BS ISO 18445:2016

BSI Standards Publication

Space data and information transfer systems — Space link extension — Application program interface for the forward CLTU service

Documents
Document Sharing Hub

National foreword

This British Standard is the UK implementation of ISO 18445:2016. It supersedes BS ISO 18445:2013 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee ACE/68, Space systems and operations.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2016.
Published by BSI Standards Limited 2016

ISBN 978 0 580 93383 7

ICS 49.140

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2016.

Amendments/Corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO 18445:2016

INTERNATIONAL STANDARD

ISO 18445

Second edition
2016-11-15

Space data and information transfer systems — Space link extension — Application program interface for the forward CLTU service

Systèmes de transfert des informations et données spatiales — Extension de liaisons spatiales — Interface du programme d'application pour le service de l'unité de transmission pour la liaison d'envoi de télécommande (CLTU)

Reference number
ISO 18445:2016(E)

Documents
Document Sharing Hub

© ISO 2016

Full document details: BS ISO 18445:2016

COPYRIGHT PROTECTED DOCUMENT

© ISO 2016

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
copyright@iso.org
Web www.iso.org

Published in Switzerland

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 18445:2016

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2. www.iso.org/directives

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received. www.iso.org/patents

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

ISO 18445 was prepared by the Consultative Committee for Space Data Systems (CCSDS) (as CCSDS 916.1-M-2, September 2015) and was adopted (without modifications except those stated in clause 2 of this International Standard) by Technical Committee ISO/TC 20, *Aircraft and space vehicles*, Subcommittee SC 13, *Space data and information transfer systems*.

This second edition cancels and replaces the first edition (ISO 18445:2013), which has been technically revised.

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 18445:2016

STATEMENT OF INTENT

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

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

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

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

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

FOREWORD

This document is a technical **Recommended Practice** for use in developing ground systems for space missions and has been prepared by the **Consultative Committee for Space Data Systems** (CCSDS). The Application Program Interface described herein is intended for missions that are cross-supported between Agencies of the CCSDS.

This **Recommended Practice** specifies service type-specific extensions of the Space Link Extension Application Program Interface for Transfer Services specified by CCSDS (reference [5]). It allows implementing organizations within each Agency to proceed with the development of compatible, derived Standards for the ground systems that are within their cognizance. Derived Agency Standards may implement only a subset of the optional features allowed by the **Recommended Practice** and may incorporate features not addressed by the **Recommended Practice**.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS has processes for identifying patent issues and for securing from the patent holder agreement that all licensing policies are reasonable and non-discriminatory. However, CCSDS does not have a patent law staff, and 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 *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

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

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

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

Member Agencies

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

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- 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.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 916.1-M-1	Space Link Extension—Application Program Interface for the Forward CLTU Service, Recommended Practice, Issue 1	October 2008	Original issue, superseded
CCSDS 916.1-M-2	Space Link Extension—Application Program Interface for the Forward CLTU Service, Recommended Practice, Issue 2	September 2015	Current issue: – updates text to accommodate changes in current version of SLE service specification; – differentiates applicability by SLE service specification version; – updates references.

NOTE – Substantive changes from the previous issue are marked with change bars in the inside margin.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

CONTENTS

Section Page

1 INTRODUCTION 1-1

1.1 PURPOSE 1-1
1.2 SCOPE 1-1
1.3 APPLICABILITY 1-2
1.4 RATIONALE 1-2
1.5 DOCUMENT STRUCTURE 1-3
1.6 DEFINITIONS, NOMENCLATURE, AND CONVENTIONS 1-5
1.7 REFERENCES 1-8

2 OVERVIEW 2-1

2.1 INTRODUCTION 2-1
2.2 PACKAGE CLTU SERVICE INSTANCES 2-1
2.3 PACKAGE CLTU OPERATIONS 2-13
2.4 SECURITY ASPECTS OF THE SLE FORWARD CLTU TRANSFER SERVICE 2-15

3 CLTU SPECIFIC SPECIFICATIONS FOR API COMPONENTS 3-1

3.1 API SERVICE ELEMENT 3-1
3.2 SLE OPERATIONS 3-15
3.3 SLE APPLICATION 3-15

ANNEX A CLTU SPECIFIC INTERFACES (NORMATIVE) A-1
ANNEX B ACRONYMS (INFORMATIVE) B-1
ANNEX C INFORMATIVE REFERENCES (INFORMATIVE) C-1

Figure

1-1 SLE Services and SLE API Documentation 1-4
2-1 CLTU Service Instances 2-2
2-2 CLTU Operation Objects 2-14

Table

2-1 Production Events Reported via the Interface ICLTU_SIUpdate 2-5
2-2 CLTU Configuration Parameters Handled by the Service Element 2-7
2-3 CLTU Status Parameters Handled by the Service Element 2-9
2-4 CLTU Production Status 2-10
2-5 Mapping of CLTU Operations to Operation Object Interfaces 2-15

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

© ISO 2016 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 18445:2016

Documents
Document Sharing Hub

© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

1 INTRODUCTION

1.1 PURPOSE

The Recommended Practice *Space Link Extension—Application Program Interface for Transfer Services—Core Specification* (reference [5]) specifies a C++ API for CCSDS Space Link Extension Transfer Services. The API is intended for use by application programs implementing SLE transfer services.

Reference [5] defines the architecture of the API and the functionality on a generic level, which is independent of specific SLE services and communication technologies. It is thus necessary to add service type-specific specifications in supplemental Recommended Practices. The purpose of this document is to specify extensions to the API needed for support of the Command Link Transmission Unit (CLTU) service defined in reference [4].

1.2 SCOPE

This Recommended Practice defines extensions to the SLE API in terms of:

a) the CLTU-specific functionality provided by API components;

b) the CLTU-specific interfaces provided by API components; and

c) the externally visible behavior associated with the CLTU interfaces exported by the components.

It does not specify

a) individual implementations or products;

b) the internal design of the components; and

c) the technology used for communications.

This Recommended Practice defines only interfaces and behavior that must be provided by implementations supporting the forward CLTU service in addition to the specification in reference [5].

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

1.3 APPLICABILITY

The Application Program Interface specified in this document supports three generations of Forward CLTU service, namely:

a) Generation 1 identified by the version number 1 in the BIND operation, as specified by reference [C2];

b) Generation 2 identified by the version number 2 in the BIND operation, as specified by reference [C3];

c) Generation 3 identified by the version number 4 in the BIND operation, as specified by reference [4].

NOTE – The use of the term 'Generation' follows the definition in the API Core Specification (reference [5]) where it is used to classify all SLE Transfer Services.

Support for Generation 1 and Generation 2 of this service is included for backward compatibility purposes for a limited time and may not be continued in future versions of this specification. Support for Generation 1 (i.e., version 1 of the CLTU service) implies that SLE API implementations of this specification are able to interoperate with peer SLE systems that comply with the specification of the Transport Mapping Layer (TML) in 'Specification of a SLE API Proxy for TCP/IP and ASN.1', ESOC, SLES-SW-API-0002-TOS-GCI, Issue 1.1, February 2001. For Generation 2 and 3 of these services, SLE API implementations of this specification are able to interoperate with peer SLE systems that comply with the specification of the Transport Mapping Layer (TML) in reference [C6].

Provisions within this Recommended Practice that are specific for one or more generations are marked as follows:

– [Gn:] for provisions specific to Generation n;

– [Gn,m:] for provisions specific to Generation n and Generation m.

Provisions that apply to all generations are not marked.

1.4 RATIONALE

This Recommended Practice specifies the mapping of the forward CLTU service specification to specific functions and parameters of the SLE API. It also specifies the distribution of responsibility for specific functions between SLE API software and application software.

The goal of this Recommended Practice is to create a standard for interoperability between:

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

a) application software using the SLE API and SLE API software implementing the SLE API; and

b) SLE user and SLE provider applications communicating with each other using the SLE API on both.

This interoperability standard also allows exchangeability of different products implementing the SLE API, as long as they adhere to the interface specification of this Recommended Practice.

1.5 DOCUMENT STRUCTURE

1.5.1 ORGANIZATION

This document is organized as follows:

– section 1 provides purpose and scope of this specification, identifies conventions, and lists definitions and references used throughout the document;

– section 2 provides an overview of the CLTU service and describes the API model extension including support for the CLTU service;

– section 3 contains detailed specifications for the API components and for applications using the API;

– annex A provides a formal specification of the API interfaces and data types specific to the CLTU service;

– annex B lists all acronyms used within this document;

– annex C lists informative references.

1.5.2 SLE SERVICE DOCUMENTATION TREE

The SLE suite of Recommended Standards is based on the cross support model defined in the SLE Reference Model (reference [3]). The services defined by the reference model constitute one of the three types of Cross Support Services:

a) Part 1: SLE Services;

b) Part 2: Ground Domain Services; and

c) Part 3: Ground Communications Services.

The SLE services are further divided into SLE service management and SLE transfer services.

The basic organization of the SLE services and SLE documentation is shown in figure 1-1. The various documents are described in the following paragraphs.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

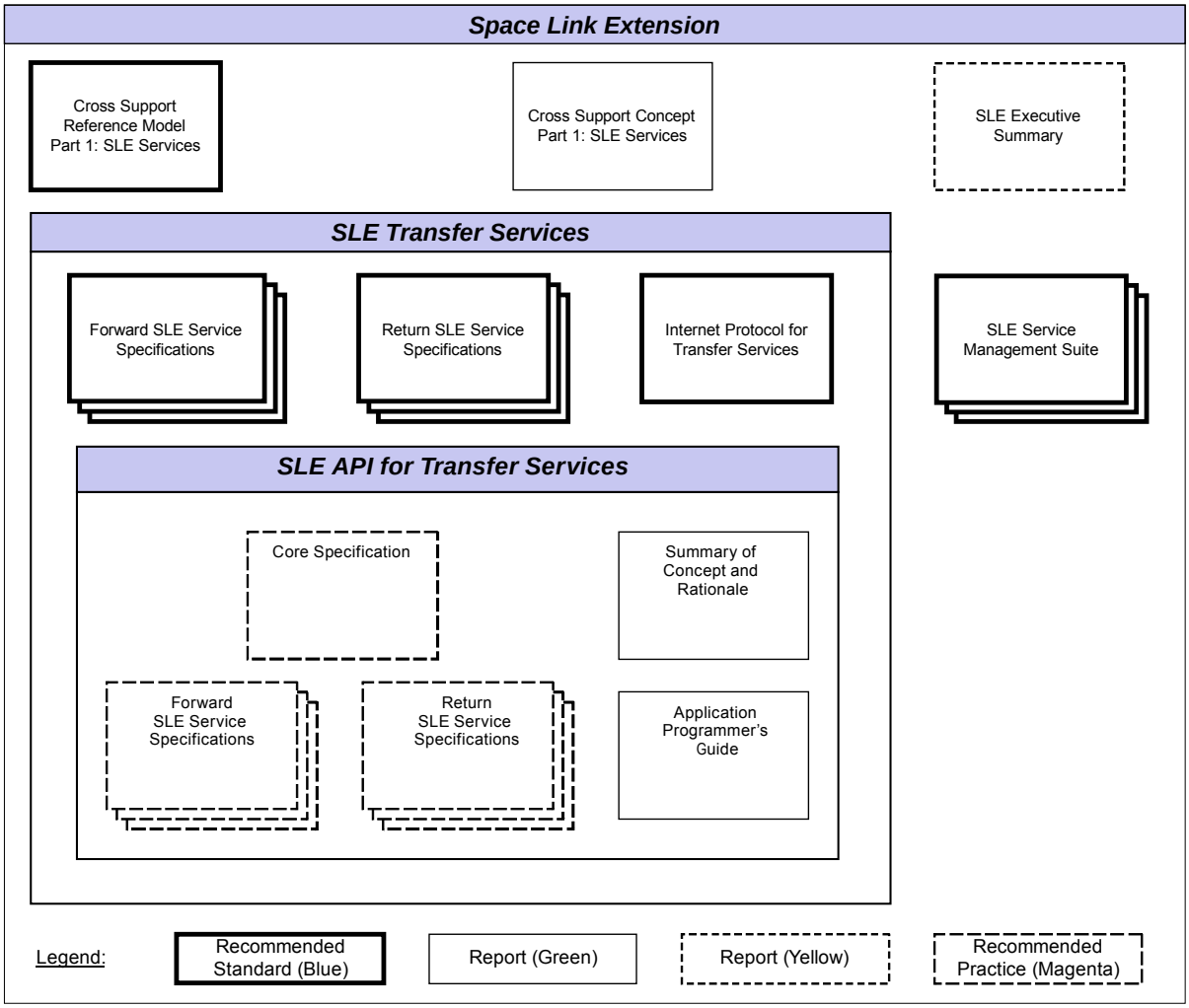

Figure 1-1: SLE Services and SLE API Documentation

a) *Cross Support Reference Model—Part 1: Space Link Extension Services*, a Recommended Standard that defines the framework and terminology for the specification of SLE services.

b) *Cross Support Concept—Part 1: Space Link Extension Services*, a Report introducing the concepts of cross support and the SLE services.

c) *Space Link Extension Services—Executive Summary*, an Administrative Report providing an overview of Space Link Extension (SLE) Services. It is designed to assist readers with their review of existing and future SLE documentation.

d) *Forward SLE Service Specifications*, a set of Recommended Standards that provide specifications of all forward link SLE services.

e) *Return SLE Service Specifications*, a set of Recommended Standards that provide specifications of all return link SLE services.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

f) *Internet Protocol for Transfer Services*, a Recommended Standard providing the specification of the wire protocol used for SLE transfer services.

g) *SLE Service Management Specifications*, a set of Recommended Standards that establish the basis of SLE service management.

h) *Application Program Interface for Transfer Services—Core Specification*, a Recommended Practice document specifying the generic part of the API for SLE transfer services.

i) *Application Program Interface for Transfer Services—Summary of Concept and Rationale*, a Report describing the concept and rationale for specification and implementation of a Application Program Interface for SLE Transfer Services.

j) *Application Program Interface for Return Services*, a set of Recommended Practice documents specifying the service type-specific extensions of the API for return link SLE services.

k) *Application Program Interface for Forward Services*, a set of Recommended Practice documents specifying the service type-specific extensions of the API for forward link SLE services.

l) *Application Program Interface for Transfer Services—Application Programmer's Guide*, a Report containing guidance material and software source code examples for software developers using the API.

1.6 DEFINITIONS, NOMENCLATURE, AND CONVENTIONS

1.6.1 DEFINITIONS

1.6.1.1 Definitions from Telecommand Channel Service

This Recommended Practice makes use of the following terms defined in reference [1]:

a) Command Link Transmission Unit (CLTU);

b) Physical Layer Operations Procedure (PLOP).

1.6.1.2 Definitions from Telecommand Data Routing Service

This Recommended Practice makes use of the following terms defined in reference [2]:

Command Link Control Word (CLCW).

1.6.1.3 Definitions from SLE Reference Model

This Recommended Practice makes use of the following terms defined in reference [3]:

a) Forward CLTU service;

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

b) operation;

c) service provider (provider);

d) service user (user);

e) SLE transfer service instance;

f) SLE transfer service production;

g) SLE transfer service provision;

h) space link data unit (SL-DU).

1.6.1.4 Definitions from CLTU Service

This Recommended Practice makes use of the following terms defined in reference [4]:

a) association;

b) communications service;

c) confirmed operation;

d) invocation;

e) parameter;

f) performance;

g) port identifier;

h) return;

i) service instance provision period;

j) unconfirmed operation.

1.6.1.5 Definitions from ASN.1 Specification

This Recommended Practice makes use of the following terms defined in reference [7]:

a) Object Identifier;

b) Octet String.

1.6.1.6 Definitions from UML Specification

This Recommended Practice makes use of the following terms defined in reference [C9]:

a) Attribute;

b) Base Class;

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

c) Class;

d) Data Type;

e) Interface;

f) Method.

1.6.1.7 Definitions from API Core Specification

This Recommended Practice makes use of the following terms defined in reference [5]:

a) Application Program Interface;

b) Component.

1.6.2 NOMENCLATURE

1.6.2.1 Normative Text

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

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

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

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

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

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

1.6.2.2 Informative Text

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

– Overview;

– Background;

– Rationale;

– Discussion.

© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

1.6.3 CONVENTIONS

This document applies the conventions defined in reference [5].

The model extensions in section 2 are presented using the Unified Modeling Language (UML) and applying the conventions defined in reference [5].

The CLTU-specific specifications for API components in section 3 are presented using the conventions specified in reference [5].

The CLTU-specific interfaces in annex A are specified using the conventions defined in reference [5].

1.7 REFERENCES

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

NOTE – A list of informative references is provided in annex C.

[1] *TC Synchronization and Channel Coding*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 231.0-B-2. Washington, D.C.: CCSDS, September 2010.

[2] *TC Space Data Link Protocol*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.0-B-2. Washington, D.C.: CCSDS, September 2010.

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

[4] *Space Link Extension—Forward CLTU Service Specification*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 912.1-B-3. Washington, D.C.: CCSDS, July 2010.

[5] *Space Link Extension—Application Program Interface for Transfer Services—Core Specification*. Issue 2. Recommendation for Space Data System Practices (Magenta Book), CCSDS 914.0-M-2. Washington, D.C.: CCSDS, September 2015.

[6] *Programming Languages—C++*. 3rd ed. International Standard, ISO/IEC 14882:2011. Geneva: ISO, 2011.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

[7] *Information Technology—Abstract Syntax Notation One (ASN.1): Specification of Basic Notation*. 4th ed. International Standard, ISO/IEC 8824-1:2008. Geneva: ISO, 2008.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

Documents
Document Sharing Hub

© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

2 OVERVIEW

2.1 INTRODUCTION

This section describes the extension of the SLE API model in reference [5] for support of the CLTU service. Extensions are needed for the API components API Service Element and SLE Operations.

In addition to the extensions defined in this section, the component API Proxy must support encoding and decoding of CLTU-specific protocol data units.

2.2 PACKAGE CLTU SERVICE INSTANCES

2.2.1 OVERVIEW

The CLTU extensions to the component API Service Element are defined by the package CLTU Service Instances. Figure 2-1 provides an overview of this package. The diagram includes classes from the package API Service Element specified in reference [5], which provide applicable specifications for the CLTU service.

The package adds two service instance classes:

a) CLTU SI User, supporting the service user role; and

b) CLTU SI Provider, supporting service provider role.

These classes correspond to the placeholder classes I<SRV>_SI User and I<SRV>_SI Provider defined in reference [5].

Both classes are able to handle the specific CLTU operations.

For the class CLTU SI User, this is the only extension of the base class SI User.

The class CLTU SI Provider adds two new interfaces:

a) `ICLTU_SIAdmin` by which the application can set CLTU-specific configuration parameters; and

b) `ICLTU_SIUpdate` by which the application must update dynamic status information, required for generation of status reports.

These interfaces correspond to the placeholder interfaces `I<SRV>_SIAdmin` and `I<SRV>_SIUpdate` defined in reference [5].

© ISO 2016 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 18445:2016

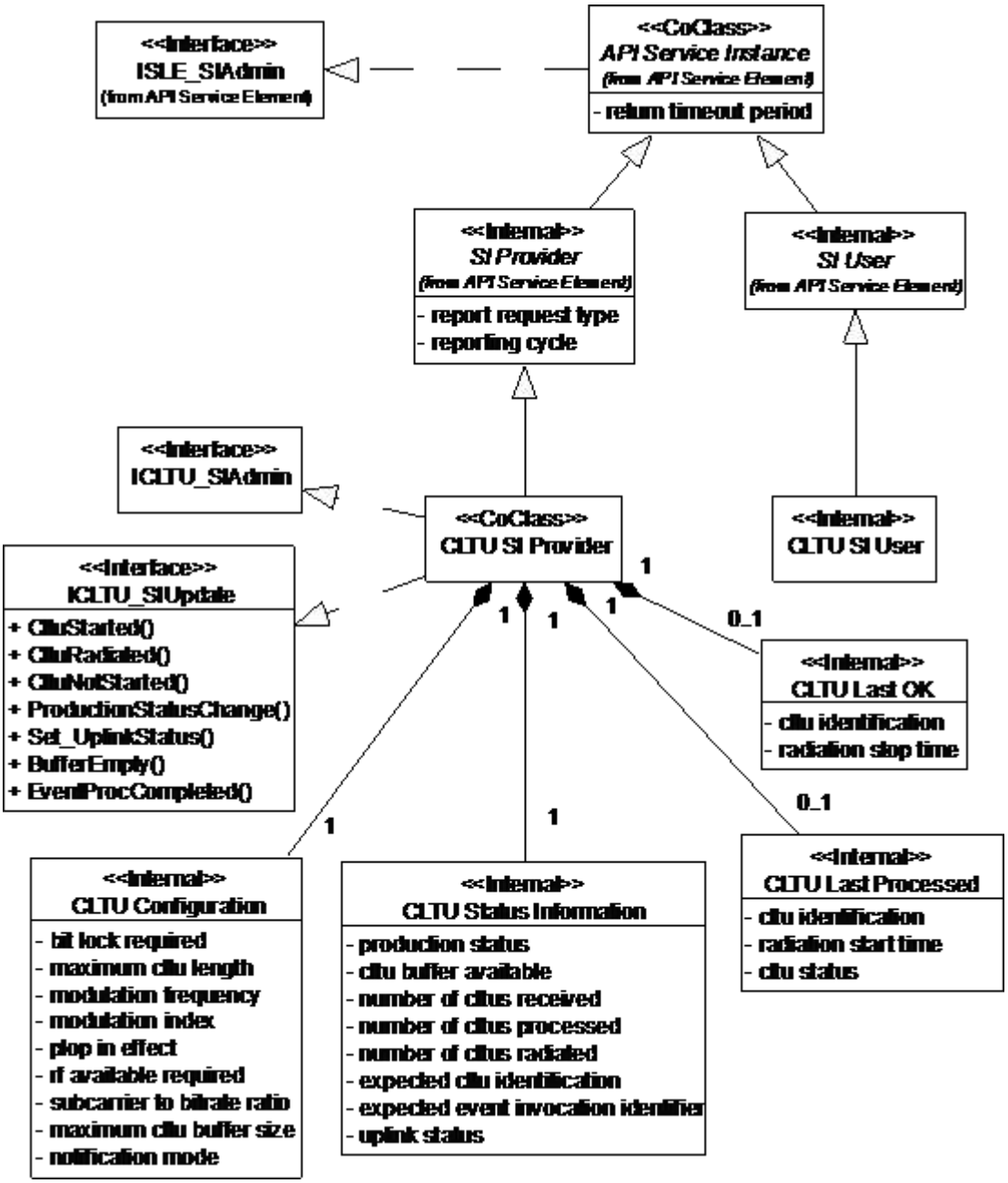

Figure 2-1: CLTU Service Instances

CLTU-specific configuration parameters are defined by the internal class CLTU Configuration. The class CLTU Status Information defines dynamic status parameters maintained by the service instance. In addition, the service instance maintains a set of parameters for the last CLTU processed and for the last CLTU that was successfully radiated. These parameters are defined by the classes CLTU Last Processed and CLTU Last OK.

Although the CLTU service allows only a single service instance to be bound to a service provider at any point of time, the service element does not constrain the number of CLTU

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

service instances on the user side or the provider side. More than one service instance might be needed for back-up purposes. In addition, this specification does not exclude that a single service element be used to serve several CLTU production engines or to connect to several providers. Therefore, the service element shall not enforce that only a single CLTU service instance is bound.

All specifications provided in this section refer to a single service instance. If more than one service instance is used, each service instance must be configured and updated independently.

2.2.2 COMPONENT CLASS CLTU SI USER

The class defines a CLTU service instance supporting the service user role. It ensures that SLE PDUs passed by the application and by the association are supported by the CLTU service and handles the CLTU operation objects defined in 2.3. It does not add further features to those provided by the base class SI User.

2.2.3 COMPONENT CLASS CLTU SI PROVIDER

2.2.3.1 General

The class defines a CLTU service instance supporting the service provider role. It exports the interfaces `ICLTU_SIAdmin` for configuration of the service instance after creation and `ICLTU_SIUpdate` for update of dynamic status parameters during operation.

2.2.3.2 Responsibilities

2.2.3.2.1 Service Specific Configuration

The service instance implements the interface `ICLTU_SIAdmin` to set the CLTU-specific configuration parameters defined by the class CLTU Configuration. The methods of this interface must be called after creation of the service instance. When all configuration parameters (including those set via the interface `ISLE_SIAdmin`) have been set, the method `ISLE_SIAdmin::ConfigCompleted()` must be called. This method verifies that all configuration parameters values are defined and are in the range defined in reference [4].

In addition, the interface `ICLTU_SIAdmin` provides read access to the configuration parameters.

2.2.3.2.2 Update of Dynamic Status Parameters

The class implements the interface `ICLTU_SIUpdate` to inform the service instance of specific events in the CLTU production process. The methods of this interface update status parameters defined by the classes CLTU Status Information, CLTU Last Processed, and CLTU Last OK. The events reported via `ICLTU_SIUpdate` and the parameters updated via this interface are listed in table 2-1.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

In order to ensure that the status information is always up to date the events listed in table 2-1 must be reported to the service instance during its complete lifetime, independent of the state of the service instance.

In addition, the class derives some of the parameters in CLTU Status Information from CLTU PDUs exchanged between the service user and the service provider. The methods used to update each of the parameters are defined in 2.2.5.

The interface `ICLTU_SIUpdate` provides read access to all status parameters.

2.2.3.2.3 Generation of Notifications

If events reported via the interface `ICLTU_SIUpdate` require that a CLTU–ASYNC–NOTIFY invocation be sent to the service user, the class generates and transmits these invocations if that is requested by the application and if the state of the service instance is 'active' or 'ready'. The notifications that are generated and transmitted by the class are listed in table 2-1.

The application can opt not to use this feature, but to generate the notification itself and transmit it using the interface `ISLE_ServiceInitiate`. It is noted that reference [4] defines additional notifications that must always be generated and transmitted by the application.

The SLE API supports different modes for generation of notifications. In 'deferred' notification mode, if no CLTU is affected and the production status changes to 'interrupted'; the notification is deferred until the attempt is made to radiate the next CLTU. In 'immediate' notification mode, the 'production interrupted' notification is generated immediately.

2.2.3.2.4 Handling of the CLTU–GET-PARAMETER Operation

The class responds autonomously to CLTU–GET–PARAMETER invocations. It generates the appropriate CLTU–GET–PARAMETER return using the parameters maintained by the classes CLTU Configuration and CLTU Status Information.

2.2.3.2.5 Status Reporting

The class generates CLTU–STATUS–REPORT invocations when required using the parameters maintained by the classes CLTU Status Information and CLTU Information.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

Table 2-1: Production Events Reported via the Interface ICLTU_SIUpdate

NOTE – The notification type actually transmitted depends on the method arguments and partially or the value of the production status.

Event	Method	Arguments	Status parameters updated	Notification sent
Radiation of a CLTU started.	`CltuStarted`	CLTU identification radiation start time available buffer size	CLTU identification last processed radiation start time CLTU status number of CLTUs processed available buffer size	none
Radiation of a CLTU completed.	`CltuRadiated`	radiation stop time radiation start time (see note below)	CLTU identification last OK radiation stop time CLTU status number of CLTUs radiated	CLTU radiated
Radiation of a CLTU could not be started because the latest radiation time expired or the production status was interrupted.	`CltuNotStarted`	CLTU identification failure reason available buffer size	CLTU identification last processed radiation start time CLTU status number of CLTUs processed available buffer size	SLDU expired production interrupted
The CLTU buffer is empty.	`BufferEmpty`		available buffer size	buffer empty
The production status changed with or without affecting a CLTU being radiated.	`ProductionStatusChange`	production status available buffer size	available buffer size production status CLTU status	production operational production interrupted production halted
Processing of a thrown event completed	`EventProcCompleted`	event id event processing result		action list completed action list not completed event condition evaluated to false
The uplink status changed	`Set_UplinkStatus`	uplink status	uplink status	none

Documents
Document Sharing Hub

© ISO 2016 – All rights reserved

Full document details: BS ISO 18445:2016

NOTE – for the method `CltuRadiated` the start time is an optional parameter that can be supplied if the exact start time is known only after radiation of the CLTU. In such a case the start time passed to the method `CltuStarted` should be the best available estimate.

2.2.3.2.6 Processing of CLTU Protocol Data Units

The class ensures that SLE PDUs passed by the application and by the association are supported by the CLTU service and handles the CLTU operation objects defined in 2.3.

2.2.3.2.7 Processing of CLTU–TRANSFER–DATA Invocations

For incoming CLTU–TRANSFER–DATA invocations the class performs the following checks in addition to those defined in [5]:

a) if the 'earliest radiation time' and the 'latest radiation time' are both specified, the 'earliest radiation time' must not be later than the 'latest radiation time';

b) the size of the CLTU contained in the PDU must not be larger than the value of the configuration parameter 'maximum-sldu-length' allows.

In contrast to handling of other confirmed operations, the service instance is allowed to pass the operation object to the application after setting the correct diagnostic if these checks fail. The application is expected to insert the next expected CLTU identification and the available buffer size into the operation object and pass it back to service instance via the interface `ISLE_ServiceInitiate`. The reasons for this specification are explained in 2.2.8.3.

2.2.3.2.8 Processing of CLTU–THROW-EVENT invocations

In contrast to handling of other confirmed operations, the service instance is allowed to pass the operation object to the application after setting the correct diagnostic if checks performed by the service element fail. The application is expected to insert the next expected event invocation identifier into the operation object and pass it back to service instance via the interface `ISLE_ServiceInitiate`. The reasons for this specification are explained in 2.2.8.3.

2.2.4 INTERNAL CLASS CLTU CONFIGURATION

The class defines the configuration parameters that can be set via the interface `ICLTU_SIAdmin`. These parameters are defined by reference [4]. Table 2-2 describes how the service instance uses these parameters. The column labeled 'Upd' indicates whether an update of these parameters is allowed after the initial configuration has been completed. It is

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

noted that an update might be inhibited by service management also when an update is possible according to the table.

Table 2-2: CLTU Configuration Parameters Handled by the Service Element

Parameter	Used for	Upd
[G3:] `acquisition-sequence-length`	CLTU–GET–PARAMETER	Y
`bit-lock-required`	CLTU–GET–PARAMETER	Y
[G3:] `clcw-global-VCID`	CLTU–GET–PARAMETER	Y
[G3:] `clcw-physical-channel`	CLTU–GET–PARAMETER	Y
`maximum-cltu-buffer-size`	value of the status parameter CLTU buffer available after configuration, CLTU-STOP, CLTU-PEER-ABORT, and protocol abort	N
`maximum-cltu-length`	CLTU–GET–PARAMETER	Y
[G3:] `minimum-delay-time`	CLTU–GET–PARAMETER	Y
`modulation-frequency`	CLTU–GET–PARAMETER	Y
`modulation-index`	CLTU–GET–PARAMETER [G2,3:] the value of the modulation index in milliradians [G1:] the amount of carrier suppression in 1/100 dB	Y
`notification-mode`	value of the notification mode, either 'immediate' or 'deferred' used to control invocation of notifications; also reported in CLTU–GET–PARAMETER	N
[G3:] `plop-1-idle-sequence-length`	CLTU–GET–PARAMETER	Y
`plop-in-effect`	CLTU–GET–PARAMETER	Y
[G3:] `protocol-abort-mode`	CLTU–GET–PARAMETER	Y
`rf-available-required`	CLTU–GET–PARAMETER	Y
`subcarrier-to-bitrate-ratio`	CLTU–GET–PARAMETER	Y

2.2.5 INTERNAL CLASS CLTU STATUS INFORMATION

The class defines global status parameters handled by the service instance. The parameters are defined by reference [4]. Table 2-3 describes how the service element updates each of the parameters and how it uses the parameters.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

2.2.6 INTERNAL CLASS CLTU LAST PROCESSED

The class defines the parameters maintained by the service instance for the last CLTU for which radiation started or radiation was attempted. These parameters are defined in reference [4].

All parameters are set via methods in the interface `ICLTU_SIUpdate` (see table 2-1) and are used in status reports and notifications.

© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

Table 2-3: CLTU Status Parameters Handled by the Service Element

Parameter	Update	Used for
`production-status`	– set by methods of `ICLTU_SIUpdate`	status reports notifications
`cltu-buffer-available`	– set to maximum at configuration time – set by methods of `ICLTU_SIUpdate` – extracted from CLTU-TRANSFER-DATA returns – reset to maximum following CLTU–STOP, CLTU-PEER–ABORT and protocol abort	status reports notifications
`number-of-cltus-received`	– set to zero at configuration time – incremented for every CLTU-TRANSFER-DATA return with a positive result	status reports
`number-of-cltus-processed`	– set to zero at configuration time – incremented with every call to `CltuStarted()` and `CltuNotStarted()`	status reports
`number-of-cltus-radiated`	– set to zero at configuration time – incremented with every call to `CltuRadiated()`	status reports
`expected-cltu-identification`	– set to zero at configuration time – [G1:] extracted from CLTU-START invocations if the application transmits a return with a positive result and the parameter first-cltu-identification is used. – [G2,3:] extracted from CLTU-START invocations if the application transmits a return with a positive result. – extracted from CLTU-TRANSFER-DATA returns	CLTU–GET–PARAMETER
`expected-event-invocation-identifier`	– set to zero at configuration time – extracted from CLTU-THROW-EVENT returns	CLTU–GET–PARAMETER
`uplink-status`	– set by methods of `ICLTU_SIUpdate`	status reports notifications

© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

2.2.7 INTERNAL CLASS CLTU LAST OK

The class defines the parameters maintained by the service instance for the last CLTU for which radiation was completed. These parameters are defined in reference [4].

All parameters are set via methods in the interface `ICLTU_SIUpdate` (see table 2-1) and are used in status reports and notifications.

2.2.8 FEATURES NOT HANDLED BY THE PROVIDER SIDE SERVICE INSTANCE

2.2.8.1 Introduction

As a general approach, this specification only states what the API is required to do. Features not identified in this specification cannot be expected from a conforming implementation. This subsection deviates from this approach by discussing features not provided by the API, with the intention to clarify the borderline between the application and the API Service Element.

In addition, this subsection outlines the rationale for the distribution of responsibilities between the application and the API Service Element in this specification.

2.2.8.2 Production Status

Reference [4] defines a parameter 'production status', which reflects the state of the CLTU production engine. The value of the production status is not only included in status reports and notifications, but also determines whether invocations of the operations CLTU–BIND and CLTU–START can be accepted or not. The production status also has an impact on processing of CLTU–TRANSFER–DATA operations, which is discussed in 2.2.8.4.

Table 2-4 lists the possible values of the production status and the required processing of BIND and START invocations.

Table 2-4: CLTU Production Status

Production Status	BIND invocation	START invocation
halted	reject (out of service)	reject (out of service)
configured	accept	accept
operational	accept	accept
interrupted	accept	reject (unable to comply)

In a multi-threaded environment, the value of the production status can change concurrently with processing within the service element. That implies, that the value can change after a

© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

PDU has been processed by the service element but before the same PDU is handled by the application. Because the service element cannot guarantee that the result of a test is still valid when the PDU reaches the application, this specification does not require that the service element check the production status.

This specification does not exclude that implementations of the service element check the production status and reject BIND or START invocation if required. If both the API and the application are single-threaded, the application could rely on such checks. However, applications cannot expect that other implementations provide the same service. Therefore, applications wishing to maintain substitutability of API components should not rely on such behavior.

2.2.8.3 Rejecting Invocations of TRANSFER-DATA and THROW-EVENT Operations

For CLTU–TRANSFER–DATA returns, reference [4] requires that the provider insert the next expected CLTU identification and the available CLTU buffer size. For CLTU-THROW-EVENT returns, reference [4] requires that the provider insert the next expected event invocation identifier. These parameters are available to the service element via the procedures described in 2.2.5. However, the following must be considered.

A service user is not required to wait for a CLTU–TRANSFER–DATA return before transmitting the next CLTU–TRANSFER–DATA invocation. Therefore, several CLTU–TRANSFER–DATA invocations can be in transit. Depending on the implementation of the service element and of the provider application, CLTU–TRANSFER–DATA invocations might be queued between the service element and the application. In such a case, the service element cannot know what values to insert for the next CLTU identification and the available buffer size when it needs to generate a CLTU–TRANSFER–DATA return with a negative result. The same considerations apply to the CLTU–THROW–EVENT operation.

Therefore, this specification defines a procedure for the CLTU–TRANSFER–DATA operation and for the CLTU–THROW–EVENT operation, which deviates from the standard approach described in reference [5]. When a check performed by the service element fails, the service element can set the appropriate diagnostic in the operation object and pass the operation object to the application. The application is expected to check the result of an invocation. If the result is negative, the application should insert the next expected CLTU identification and the available buffer size or the next expected event invocation identifier into the operation object and then pass it back to the service element using the method `InitiateOpReturn()` in the interface `ISLE_ServiceInitiate`.

This specification does not exclude that implementations generate a CLTU–TRANSFER–DATA return or a CLTU–THROW–EVENT return if it is possible to insert the correct values for the return parameters. An implementation can apply any of the following approaches:

a) an implementation can always pass invocations for which a check has failed to the application;

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

b) an implementation can prevent queuing of invocations by withholding an invocation until the previous invocation has been confirmed by the application. In that case, it can always generate the appropriate return when needed; or

c) an implementation can decide to pass invocations to the application on a case by case basis.

Applications wishing to maintain substitutability of API components should always expect to receive CLTU–TRANSFER–DATA invocations and CLTU–THROW–EVENT invocations with a negative result from the service element.

2.2.8.4 Processing of TRANSFER-DATA Invocations

2.2.8.4.1 Blocked State of the Service Instance

When a CLTU cannot be radiated because the production status becomes non-operational or because the latest radiation start time expired, the service instance becomes blocked and further CLTU–TRANSFER–DATA invocations must be rejected. In order to clear the situation, the service user must invoke a CLTU–STOP operation followed by a CLTU–START operation.

The event causing the blocked state of the service instance can depend on the production status, which can change concurrently with processing in the service element. In a multi-threaded environment, the service element cannot guarantee that a CLTU–TRANSFER–DATA invocation that passed the test of the blocked state is still valid when it reaches the application. Therefore, this specification does not require the service element to perform that check.

This specification does not exclude that implementations check the blocked state of the service instance. If both the API and the application are single-threaded, the application could rely on such checks. However, applications cannot expect that other implementations provide the same service. Therefore, applications wishing to maintain substitutability of API components should not rely on such behavior.

2.2.8.4.2 Checking of Time Parameters

CLTU–TRANSFER–DATA invocations carry parameters that specify the earliest and latest radiation times. Reference [4] requires the service provider to check that these times are not expired at the time the invocation reaches the provider. It cannot be excluded that such a time expires after the invocation has been processed by the service element, but before it reaches the application. Therefore, this specification does not require the service element to perform these checks. The service element is, however, required to verify that time periods are defined in a consistent manner.

This specification does not exclude that implementations check times against current time. However, applications wishing to maintain substitutability of API components should not rely on such behavior.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

2.2.8.5 Production Time

Reference [4] defines a production period, i.e., the period in which the CLTU production engine is able to radiate CLTUs. This period must overlap with the scheduled provision period of the service instance but need not be the same. Reference [4] requires the service provider to check the validity of CLTU–START invocations and CLTU–TRANSFER–DATA invocations against the production period.

This specification does not require a service element to perform these checks, as they are related to service production and not to service provisioning.

2.3 PACKAGE CLTU OPERATIONS

Figure 2-2 shows the operation object interfaces required for the CLTU service. The package CLTU Operations adds operation objects for the following CLTU operations:

- CLTU–START;
- CLTU–TRANSFER–DATA;
- CLTU–ASYNC–NOTIFY;
- CLTU–STATUS–REPORT;
- CLTU–GET–PARAMETER;
- CLTU–THROW–EVENT.

For other operations the API uses the common operation objects defined in reference [5].

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016

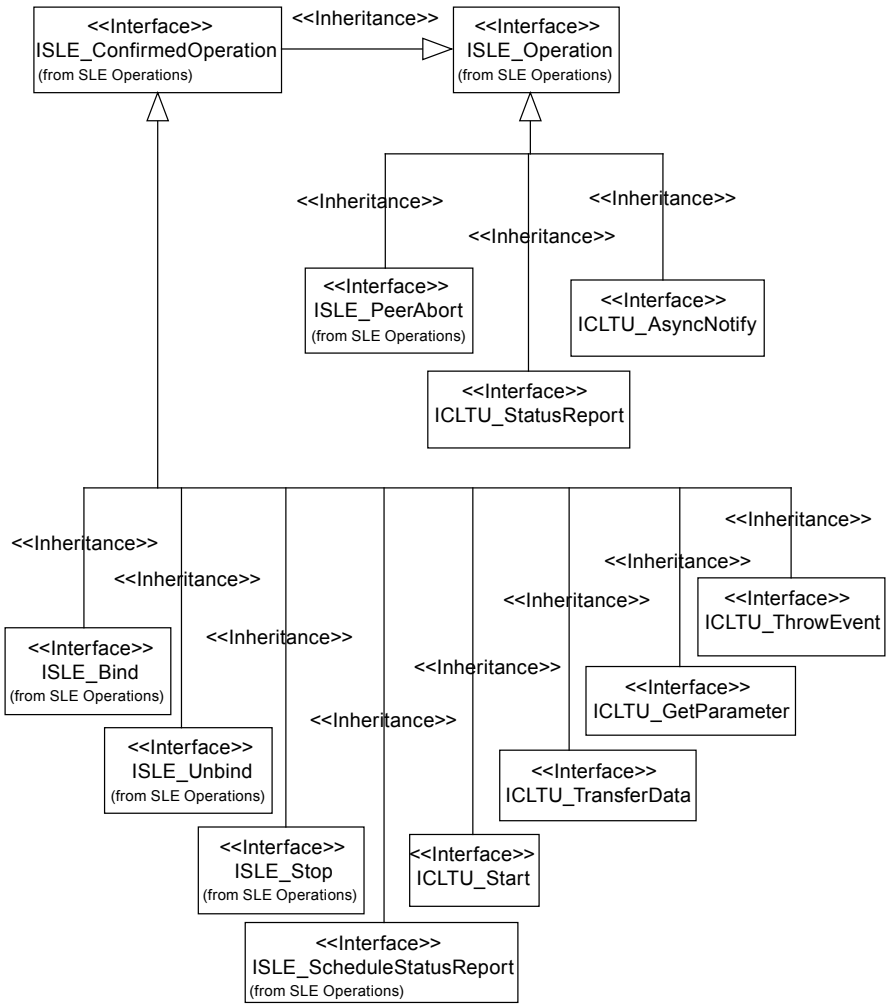

Figure 2-2: CLTU Operation Objects

© ISO 2016 – All rights reserved
Full document details: BS ISO 18445:2016