BS ISO 18446:2016

BSI Standards Publication

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

Documents
Document Sharing Hub

Full document details: BS ISO 18446:2016

National foreword

This British Standard is the UK implementation of ISO 18446:2016. It supersedes BS ISO 18446:2013 which is withdrawn.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 93384 4

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 18446:2016

INTERNATIONAL STANDARD

ISO 18446

Second edition

2016-11-15

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

Systèmes de transfert des informations et données spatiales — Extension de liaisons spatiales — Interface du programme d'application pour le service d'envoi de données spatiales par paquets

Reference number
ISO 18446:2016(E)

Documents
Document Sharing Hub

© ISO 2016

Full document details: BS ISO 18446:2016

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 18446:2016

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 18446 was prepared by the Consultative Committee for Space Data Systems (CCSDS) (as CCSDS 916.3-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 18446:2013), which has been technically revised.

© ISO 2016 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 18446:2016

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 18446:2016

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 [4]). 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 18446:2016

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 18446:2016

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 916.3-M-1	Space Link Extension—Application Program Interface for the Forward Space Packet Service, Recommended Practice, Issue 1	October 2008	Original issue, superseded
CCSDS 916.3-M-2	Space Link Extension—Application Program Interface for the Forward Space Packet 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 18446:2016

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 FSP SERVICE INSTANCES 2-1
2.3 PACKAGE FSP OPERATIONS 2-17
2.4 SECURITY ASPECTS OF THE SLE FORWARD SPACE PACKET (FSP) TRANSFER SERVICE 2-19

3 FSP SPECIFIC SPECIFICATIONS FOR API COMPONENTS 3-1

3.1 API SERVICE ELEMENT 3-1
3.2 SLE OPERATIONS 3-22
3.3 SLE APPLICATION 3-23
3.4 SEQUENCE OF DIAGNOSTIC CODES 3-25

ANNEX A FSP 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 FSP Service Instances 2-2
2-2 FSP Operation Objects 2-18

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

CONTENTS (continued)

Table Page

2-1 Production Events Reported via the Interface IFSP_SIUpdate 2-6
2-2 FSP Configuration Parameters Handled by the Service Element 2-10
2-3 FSP FOP Parameters Handled by the Service Element 2-11
2-4 FSP Status Parameters Handled by the Service Element 2-12
2-5 FSP Production Status 2-14
2-6 Mapping of FSP Operations to Operation Object Interfaces 2-19

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

Documents
Document Sharing Hub

© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

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 that is independent of specific SLE service types.

The purpose of this document is to specify extensions to the API needed for support of the Forward Space Packet Service defined in reference [3].

1.2 SCOPE

This specification defines extensions to the SLE API in terms of:

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

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

c) the externally visible behavior associated with the FSP 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 Space Packet service in addition to the specification in reference [5].

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

1.3 APPLICABILITY

The Application Program Interface specified in this document supports two generations of Forward Space Packet service, namely:

a) Generation 2 identified by the version number 1 in the BIND operation, as specified by reference [C8];

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

NOTES

1 The use of the term 'Generation' follows the definition in the API Core Specification (reference [4]), where it is used to classify all SLE Transfer Services.

2 Support for Generation 1 is not valid in this book because the FSP initial issue was written as part of Generation 2 of the RAF, RCF and CLTU specific version 1 recommendations.

Support for Generation 2 of the service is included for backward compatibility purposes for a limited time and may not be continued in future versions of this specification.

Provisions within this Recommended Practice that are specific for a specific generation are marked as:

– [G2:] for provisions specific to Generation 2;

– [G3:] for provisions specific to Generation 3.

Provisions that apply to all generations are not marked.

1.4 RATIONALE

This Recommended Practice specifies the mapping of the Forward Space Packet 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:

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.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

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 Recommended Practice, identifies conventions, and lists definitions and references used throughout the document;
- section 2 describes the extension of the API model defined in reference [5] to include support for the FSP 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 FSP 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 [2]). 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 18446:2016

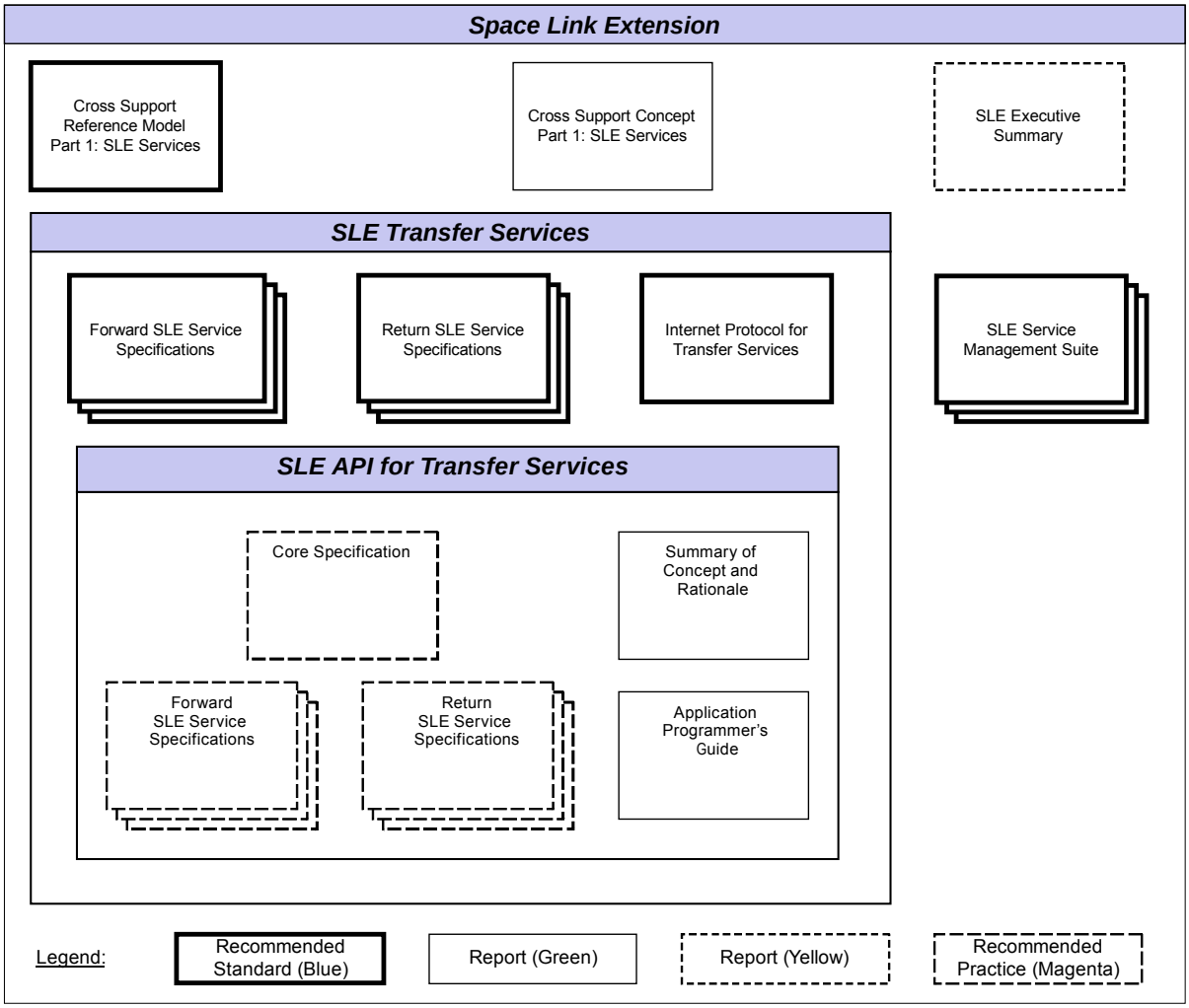

Figure 1-1: SLE Services and SLE API Documentation

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 18446:2016

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 TC Space Data Link Protocol

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

a) AD, BD, BC;

b) Command Link Control Word (CLCW);

c) Frame Operation Procedure (FOP);

d) Multiplexer Access Point (MAP);

e) Virtual Channel (VC).

1.6.1.2 Definitions from SLE Reference Model

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

a) Forward Space Packet service;

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

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).

1.6.1.3 Definitions from FSP Service

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

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.

1.6.1.4 Definitions from ASN.1 Specification

This Recommended Practice makes use of the following terms defined in reference [6]:

a) Object Identifier;

b) Octet String.

1.6.1.5 Definitions from UML Specification

This Recommended Practice makes use of the following terms defined in reference [C7]:

a) Attribute;

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

b) Base Class;

c) Class;

d) Data Type;

e) Interface;

f) Method.

1.6.1.6 Definitions from API Core Specification

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

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.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

1.6.3 CONVENTIONS

This document applies the conventions defined in reference [4].

The model extensions in section 2 are presented using the Unified Modeling Language (UML) and applying the conventions defined in reference [4].

The FSP-specific specifications for API components in section 3 are presented using the conventions specified in reference [4].

The FSP-specific interfaces in annex A are specified using the conventions defined in reference [4].

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 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.

[2] *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.

[3] *Space Link Extension—Forward Space Packet Service Specification*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 912.3-B-2. Washington, D.C.: CCSDS, July 2010.

[4] *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.

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

[6] *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 18446:2016

2 OVERVIEW

2.1 INTRODUCTION

This section describes the extension of the SLE API model in reference [5] for support of the FSP 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 FSP-specific protocol data units.

2.2 PACKAGE FSP SERVICE INSTANCES

2.2.1 OVERVIEW

The FSP extensions to the component API Service Element are defined by the package FSP 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 FSP service.

The package adds two service instance classes:

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

b) FSP 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 FSP operations.

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

The class FSP SI Provider adds three new interfaces:

a) `IFSP_SIAdmin` by which the application can set FSP-specific configuration parameters;

b) `IFSP_FOPMonitor` by which the application can initialize and update parameters related to the FOP machine; and

c) `IFSP_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]. For the FSP service, the conceptual interface `I<SRV>_SIAdmin` is split into the two interfaces `IFSP_SIAdmin` and `IFSP_FOPMonitor` because of the large number of parameters that must be handled.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

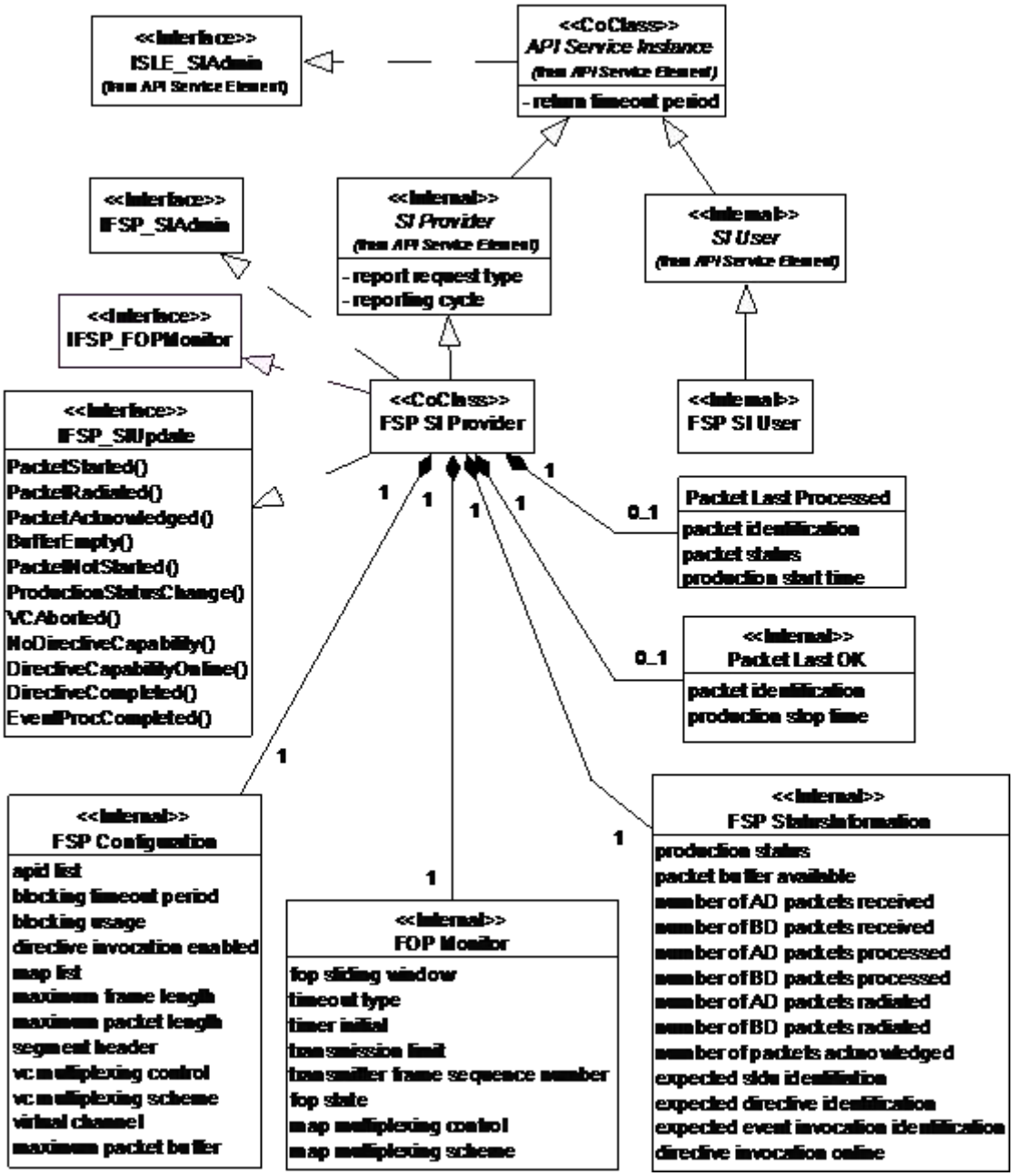

Figure 2-1: FSP Service Instances

FSP-specific service parameters are defined by the internal classes FSP Configuration and FOP Monitor. The class FSP Status Information defines dynamic status parameters maintained by the service instance. In addition, the service instance maintains a set of parameters for the last packet for which processing started and for the last packet for which processing was successfully completed. These parameters are defined by the classes Packet Last Processed and Packet Last OK.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

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 FSP SI USER

The class defines a FSP service instance supporting the service user role. It ensures that SLE PDUs passed by the application and by the association are supported by the FSP service and handles the FSP 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 FSP SI PROVIDER

The class defines a FSP service instance supporting the service provider role. It exports the interfaces `IFSP_SIAdmin` for configuration of the service instance after creation, `IFSP_FOPMonitor` for update of `FOP` parameters, and `IFSP_SIUpdate` for update of dynamic status parameters during operation.

2.2.3.1 Responsibilities

2.2.3.1.1 Service Specific Configuration

The service instance implements the interface `IFSP_SIAdmin` to set the FSP-specific configuration parameters defined by the class FSP 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` and the interface `IFSP_FOPMonitor`) have been set, the method `ISLE_SIAdmin::ConfigCompleted()` must be called. This method verifies that all configuration parameter values are defined and are in the range defined in reference [3].

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

2.2.3.1.2 Initialization and Update of FOP Parameters

The service instance implements the interface `IFSP_FOPMonitor` for initialization and update of the parameters defined by the class FOP Monitor. The API service instance uses these parameters only to respond to GET-PARAMETER invocations. The application must initialize these parameters when configuring the service instance and update them whenever they change during the lifetime of the service instance.

Changes of the parameter values might occur because of directives invoked by a service user on the same or on a different service instance, because of events detected by the FOP machine, or because of management action. In order to ensure that the service instance

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

always reports the correct parameter value, updates must be forwarded independent of the service instance state.

2.2.3.1.3 Update of Dynamic Status Parameters

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

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 FSP Status Information from FSP PDUs exchanged between the service user and the service provider. The methods used to update each of the parameters are defined in 3.1.4.13.

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

2.2.3.1.4 Generation of Notifications

If events reported via the interface `IFSP_SIUpdate` require that a FSP–ASYNC–NOTIFY invocation be sent to the service user, the class generates and transmits the invocation 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`.

2.2.3.1.5 Handling of the FSP–GET-PARAMETER Operation

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

2.2.3.1.6 Status Reporting

The class generates FSP–STATUS–REPORT invocations when required using the parameters maintained by the classes FSP Status Information, Packet Last Processed, and Packet Last OK.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

2.2.3.1.7 Processing of FSP Protocol Data Units

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

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

Table 2-1: Production Events Reported via the Interface IFSP_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
Processing of a packet started.	`PacketStarted`	packet id transmission mode start time available buffer size	packet id last processed production start time packet status number of AD packets processed (See Note 1) number of BD packets processed (See Note 2) packet buffer available	packet processing started
Radiation of a packet completed	`PacketRadiated`	packet id transmission mode radiation end time (See Note 2)	packet id last OK (See Note 2) packet status production stop time(See Note 2) number of AD packets radiated(See Note 1) number of BD packets radiated(See Note 2)	packet radiated
All segments of an AD packet acknowledged via the CLCW	`PacketAcknowledged`	packet id acknowledge time	packet id last OK packet status production stop time number of packets acknowledged	packet acknowledged
The packet buffer is empty	`BufferEmpty`		packet buffer available	buffer empty

Documents
Document Sharing Hub

© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

Event	Method	Arguments	Status parameters updated	Notification sent
Processing of a packet could not be started because the latest production time expired; the production status was interrupted; The required transmission mode capability was not available	`PacketNotStarted`	packet id transmission mode start time failure reason affected packets list available buffer size	packet id last processed packet status production start time packet buffer available	sldu expired production interrupted transmission mode mismatch
The production status changed	`ProductionStatusChange`	production status affected packets list (See Note 4) fop alert (See Note 5) available buffer size	production status packet status (See Note 3) packet buffer available	production operational production interrupted production halted transmission mode capability change transmission mode mismatch
The VC was aborted by a directive	`VCAborted`	affected packets list (See Note 4) available buffer size	packet status (See Note 3) production status packet buffer available	VC aborted
The service instance with directive invocation capability is no longer bound	`NoDirectiveCapability`		directive invocation online	no invoke directive capability on this VC
A service instance with directive invocation capability has bound	`DirectiveCapabilityOnline`		directive invocation online	invoke directive capability established on this VC
Processing of a directive completed	`DirectiveCompleted`	directive id result fop alert (See Note 6)		positive confirm response to directive negative confirm response to directive

Documents
Document Sharing Hub

© ISO 2016 – All rights reserved

Full document details: BS ISO 18446:2016

Event	Method	Arguments	Status parameters updated	Notification sent
Processing of a thrown event completed	`EventProcCompleted`	event invocation id event proc result		action list completed action list not completed event condition evaluated to false

NOTES

1 If the transmission mode is sequence controlled.

2 If the transmission mode is expedited.

3 If the packet id last processed is contained in the affected packets list argument.

4 If no packets were affected, the list is empty.

5 Only needed in case of a transmission mode capability change.

6 Only needed in case of a negative result.

Documents
Document Sharing Hub

© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

2.2.3.1.8 Processing of FSP–TRANSFER–DATA Invocations

For incoming FSP–TRANSFER–DATA invocations the class performs the checks defined in 3.1.5.1 in addition to those defined in reference [4].

In contrast to standard handling of 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 packet 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 2.2.9.3.

2.2.3.1.9 Processing of FSP–THROW-EVENT invocations

In contrast to standard handling of 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 2.2.9.3.

2.2.3.1.10 Processing of FSP–INVOKE-DIRECTIVE invocations

For incoming FSP–INVOKE-DIRECTIVE invocations the class verifies that the service instance has the capability to invoke directives in addition to the checks defined in reference [5].

In contrast to standard handling of 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 directive 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 2.2.9.3.

2.2.4 INTERNAL CLASS FSP CONFIGURATION

The class defines the configuration parameters that can be set via the interface `IFSP_SIAdmin`. These parameters are defined by reference [3]. 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. Table 2-2 only indicates which parameters must not be modified in order to ensure proper operation of the API. Updates allowed by the table might be inhibited because of other constraints.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

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

Parameter	Used for	Upd
`apid-list`	FSP–GET–PARAMETER	Y
[G3:] `bit-lock-required`	FSP–GET–PARAMETER	Y
`blocking-timeout-period`	FSP–GET–PARAMETER	Y
`blocking-usage`	FSP–GET–PARAMETER	Y
`directive-invocation-enabled`	FSP–GET–PARAMETER Checking of FSP-INVOKE-DIRECTIVE	Y
`map-list`	FSP–GET–PARAMETER Checking of FSP-TRANSFER-DATA	Y
`maximum-frame-length`	FSP–GET–PARAMETER	Y
`maximum-packet-length`	FSP–GET–PARAMETER Checking of FSP-TRANSFER-DATA	Y
[G3:] `rf-available-required`	FSP–GET–PARAMETER	Y
`segment-header`	FSP–GET–PARAMETER	Y
`vc-multiplexing-control`	FSP–GET–PARAMETER	Y
`vc-multiplexing-scheme`	FSP–GET–PARAMETER	Y
`virtual-channel`	FSP–GET–PARAMETER	Y
`maximum-packet-buffer-size`	Value of the status parameter packet buffer available after configuration, FSP-STOP, FSP-PEER-ABORT, and protocol abort	N

© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016
Documents
Document Sharing Hub

2.2.5 INTERNAL CLASS FOP MONITOR

The class defines the FOP parameters that can be initialized and updated via the interface `IFSP_FOPMonitor`. These parameters are defined by reference [3]. Table 2-3 describes how the service element uses these parameters. The parameters 'map multiplexing scheme' and 'map multiplexing control' are assigned to this class because 'map multiplexing control' can be modified by a service user via a directive in the same way as FOP control parameters.

Table 2-3: FSP FOP Parameters Handled by the Service Element

Parameter	Used for
`fop-sliding-window`	FSP–GET–PARAMETER
`timeout-type`	FSP–GET–PARAMETER
`timer-initial`	FSP–GET–PARAMETER
`transmission-limit`	FSP–GET–PARAMETER
`transmitter-frame-sequence-number`	FSP–GET–PARAMETER
`fop-state`	FSP–GET–PARAMETER
`map-multiplexing-control`	FSP–GET–PARAMETER
`map-multiplexing-scheme`	FSP–GET–PARAMETER

2.2.6 INTERNAL CLASS FSP STATUS INFORMATION

The class defines status parameters handled by the service instance. The parameters are defined by reference [3]. Table 2-4 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 18446:2016

Table 2-4: FSP Status Parameters Handled by the Service Element

Parameter	Update	Used for
`production-status`	– set by methods of `IFSP_SIUpdate`	status reports notifications
`packet-buffer-available`	– set to maximum at configuration time – set by methods of `IFSP_SIUpdate` – extracted from FSP-TRANSFER-DATA returns – reset to maximum following a notification 'buffer empty' – reset to maximum following FSP–STOP, FSP–PEER–ABORT and protocol abort	status reports notifications
`number-of-AD-packets-received`	– set to zero at configuration time – incremented for every FSP-TRANSFER-DATA return with transmission mode 'sequence controlled' and a positive result	status reports
`number-of-BD-packets-received`	– set to zero at configuration time – incremented for every FSP-TRANSFER-DATA return with transmission mode 'expedited' and a positive result	status reports
`number-of-AD-packets-processed`	– set to zero at configuration time – incremented with every call to `PacketStarted()` and `PacketNotStarted()` with the argument transmission mode set to 'sequence controlled'	status reports
`number-of-BD-packets-processed`	– set to zero at configuration time – incremented with every call to `PacketStarted()` and `PacketNotStarted()` with the argument transmission mode set to 'expedited'	status reports
`number-of-AD-packets-radiated`	– set to zero at configuration time – incremented with every call to `PacketRadiated()` with the argument transmission mode set to 'sequence controlled'	status reports
`number-of-BD-packets-radiated`	– set to zero at configuration time – incremented with every call to `PacketRadiated()` with the argument transmission mode set to 'expedited'	status reports
`number-of-packets-acknowledged`	– set to zero at configuration time – incremented with every call to `PacketAcknowledged()`	status reports

© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016
Documents
Document Sharing Hub

Parameter	Update	Used for
`expected-sldu-identification`	– set to zero at configuration time – copied from the first packet identification parameter of FSP-START invocations if the application transmits a return with a positive result. – copied from packet identification of FSP-TRANSFER-DATA returns	FSP-GET-PARAMETER
`expected-directive-identification`	– set to zero at configuration time – extracted from FSP-INVOKE-DIRECTIVE returns	FSP-GET-PARAMETER
`expected-event-invocation-identification`	– set to zero at configuration time – extracted from FSP-THROW-EVENT returns	FSP-GET-PARAMETER
`directive-invocation-online`	See Note	FSP-GET-PARAMETER

NOTE – The value of the parameter `directive-invocation-online` at the time the service instance is configured must be specified by the application if directive invocation is not enabled for the service instance. Subsequently, the API updates the parameter when the methods `NoDirectiveCapability()` and `DirectiveCapabilityOnline()` are invoked on the interface `IFSP_SIUpdate`. For service instances with invocation capability enabled, the parameter is initialized to 'no' and subsequently set to 'yes' when the user binds and to 'no' when the user unbinds or the association is aborted.

2.2.7 INTERNAL CLASS PACKET LAST PROCESSED

The class defines the parameters maintained by the service instance for the last packet for which processing started or was attempted. These parameters are defined in reference [3].

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

2.2.8 INTERNAL CLASS PACKET LAST OK

The class defines the parameters maintained by the service instance for the last packet for which processing was completed. For BD packets completion implies that the packet was radiated. AD packets are considered complete when all segments of the packet have been acknowledged by a CLCW.

These parameters are defined in reference [3]. All parameters are set via methods in the interface `IFSP_SIUpdate` (see table 2-1) and are used in status reports and notifications.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

2.2.9 FEATURES NOT HANDLED BY THE PROVIDER SIDE SERVICE INSTANCE

2.2.9.1 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.

2.2.9.2 Production Status

Reference [3] defines the parameter production status, which reflects the state of the FSP 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 FSP–BIND and FSP–START can be accepted or not. The production status also has an impact on processing of FSP–TRANSFER–DATA operations, which is discussed in 2.2.9.4.

Table 2-5 lists the possible values of the production status and the required processing of BIND and START invocations.

Table 2-5: FSP 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 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 invocations 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.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

2.2.9.3 TRANSFER-DATA, INVOKE DIRECTIVE, and THROW-EVENT

For FSP–TRANSFER–DATA returns, reference [3] requires that the provider inserts the next expected packet identification and the available packet buffer size. For FSP-INVOKE-DIRECTIVE and FSP-THROW-EVENT returns, reference [3] requires that the provider inserts the next expected directive identification or event invocation identification, respectively. These parameters are available to the service element via the procedures described in 2.2.6. However, the following must be considered.

A service user is not required to wait for a FSP–TRANSFER–DATA return before transmitting the next FSP–TRANSFER–DATA invocation. Therefore, several FSP–TRANSFER–DATA invocations can be in transit. Depending on the implementation of the service element and of the provider application, FSP–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 packet identification and the available buffer size when it needs to generate a FSP–TRANSFER–DATA return with a negative result. The same considerations apply to the FSP-INVOKE-DIRECTIVE and FSP–THROW–EVENT operations.

Therefore, this Recommended Practice defines a procedure for the operations FSP–TRANSFER–DATA, FSP-INVOKE-DIRECTIVE, and FSP–THROW–EVENT, which deviates from the standard approach described in reference [5]. When a check performed by the service element fails, the service element may 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 shall insert the next expected packet identification and the available buffer size, the next expected directive identification, or the next expected event invocation identification 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 FSP–TRANSFER–DATA return, a FSP-INVOKE-DIRECTIVE return, or a FSP–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;

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 FSP–TRANSFER–DATA invocations, FSP-INVOKE-DIRECTIVE invocations, and FSP–THROW–EVENT invocations with a negative result from the service element.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

2.2.9.4 Processing of TRANSFER-DATA Invocations

2.2.9.4.1 Blocked State of the Service Instance

When a packet cannot be processed because the production status becomes non-operational or because the latest production start time expired the service instance becomes blocked and further FSP–TRANSFER–DATA invocations must be rejected. In order to clear the situation, the service user must invoke a FSP–STOP operation followed by a FSP–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 FSP–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. Applications wishing to maintain substitutability of API components must not rely on such behavior.

2.2.9.4.2 Transmission Mode Capability

When an AD packet cannot be processed because the required transmission mode capability is not available the AD mode becomes blocked and further AD packets must be rejected. In order to clear the situation, the user must invoke an FSP-TRANSFER-DATA operation with the transmission mode ‘sequence controlled and unblock AD’.

The transmission mode capability can change and the AD service can become blocked concurrently with processing in the service element. In a multi-threaded environment, the service element cannot guarantee that a FSP–TRANSFER–DATA invocation that passed the test of the AD blocked state is still acceptable 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 whether the AD service is blocked. 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. Applications wishing to maintain substitutability of API components must not rely on such behavior.

2.2.9.4.3 Checking of Time Parameters

FSP–TRANSFER–DATA invocations carry parameters that specify the earliest and latest production start times. Reference [3] 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

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

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

2.2.9.4.4 APID, Packet Version, and Packet Type

Reference [3] requires that the FSP service provider verify that:

a) the APID of a packet matches one of the entries in the list of permitted APIDs;

b) the packet version is supported by CCSDS and the service instance; and

c) the packet type specifies telecommand.

For these checks it is necessary to inspect the packet delivered by the parameter 'data' in the TRANSFER-DATA operation. Because the SLE Application Program Interface generally handles space link data units as opaque data types, this specification does not require a service element to perform these checks.

This specification does not exclude that implementations check the APID, packet version, and packet type. However, applications wishing to maintain substitutability of API components must not rely on such behavior.

2.2.9.5 Production Time

Reference [3] defines a production period, i.e., the period in which the FSP production engine is able to process packets. This period must overlap with the scheduled provision period of the service instance but need not be the same. Reference [3] requires the service provider to check the validity of FSP–START invocations and FSP–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 FSP OPERATIONS

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

a) FSP–START;

b) FSP–TRANSFER–DATA;

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

c) FSP–ASYNC–NOTIFY;

d) FSP–STATUS–REPORT;

e) FSP–GET–PARAMETER;

f) FSP–THROW–EVENT;

g) FSP–INVOKE–DIRECTIVE.

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

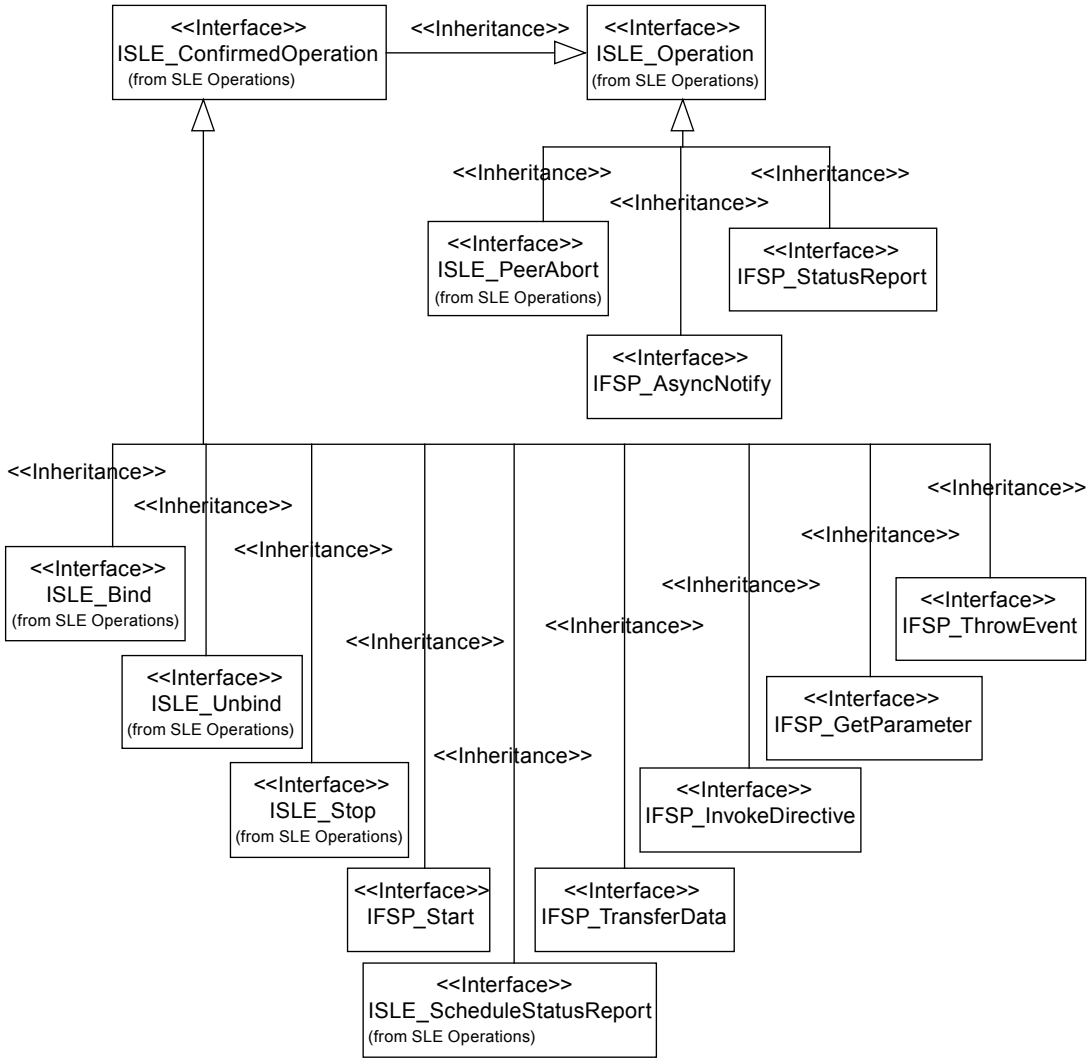

Figure 2-2: FSP Operation Objects

© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016
Documents
Document Sharing Hub

Table 2-6 maps FSP operations to operation object interfaces.

Table 2-6: Mapping of FSP Operations to Operation Object Interfaces

FSP Operation	Operation Object Interface	Defined in Package
FSP–BIND	`ISLE_Bind`	SLE Operations
FSP–UNBIND	`ISLE_Unbind`	SLE Operations
FSP–START	`IFSP_Start`	FSP Operations
FSP–STOP	`ISLE_Stop`	SLE Operations
FSP–TRANSFER–DATA	`IFSP_TransferData`	FSP Operations
FSP–ASYNC–NOTIFY	`IFSP_AsyncNotify`	FSP Operations
FSP–SCHEDULE–STATUS–REPORT	`ISLE_ScheduleStatusReport`	SLE Operations
FSP–STATUS–REPORT	`IFSP_StatusReport`	FSP Operations
FSP–GET–PARAMETER	`IFSP_GetParameter`	FSP Operations
FSP–THROW–EVENT	`IFSP_ThrowEvent`	FSP Operations
FSP–INVOKE–DIRECTIVE	`IFSP_InvokeDirective`	FSP Operations
FSP–PEER–ABORT	`ISLE_PeerAbort`	SLE Operations

2.4 SECURITY ASPECTS OF THE SLE FORWARD SPACE PACKET (FSP) TRANSFER SERVICE

2.4.1 SECURITY BACKGROUND/INTRODUCTION

The SLE transfer services explicitly provide authentication and access control. Additional security capabilities, if required, are levied on the underlying communication services that support the SLE transfer services. The SLE transfer services are defined as layered application services operating over underlying communication services that must meet certain requirements but which are otherwise unspecified. Selection of the underlying communication services over which real SLE implementations connect is based on the requirements of the communicating parties and/or the availability of CCSDS-standard communication technology profiles and proxy specifications. Different underlying communication technology profiles are intended to address not only different performance requirements but also different security requirements. Missions and service providers are expected to select from these technology profiles to acquire the performance and security capabilities appropriate to the mission. Specification of the various underlying communication technologies, and in particular their associated security provisions, are outside the scope of this Recommendation.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

The SLE FSP transfer service transfers data that is destined for a mission spacecraft. As such, the SLE FSP transfer service has custody of the data for only a portion of the end-to-end data path between MDOS and mission spacecraft. Consequently the ability of an SLE transfer service to secure the transfer of mission spacecraft data is limited to that portion of the end-to-end path that is provided by the SLE transfer service (i.e., the terrestrial link between the MDOS and the ground termination of the ground-space link to the mission spacecraft). End-to-end security must also involve securing the data as it crosses the ground-space link, which can be provided by some combination of securing the mission data itself (e.g., encryption of the mission data within CCSDS space packets) and securing the ground-space link (e.g., encryption of the physical ground-space link). Thus while the SLE FSP transfer service plays a role in the end-to-end security of the data path, it does not control and cannot ensure that end-to-end security. This component perspective is reflected in the security provisions of the SLE transfer services.

2.4.2 STATEMENTS OF SECURITY CONCERNS

2.4.2.1 Overview

This subsection identifies SLE FSP transfer service support for capabilities that responds to these security concerns in the areas of data privacy, data integrity, authentication, access control, availability of resources, and auditing.

2.4.2.2 Data Privacy (Also Known As Confidentiality)

This SLE FSP transfer service specification does not define explicit data privacy requirements or capabilities to ensure data privacy. Data privacy is expected to be ensured outside of the SLE transfer service layer, by the mission application processes that communicate over the SLE transfer service, in the underlying communication service that lies under the SLE transfer service, or some combination of both. For example, mission application processes might apply end-to-end encryption to the contents of the CCSDS space link data units carried as data by the SLE transfer service. Alternatively or in addition, the network connection between the SLE entities might be encrypted to provide data privacy in the underlying communication network.

2.4.2.3 Data Integrity

The SLE FSP service requires that each transferred space packet be accompanied by a sequence number, which must increase monotonically. Failure of a space packet to be accompanied by the expected sequence number causes the space packet to be rejected (see 3.6.2.18.1 d) in reference [3]). This constrains the ability of a third party to inject additional command data into an active FSP transfer service instance.

The SLE FSP transfer service defines and enforces a strict sequence of operations that constrain the ability of a third party to inject operation invocations or returns into the transfer

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

service association between a service user and provider (see 4.2.2 in reference [3]). This constrains the ability of a third party to seize control of an active FSP transfer service instance without detection.

The SLE FSP transfer service requires that the underlying communication service transfer data in sequence, completely and with integrity, without duplication, with flow control that notifies the application layer in the event of congestion, and with notification to the application layer in the event that communication between the service user and the service provider is disrupted (see 1.3.1 in reference [3]). No specific mechanisms are identified, as they will be an integral part of the underlying communication service.

2.4.2.4 Authentication

This SLE FSP transfer service specification defines authentication requirements (see 3.1.5 in reference [3]), and defines `initiator-identifier`, `responder-identifier`, `invoker-credentials`, and `performer-credentials` parameters of the service operation invocations and returns that are used to perform SLE transfer service authentication. The procedure by which SLE transfer service operation invocations and returns are authenticated is described in annex F of the Cross Support Service Green Book (reference [C2]). The SLE transfer service authentication capability can be selectively set to authenticate at one of three levels: authenticate every invocation and return, authenticate only the BIND operation invocation and return, or perform no authentication. Depending upon the inherent authentication available from the underlying communication network, the security environment in which the SLE service user and provider are operating, and the security requirements of the spaceflight mission, the SLE transfer service authentication level can be adapted by choosing the SLE operation invocations and returns that shall be authenticated. Furthermore, the mechanism used for generating and checking the credentials and thus the level of protection against masquerading (simple or strong authentication) can be selected in accordance with the results of a threat analysis.

2.4.2.5 Access Control

This SLE FSP transfer service specification defines access control requirements (see 3.1.4 in reference [3]), and defines `initiator-identifier` and `responder-identifier` parameters of the service operation invocations and returns that are used to perform SLE transfer service access control. The procedure by which access to SLE transfer services is controlled is described in annex F of the Cross Support Service Green Book (reference [C2]).

2.4.2.6 Availability of Resources

The SLE transfer services are provided via communication networks that have some limit to the resources available to support those SLE transfer services. If these resources can be diverted from their support of the SLE transfer services (in what is commonly known as “denial of service”) then the performance of the SLE transfer services may be curtailed or

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

inhibited. This SLE FSP transfer service specification does not define explicit capabilities to prevent denial of service. Resource availability is expected to be ensured by appropriate capabilities in the underlying communication service. The specific capabilities will be dependent upon the technologies used in the underlying communication service and the security environment in which the transfer service user and provider operate.

2.4.2.7 Auditing

This SLE FSP transfer service specification does not define explicit security auditing requirements or capabilities. Security auditing is expected to be negotiated and implemented bilaterally between the spaceflight mission and the service provider.

2.4.3 POTENTIAL THREATS AND ATTACK SCENARIOS

The SLE FSP transfer service depends on unspecified mechanisms operating above the SLE transfer service (between a mission spacecraft application process and its peer application process on the ground), underneath the SLE transfer service in the underlying communication service, or some combination of both, 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 command data contained in the FSP protocol data units as they traverse the WAN between service user and service provider.

The SLE FSP transfer service constrains the ability of a third party to seize control of an active SLE transfer service instance, or to inject extra command data into a service instance, but it does not specify mechanisms that would prevent an attacker from intercepting the protocol data units and replacing the contents of the `data` parameter. The prevention of such a replacement attack depends on unspecified mechanisms operating above the SLE transfer service (between a mission spacecraft application process and its peer application process on the ground), underneath the SLE transfer service in the underlying communication service, in bilaterally-agreed extra capabilities applied to the SLE transfer service (e.g., encryption of the `data` parameter) or some combination of the three. 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 “hijack” an established SLE FSP transfer service instance and overwrite the commands in the protocol data units to subvert or destroy the operation of the spacecraft.

If the SLE transfer service authentication capability is not used and if authentication is not ensured by the underlying communication service, attackers may somehow obtain valid `initiator-identifier` values and use them to initiate SLE transfer service instances by which they could subvert or destroy the mission.

The SLE FSP transfer service depends on unspecified mechanisms operating in the underlying communication service to ensure that the supporting network has sufficient resources to provide sufficient support to legitimate users. If no such mechanisms are actually

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

implemented, or the mechanisms selected are inadequate or inappropriate to the network environment in which the mission is operating, an attacker could prevent legitimate users from communicating with their spacecraft, causing degradation or even loss of the mission.

If the provider of SLE FSP transfers service provides no security auditing capabilities, or if a user chooses not to employ auditing capabilities that do exist, then attackers may delay or escape detection long enough to do serious (or increasingly serious) harm to the mission.

2.4.4 CONSEQUENCES OF NOT APPLYING SECURITY

The consequences of not applying security to the SLE FSP transfer service are possible degradation and loss of ability to command the spacecraft, and even loss of the spacecraft itself.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

Documents
Document Sharing Hub

© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

3 FSP SPECIFIC SPECIFICATIONS FOR API COMPONENTS

3.1 API SERVICE ELEMENT

3.1.1 SERVICE INSTANCE CREATION

3.1.1.1 The service element shall allow creation of service instances supporting the Forward Space Packet (FSP) service.

3.1.1.2 FSP service instances shall be provided to support the service provider role and the service user role.

NOTE – As specified a by reference [5], a given implementation of the component API Service Element might support only the user role, only the provider role, or both roles.

3.1.2 SERVICE INSTANCE CONFIGURATION

3.1.2.1 The service element shall provide the interface `IFSP_SIAdmin` for configuration of a provider-side service instance after creation.

3.1.2.2 The interface `IFSP_SIAdmin` shall provide methods to set the following configuration parameters, which the service element uses to respond to GET-PARAMETER invocations received from the service user:

a) `apid-list`;

b) [G3:] `bit-lock-required`;

c) `blocking-timeout-period`;

d) `blocking-usage`;

e) `directive-invocation-enabled`;

f) `map-list`;

g) `maximum-frame-length`;

h) `maximum-packet-length`;

i) [G3:] `rf-available-required`;

j) `segment-header`;

k) `vc-multiplexing-control`;

l) `vc-multiplexing-scheme`; and

m) `virtual-channel`.

Documents Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

NOTE – These parameters are defined in reference [3] for the operation FSP-GET-PARAMETER.

3.1.2.3 The interface `IFSP_SIAdmin` shall provide a method to set the initial value of the parameter `directive-invocation-online` if the configuration parameter `directive-invocation-enabled` is set to 'no'.

NOTE – The parameter `directive-invocation-online` is defined in reference [3] for the operation FSP-GET-PARAMETER. Further processing of this parameter is described in 3.1.4.13.

3.1.2.4 The interface `IFSP_SIAdmin` shall provide methods to set the following parameters, which the service instance uses to initialize parameters of the status report:

a) the maximum size of the packet buffer shall be used to initialize the parameter `packet-buffer-available`;

b) the value of the production status at the time the service instance is configured.

NOTE – Further configuration parameters must be set using the interface `ISLE_SIAdmin` specified in reference [3] and the interface `IFSP_FOPMonitor` defined in 3.1.4.

3.1.2.5 All configuration parameters must be set before the method `ConfigCompleted()` of the interface `ISLE_SIAdmin` is called. If one of the parameters is omitted or the value of a parameter is not within the range specified by reference [3], the method `ConfigCompleted()` shall return an error.

NOTES

1 As defined in reference [5], the service element starts processing of the service instance only after successful configuration.

2 The range of specific parameter values might be further constrained by service management. The service element shall only check on the limits specified by reference [3].

3.1.2.6 Configuration parameters listed in 3.1.2.2 can be modified at any time during operation of the service instance. The service element shall always use the most recent value.

NOTE – Modification of these parameters during the scheduled provision period of the service instance might be inhibited by service management. Such constraints must be handled by the application.

3.1.2.7 Configuration parameters defined in 3.1.2.3 and 3.1.2.4 must not be modified after successful return of the method `ConfigCompleted()` defined in the interface

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

`ISLE_SIAdmin`. The effect of an attempt to set these parameters after completion of the configuration is undefined.

3.1.2.8 The values of the configuration parameters identified in 3.1.2.2 shall remain unmodified following an FSP-UNBIND or FSP-PEER-ABORT operation and following a protocol-abort.

3.1.2.9 The interface `IFSP_SIAdmin` shall provide methods to retrieve the values of the configuration parameters. The values returned by these methods before configuration has been completed are undefined.

3.1.3 INITIALIZATION AND UPDATE OF FOP PARAMETERS

3.1.3.1 The service element shall provide the interface `IFSP_FOPMonitor` for initialization and update of parameters related to the FOP machine in a provider side service instance.

3.1.3.2 The interface `IFSP_FOPMonitor` shall provide methods to set the following parameters, which the service element uses to respond to GET-PARAMETER invocations received from the service user:

a) `fop-sliding-window`;

b) `timeout-type`;

c) `timer-initial`;

d) `transmission-limit`;

e) `transmitter-frame-sequence-number`;

f) `fop-state`;

g) `map-multiplexing-control`; and

h) `map-multiplexing-scheme`.

NOTES

1 These parameters are defined in reference [3] for the operation FSP-GET-PARAMETER.

2 The parameters `map-multiplexing-scheme` and `map-multiplexing-control` are included in this interface because `map-multiplexing-control` can be modified by the service user via a directive in the same way as FOP control parameters.

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

3.1.3.3 Initial values for the parameters listed in 3.1.3.2 must be set before the method `ConfigCompleted()` of the interface `ISLE_SIAdmin` is called. If one of the parameters is omitted, or the value of a parameter is not within the range specified by reference [3], the method `ConfigCompleted()` shall return an error.

3.1.3.4 During the complete lifetime of the service instance the parameters listed in 3.1.3.2 must be updated via the interface `IFSP_FOPMonitor`, whenever their value changes.

NOTES

1 Changes might occur because of directives invoked by a service user on the same or on a different service instance, because of events detected by the FOP machine, or because of management action.

2 In order to ensure that the service instance always reports the correct parameter value, updates must be reported independent of the service instance state.

3.1.3.5 The interface `IFSP_FOPMonitor` shall provide methods to retrieve the values of the parameters. The values returned by these methods before configuration has been completed are undefined.

3.1.4 STATUS INFORMATION

3.1.4.1 Status Parameters

3.1.4.1.1 The service element shall maintain status parameters for every service instance and use them for generation of status reports, notifications, and for FSP–GET–PARAMETER returns.

NOTES

1 The parameters are defined in reference [3] for the operations FSP-ASYNC-NOTIFY, FSP-STATUS-REPORT, and FSP-GET-PARAMETER.

2 Handling of the parameter `reporting-cycle` defined for the FSP-GET-PARAMETER operation shall be specified in reference [5].

3.1.4.1.2 The service element shall update the following status parameters in the methods of the interface `IFSP_SIUpdate` described in 3.1.4.2:

a) `packet-identification-last-processed`;

b) `packet-status`;

c) `production-start-time`;

d) `packet-identification-last-OK`;

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

e) `production-stop-time`;

f) `production-status`;

g) `number-of-AD-packets-processed`;

h) `number-of-BD-packets-processed`;

i) `number-of-AD-packets-radiated`;

j) `number-of-BD-packets-radiated`; and

k) `number-of-AD-packets-acknowledged`.

NOTE – The initial values of these parameters following configuration of the service instance are defined in A4.3.

3.1.4.1.3 The service element shall handle the parameter `directive-invocation-online` as defined by the following specifications:

NOTE – The parameter `directive-invocation-online` can be requested by an FSP-GET-PARAMETER invocation.

a) when the parameter `directive-invocation-enabled` is 'no', the initial value shall be set by configuration of the service instance. Subsequently the value shall be set to 'yes' when the method `DirectiveCapabilityOnline()` is invoked and it shall be set to 'no' when the method `NoDirectiveCapability()` is called on the interface `ISLE_SIUpdate`;

b) when the parameter `directive-invocation-enabled` is 'yes', the initial value shall be set to 'no'. Subsequently the value shall be set to 'yes' when the user binds and it shall be reset to 'no' when the user unbinds or when the association is aborted.

3.1.4.1.4 The service element shall handle the parameter `expected-sldu-identification` as defined by the following specifications:

NOTE – The parameter `expected-sldu-identification` can be requested by an FSP-GET-PARAMETER invocation.

a) the parameter shall be set to zero when the service instance is configured;

b) when the application transmits an FSP-START return with a positive result, the value shall be set to the value of the parameter `first-packet-identification` in the FSP-START invocation;

c) the value shall be copied from the parameter `packet-identification` in every FSP-TRANSFER-DATA return issued by the application.

3.1.4.1.5 The service element shall handle the parameter `expected-directive-identification` as defined by the following specifications:

Documents Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016

NOTE – The parameter `expected-directive-invocation-identifier` can be requested by an FSP-GET-PARAMETER invocation.

a) the parameter shall be set to zero when the service instance is configured;

b) the value shall be copied from every FSP-INVOKE-DIRECTIVE return issued by the application.

3.1.4.1.6 The service element shall handle the parameter `expected-event-invocation-identifier` as defined by the following specifications:

NOTE – The parameter `expected-event-invocation-identifier` can be requested by an FSP-GET-PARAMETER invocation.

a) the parameter shall be set to zero when the service instance is configured;

b) the value shall be copied from every FSP-THROW–EVENT return issued by the application.

3.1.4.1.7 The service element shall handle the parameter `packet-buffer-available` as defined by the following specifications:

a) at configuration time, the value shall be copied from the configuration parameter `maximum-packet-buffer`, defined in 3.1.2.4;

b) when the application transmits an FSP-TRANSFER-DATA return the value shall be copied from the parameter set by the application;

c) the value shall be updated whenever passed as argument by one of the methods in the interface `IFSP_SIUpdate`;

d) when the application transmits an FSP-STOP return with a positive result, the value shall be copied from the configuration parameter `maximum-packet-buffer`;

e) when the application or the service element transmits an FSP-ASYNC-NOTIFY invocation with the parameter `notification-type` set to 'buffer empty', the value shall be copied from the configuration parameter `maximum-packet-buffer`;

NOTE – The service element shall transmit the notification when requested by the application via the interface `IFSP_SIUpdate` specified in 3.1.4.2.

f) following an FSP-PEER-ABORT operation and following protocol-abort, the value shall be copied from the configuration parameter `maximum-packet-buffer`.

3.1.4.1.8 The service element shall handle the parameter `number-of-AD-packets-received` as defined by the following specifications:

a) the parameter shall be set to zero when the service instance is configured;

Documents
Document Sharing Hub
© ISO 2016 – All rights reserved
Full document details: BS ISO 18446:2016