable modulation scheme does not use a subcarrier, this parameter shall be flagged as undefined.	SEQUENCE (SIZE (1)) OF IntUnsigned (2000 .. 300000)	3/112/4/4/2/1/9/1/4

The Return Space Link Subcarrier Reception `eventsId` node is the second subnode of the `rtnSpaceLinkSubcarrierReception` node. Thus the OID for the Return Space Link Subcarrier Reception `eventsId` node is (3/112/4/4/2/1/9/2).

E9 THE RETURN SPACE LINK SUBCARRIER RECEPTION FUNCTIONAL RESOURCE TYPE HAS NO FUNCTIONAL RESOURCE TYPE-SPECIFIC EVENTS THAT WERE USED IN THE OPERATIONAL SCENARIO.RETURN SYMBOL STREAM RECEPTION

The Return Symbol Stream Reception Functional Resource Type is the tenth node under the `crossSupportFunctionalities` node:

```
{crossSupportFunctionalities  rtnSymbolStreamReception (10)}
```

Thus the OID for the Return Symbol Stream Reception Functional Resource Type is (3/112/4/4/2/1/10).

The Return Symbol Stream Reception `parametersId` node is the first subnode of the `rtnSymbolStreamReception` node. Thus the OID for the Return Symbol Stream Reception `parametersId` node is (3/112/4/4/2/1/10/1).

Table E-7 identifies the parameters of the Return Symbol Stream Reception Functional Resource Type used in the Operational Scenario and provides their descriptions, types, and OIDs (registered under the Return Symbol Stream Reception `parametersId` node).

Table E-7: Return Symbol Stream Reception Functional Resource Type Parameters

Parameter	Description	Type	OID
symbol-synchronizer-lock-status	This enumerated parameter reports the symbol synchronizer lock status. The values the parameter may have are: - 'locked' – the symbol synchronizer has locked on the return link symbol stream; - 'not-locked' – the symbol synchronizer has not locked on the symbol stream.	SEQUENCE (SIZE (1)) OF IntUnsigned (0 .. 1) { locked (0) , not-locked (1) }	3/112/4/4/2/1/10/1/2

The Return Symbol Stream Reception `eventsId` node is the second subnode of the `rtnSymbolStreamReception` node. Thus the OID for the Return Symbol Stream Reception `eventsId` node is (3/112/4/4/2/1/10/2).

The Return Symbol Stream Reception Functional Resource Type has no functional resource type-specific events that were used in the Operational Scenario.

E10 RETURN TM SYNCHRONIZATION AND DECODING

The Return Telemetry (TM) Synchronization and Decoding Functional Resource Type is the eleventh node under the `crossSupportFunctionalities` node:

```
{crossSupportFunctionalities  rtnTmSyncAndDecode (11)}
```

Thus the OID for the Return TM Synchronization and Decoding Functional Resource Type is (3/112/4/4/2/1/11).

The Return TM Synchronization and Decoding `parametersId` node is the first subnode of the `rtnTmSyncAndDecode` node. Thus the OID for the Return TM Synchronization and Decoding `parametersId` node is (3/112/4/4/2/1/11/1).

The Return Symbol Stream Reception Functional Resource Type has no functional resource type-specific parameters that were used in the Operational Scenario.

The Return TM Synchronization and Decoding `eventsId` node is the second subnode of the `rtnTmSyncAndDecode` node. Thus the OID for the Return TM Synchronization and Decoding `eventsId` node is (3/112/4/4/2/1/11/2).

Table E-8 identifies the events of the Return TM Synchronization and Decoding Functional Resource Type used in the Operational Scenario and provides their descriptionsand OIDs (registered under the Return TM Synchronization and Decoding `eventsId` node).

Table E-8: Return TM Synchronization and Decoding Functional Resource Type Events

Parameter	Description	OID
`frameLock-Acquired`	The frame synchronizer has transitioned to the 'locked' status	3/112/4/4/2/1/11/2/1
`lossOfFrame-Lock`	The frame synchronizer has transitioned to the 'not locked' status	3/112/4/4/2/1/11/2/2

E11 RETURN ALL FRAMES TRANSFER SERVICE PROVIDER

The Return All Frames (RAF) Transfer Service (TS) Provider Functional Resource Type is the thirteenth node under the `crossSupportFunctionalities` node:

```
{crossSupportFunctionalities  rafTsProvider (13)}
```

Thus the OID for the RAF TS Provider Functional Resource Type is (3/112/4/4/2/1/13).

The RAF TS Provider `parametersId` node is the first subnode of the `rafTsProvider` node. Thus the OID for the RAF TS Provider `parametersId` node is (3/112/4/4/2/1/13/1).

Table E-9 identifies the parameters of the RAF TS Provider Functional Resource Type used in the Operational Scenario and provides their descriptions, types, and OIDs (registered under the RAF TS Provider `parametersId` node).

The RAF TS Provider `eventsId` node is the second subnode of the `rafTsProvider` node. Thus the OID for the RAF TS Provider `eventsId` node is (3/112/4/4/2/1/13/2).

Table E-10 identifies the events of the RAF TS Provider Functional Resource Type used in the Operational Scenario and provides their description sand OIDs (registered under the RAF TS Provider `eventsId` node).

Table E-9: RAF TS Provider Functional Resource Type Parameters

Parameter	Description	Type	OID
si-state	This enumerated parameter reports the status of the given instance of the RAF service. It can take on the following values: - 'unbound' – All resources required to enable the provision of the RAF 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 RAF transfer service provider port is not bound; - 'ready'– An association has been established between the user and the provider, and they may interact by means of the service operations. However, sending of telemetry frames from the provider to the user (by means of the RAF-TRANSFER-DATA operation) is not permitted; the user may enable the delivery of telemetry frames by means of the appropriate service operation (RAF-START), which, in turn, will cause the provider to transition to the state 'active'; - 'active' – This state resembles state 'ready', except that now the provider will send telemetry frames provided frames of the selected characteristics are made available by the RAF production process; the service continues in this state until the user invokes the RAF-STOP operation to cause the provider to suspend delivery of telemetry frames and transition back to state 'ready' or the PEER-ABORT invocation to cause the service to transition back to the 'unbound' state.	SEQUENCE (SIZE (1)) OF IntUnsigned (0 .. 2) { unbound (0) , ready (1) , active (2) }	3/112/4/4/2/1/13/1/7

Parameter	Description	Type	OID
number-of-frames-delivered	This parameter reports the total number of telemetry frames that were delivered to the user since the start of the service instance provision period.	SEQUENCE (SIZE (1)) OF IntUnsigned	3/112/4/4/2/1/13/1/12

Table E-10: RAF TS Provider Functional Resource Type Events

Parameter	Description	OID
'production running'	The RAF production process is capable of processing a return space link physical channel, if available	3/112/4/4/2/1/13/2/1
'production halted'	The RAF production process is stopped and production equipment is out of service because of management action.	3/112/4/4/2/1/13/2/2
'production interrupted'	The RAF production process is stopped because of a fault.	3/112/4/4/2/1/13/2/3

E12 MD-CSTS PROVIDER

The MD-CSTS Provider Functional Resource Type is the seventeenth node under the `crossSupportFunctionalities` node:

```
{crossSupportFunctionalities  mdCstsProvider (17)}
```

Thus the OID for the MD-CSTS Provider Functional Resource Type is (3/112/4/4/2/1/17).

The MD-CSTS Provider Functional Resource Type has no functional resource type-specific parameters or events that are used in the Operational Scenario. However, it does have four CSTS *service-generic* events that are used in the Operational Scenario.

The CSTS Specification Framework (reference [1]) defines certain service-generic parameters and events that are available for use by CSTSes, which in Functional resource terms means that these service-generic parameters and events are available to the MD-CSTS via the CSTS Provider Functional Resource Types for those CSTSes, These CSTS Provider service-generic parameters and events are registered under the `serviceGenericId` subnode of the (CSTS) `framework` node:

```
{iso identified-organization (3)  standards-producing-organization(112)
ccsds(4) css (4) csts (1) framework (1) serviceGenericFunctionalities (6)  serviceGenericId (1)}
```

The service-generic functional resource parameters are registered under the `svcParametersId` subnode of `serviceGenericId`:

```
{serviceGenericId  svcParametersId (1)}
```

The service-generic functional resource events are registered under the `svcEventsId` subnode of `serviceGenericId`:

```
{serviceGenericId  svcEventsId (2)}
```

The MD-CSTS Provider Functional Resource Type implements the `svcProductionConfigured`, `svcProductionInterrupted`, `svcProductionHalted`, `svcProductionOperational`, and `svcProductionConfigurationChange` service-generic functional resource events. The descriptions, and OIDs of these service-generic events can be found in reference [1].

ANNEX F

INFORMATIVE REFERENCES

(INFORMATIVE)

[E1] *Cross Support Concept—Part 1: Space Link Extension Services*. Report Concerning Space Data Systems Standards, CCSDS 910.3-G-2. Green Book. Issue 2. Washington, D.C.: CCSDS, July 2002.

[E2] *Space Link Extension – Internet Protocol for Transfer Services.* Recommended Standard, CCSDS 913.1-B-1. Blue Book. September 2008.

[E3] *Cross Support Transfer Services Specification Framework Concepts.* TBS.

[E4] *Space Link Extension – Forward CLTU Service Specification*. Recommended Standard, CCSDS 912.1-B-3. Blue Book. July 2010.

[E5] *Space Link Extension – Return All Frames Service Specification*. Recommended Standard, CCSDS 911.1-B-3. Blue Book. January 2010.

[E6] *Space Communication Cross Support Service Management Service Specification*. Recommended Standard, CCSDS 910.11-B-1. Blue Book. August 2009.

ANNEX G

ACRONYMS

(INFORMATIVE)

BPSK	Binary Phase Shift Keying
CCSDS	Consultative Committee for Space Data Systems
CLTU	Communication Link Transmission Unit
CM	Complex Management
CSTS	Cross Support Transfer Service
CSTS SFW	CSTS Specification Framework (specification)
FCLTU	Forward CLTU
ISO	International Organization for Standardization
MD-CSTS	Monitored Data Cross Support Transfer Service
MDOS	Mission Data Operation System
MUE	Mission User Entity
OID	Object Identifier
QPSK	Quadrature Phase Shift Keying
RAF	Return All Frames
RF	Radio Frequency
SANA	Space Assigned Numbers Authority
SLE	Space Link Extension
UM	Utilization Management

ANNEX H

BOOKMARKS

(THIS ANNEX IS AN ARTIFACT OF CONSTRUCTING THIS RECOMMENDED STANDARD AND WILL BE REMOVED PRIOR TO PUBLICATION)

1 Ref_CSTS_SFW

2 Ref_CSRM