The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

TERRESTRIAL GENERIC FILE TRANSFER

DRAFT RECOMMENDED STANDARD

CCSDS 927.1-R-1

RED BOOK

November 2019

The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

TERRESTRIAL GENERIC FILE TRANSFER

DRAFT RECOMMENDED STANDARD

CCSDS 927.1-R-1

RED BOOK
November 2019

AUTHORITY

Issue:	Red Book, Issue 1
Date:	November 2019
Location:	Not Applicable

(WHEN THIS RECOMMENDED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the email address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
Email: secretariat@mailman.ccsds.org

STATEMENT OF INTENT

(WHEN THIS RECOMMENDED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF INTENT:)

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

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

- Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.
- Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:
  - The **standard** itself.
  - The anticipated date of initial operational capability.
  - The anticipated duration of operational service.
- Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

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

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

FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in the *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

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

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

Member Agencies

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

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian 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.
- China Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Hellenic Space Agency (HSA)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

PREFACE

This document is a draft CCSDS Recommended Standard. Its 'Red Book' status indicates that the CCSDS believes the document to be technically mature and has released it for formal review by appropriate technical organizations. As such, its technical contents are not stable, and several iterations of it may occur in response to comments received during the review process.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document's technical content.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 927.1-R-1	Terrestrial Generic File Transfer, Draft Recommended Standard, Issue 1	November 2019	Current draft

CONTENTS

CONTENTS (continued)

Section Page

Figure

Table

1 INTRODUCTION

1.1 PURPOSE AND SCOPE

1.1.1 GENERAL

This Terrestrial Generic File Transfer (TGFT) Recommended Standard specifies a standard mechanism for transferring files and associated metadata between space agencies.

1.1.2 PURPOSE

The purpose of the Recommended Standard is to provide a standard mechanism by which files can be exchanged between space agencies. Files exchanged between agencies may be used in, but are not limited to

a) mission design, that is, in investigating the feasibility of a mission with respect to the support available from another agency;

b) mission operations, that is, to transfer files required for the successful operation of a mission between two or more agencies or ground system elements; and

c) post-mission activities such as archiving data related to the mission.

1.1.3 SCOPE

The scope of this Recommended Standard is limited to terrestrial file transfer and in particular deals with the point-to-point delivery of files in the terrestrial context. It should be noted that, whilst the scope of the document is limited to the terrestrial case, it is envisaged that mechanisms can be supported whereby a file may be delivered via TGFT along with metadata that enables the file to be injected into the CCSDS File Delivery Protocol (CFDP—see references [H9] and [H10]) entity for forwarding to a spacecraft via the space link, or similarly files received by a CFDP entity could be injected into the TGFT for delivery to another agency. It should be noted that the mechanism by which the injection to/from a CFDP entity is managed is outside the scope of this document.

Figure 1-1 shows the conceptual model for TGFT. This is a conceptual diagram only and is not intended to reflect the real-world configurations of any agency's network setup. It is assumed that, because of security considerations, the end points (i.e., originating and destination nodes) of the file transfer will be positioned on some sort of firewall demilitarized zone.

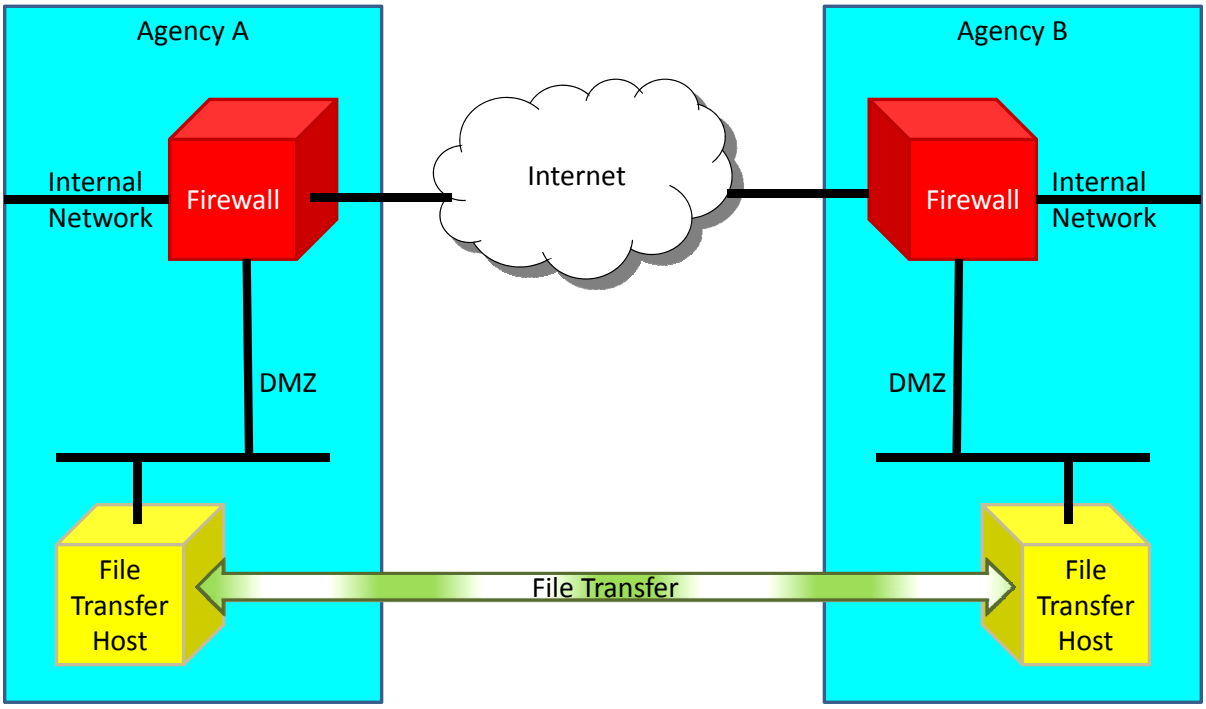

Figure 1-1: Terrestrial File Transfer Conceptual Model

The file transfer hosts, although only one is illustrated at each agency, may be redundant.

The following points should also be noted:

a) Whilst it is probable that each host will be protected by a firewall and be situated on a DeMilitarized Zone (DMZ), this is by no means mandatory. From the point of view of the file transfer, the only necessity is that it be possible to establish between the two hosts a connection that permits the required file transfer activities to take place. How an agency configures the security of its system is not the concern of this Recommended Standard.

b) The above diagram essentially illustrates the symmetric case for file transfer; that is, files can be transferred in either direction. However, each TGFT association is unidirectional in nature, and for a given TGFT association between two hosts, one host is the sender of files, and the other is the receiver of files. A given file transfer host can act as the file sender in some TGFT associations and as the file receiver in other associations. Thus a pair of file transfer hosts can carry bidirectional file transfers between them, but only in the form of multiple TGFT associations.

1.2 APPLICABILITY

It is intended that this Recommended Standard be generally applicable for terrestrial file transfers between space agencies.

1.3 RATIONALE

1.3.1 GENERAL

The primary goal of CCSDS is to increase the level of interoperability among agencies. This Recommended Standard furthers that goal by establishing a standardized mechanism by which files and associated metadata can be packaged and exchanged point to point on the ground between space agencies.

1.3.2 USE CASES

The use cases of the TGFT are ubiquitous; they are carried out on a day-by-day basis using a variety of ad hoc mechanisms between agencies. The prime use case can therefore be seen as the standardization of the file transfer mechanism.

1.4 DOCUMENT STRUCTURE

This document is organized as follows:

Section 1 provides the purpose, scope, applicability, and rationale of this Recommended Standard and identifies the conventions and references used throughout the document. This section also describes how this document is organized. A brief description is provided for each section and annex so that the reader will have an idea of where information can be found in the document. It also identifies terminology that is used in this document but is defined elsewhere.

Section 2 provides a brief overview of the CCSDS-recommended *Terrestrial Generic File Transfer* standard.

Section 3 provides details of the protocols to be used, expected user accounts, directory structures, naming conventions, etc.

Section 4 defines the composition rules for eXtensible Markup Language (XML) Formatted Data Unit (XFDU) Packages in the context of TGFT.

Annex A provides the normative Implementation Conformance Statement proforma.

Annex B defines the normative TGFT Service Agreement.

Annex C discusses security, SANA, and patent considerations.

Annex D contains an informative XFDU manifest example.

Annex E provides an informative description of the possible relationships between TGFT and other file-related services.

Annex F provides an informative description of the concept for how XFDUs are used in the context of the TGFT.

Annex G contains a list of abbreviation and acronyms applicable to the Terrestrial Generic File Transfer.

Annex H lists informative references.

1.5 DEFINITIONS

1.5.1 DEFINITIONS FROM XML FORMATTED DATA UNIT STRUCTURE AND CONSTRUCTION RULES (REFERENCE [3])

XFDU Package: A Package Interchange File that contains an XFDU Manifest and is conformant to the semantics specified in reference [3]. An XFDU Package is a specialization of Package Interchange File.

XFDU Manifest: A Manifest that is conformant to the XML schema specified in reference [3].

1.5.2 TERMS DEFINED IN THIS RECOMMENDED STANDARD

For the purposes of this document, the following definitions apply:

agency: A satellite operator or satellite service provider.

compressed folder file type extension part: The part of the transferred XFDU Package name that specifies the file type extension (‘zip’ or ‘tar’).

NOTE – This indicates the method used to compress the XFDU Package folder/subdirectory to a single file.

file recipient: The application process that consumes the file(s) (enclosed in an XFDU Package) from a TGFT Receiver.

file source: The application process that generates the files to be transmitted (enclosed in an XFDU Package) and uses the TGFT Sender to send them to the target TGFT Receiver.

NOTE – TGFT is a ‘push-only’ mechanism: that is, file transfer is always initiated by the holder of the file. As such, terms such as ‘user’ and ‘provider’ can be ambiguous, especially with respect to the relationship to a cross-support service. In order to avoid these ambiguities with respect to the term ‘service’, this Recommended Standard uses the terms *file source* and *file recipient* to indicate the roles with respect to TGFT, in which the file source is the entity that pushes the file to the file recipient.

in-tray: The directory used as the target directory for files pushed to the TGFT Receiver.

metadata file: A file that contains data that is used to interpret or process data contained in a payload data file.

package folder name part: The part of the transfer XFDU Package name that is the name of the computer file system folder/subdirectory that contains the files of the XFDU Package.

NOTE – The folder/subdirectory is compressed to form the XFDU Package file.

payload data file: A file that contains the essential data of the data transfer.

NOTE – One or more payload data files are contained in an XFDU Package, which is the 'service data unit' of TGFT. A payload data file may be accompanied in the same XFDU Package by one or more metadata files that provide information necessary to the interpretation or processing of the payload data.

TGFT Sender: The TGFT entity that sends the XFDU Package containing the file(s).

TGFT Receiver: The TGFT entity that receives the XFDU Package containing the file(s).

TGFT-using application: An application that uses TGFT to exchange a file (or files) with a peer application. A TGFT-using application has a file sender entity and a file recipient entity.

timestamp part: The part of the transferred XFDU Package name that contains the timestamp that indicates the time at which the file begins transmission from the TGFT Sender to the TGFT Receiver.

1.6 NOMENCLATURE

1.6.1 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 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.

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.

[1] ".ZIP File Format Specification." https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT.

[2] *Standard for Information Technology—Portable Operating System Interface (POSIX(R))*. Issue 7. IEEE Std 1003.1-2017. San Francisco and Piscataway, NJ: The Open Group and IEEE, 2018.

[3] *XML Formatted Data Unit (XFDU) Structure and Construction Rules*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 661.0-B-1. Washington, D.C.: CCSDS, September 2008.

[4] E. Rescorla. *HTTP Over TLS*. RFC 2818. Reston, Virginia: ISOC, May 2000.

[5] *Time Code Formats*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 301.0-B-4. Washington, D.C.: CCSDS, November 2010.

[6] *Reference Model for an Open Archival Information System (OAIS)*. Issue 2. Recommendation for Space Data System Practices (Magenta Book), CCSDS 650.0-M-2. Washington, D.C.: CCSDS, June 2012.

[7] *XML Specification for Navigation Data Messages*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 505.0-B-1. Washington, D.C.: CCSDS, December 2010.

2 OVERVIEW

2.1 GENERAL

This section provides a high-level overview of the CCSDS-recommended *Terrestrial Generic File Transfer* standard, which is designed to facilitate standardized exchanges of files and associated metadata between space agencies.

2.2 PROTOCOL

The underlying protocol chosen to support TGFT is HTTPS - HTTP over Transport Layer Security (TLS) (reference [4]).

2.3 PAYLOAD DATA, METADATA AND FILE PACKAGING

Metadata is often required to aid in interpreting or processing payload data, and TGFT supports concurrent transfer of metadata and payload data. CCSDS has a specification that supports packaging metadata with a file (or files) of payload data, the XML Formatted Data Unit (XFDU) Recommended Standard (reference [3]). The XFDU Recommended Standard is based on the use of XML. Per the XFDU Recommended Standard , an XFDU consists of (1) the *XFDU Manifest*, (2) payload data and metadata contained within the Manifest itself, and (3) all payload data files, metadata files, and other XFDUs referenced by the Manifest. The Manifest and some or all of the referenced files may be contained in an *XFDU Package*, which may be compressed into a single file through the use of compression such as ZIP (reference [1]) or TAR (reference [2]).

NOTE – The distinction between payload data and metadata depends on the points of reference from which the data are being viewed. A ‘payload’ file from the perspective of the TGFT may have an internal structure that itself decomposes into lower-levels of payload and metadata; for example, a file to be transmitted to a spacecraft is the payload file as far as the ground station and TGFT are concerned, but that payload file may itself be a ‘package’ containing both a file of commands to be relayed to a planetary rover and a metadata file containing instructions for transmission of that payload file. From the perspective of the receiving spacecraft, the payload data is that enclosed command file.

TGFT uses a modified form of the CCSDS XFDU Package as its ‘service data unit’. The CCSDS XFDU Recommended Standard permits references in the Manifest to files outside the XFDU Package. In such cases, the XFDU is a logical entity and does not exist as a single physical entity. Since the purpose of TGFT is to transfer the payload files, for the purposes of the TGFT, the XFDU Package physically contains the payload data files that are being transferred; that is, the XFDU Manifest references only payload data files within the XFDU Package. The XFDU Package may also contain one or more metadata files that are used in the interpretation or processing of one or more of the payload data files within that XFDU Package, and the XFDU Manifest references those metadata files and specifies their

relationships to the payload data files within the XFDU Package. Also, references to metadata files outside the XFDU Package are permitted for use with TGFT so that commonly used metadata (e.g., XML schema files) can be referenced without having to actually bundle it along with the payload data in the XFDU Package. The XFDU Manifest references any such external metadata file and specifies its relationships to the payload data files within the XFDU Package.

The XFDU Package is constrained to contain only a single XFDU Manifest file, one or more payload files, and zero or more metadata files.

The use of either TAR or ZIP for creation of the XFDU Package is permitted by the TGFT.

3 TGFT FILE TRANSFER SPECIFICATION

3.1 OVERVIEW

The following subsections specify how files shall be exchanged between space agencies.

3.2 TGFT DEFINITION

3.2.1 XFDU Packages shall be transferred as files from the TGFT Sender to the TGFT Receiver using a Hypertext Transfer Protocol (HTTP) Secure (HTTPS) connection.

NOTE – The basic concept adopted in defining the TGFT is that the files always be 'pushed' from the TGFT Sender to the TGFT Receiver.

3.2.2 The application using the TGFT Sender shall have access to a directory on the target TGFT Receiver system.

3.2.3 The target directory and system shall be specified in a service agreement between the agencies.

NOTES

1 This service agreement is defined in annex B below.

2 Bidirectional file transfer is supported by the TGFT recommended standard. In this case, both agencies will require a target TGFT Receiver system with a Hypertext Transfer Protocol server installed and the TGFT Sender and Receiver roles will be as shown in table 3-1.

Table 3-1: Bidirectional File Transfer

File Transfer Direction	TGFT Sender	TGFT Receiver
Agency A –> B	A	B – requires HTTPS server
Agency B –> A	B	A – requires HTTPS server

3.2.4 TRANSFER PROTOCOL

The transfer protocol to be used shall be HTTPS.

NOTE – The use of the HTTPS protocol will require that a suitable HTTPS server be installed on the TGFT Receiver system. Instigation of the file transfer from the TGFT Sender can be done by using any suitable tool; for example, curl is a command line tool for Linux systems.

3.2.5 FILE COMPRESSION

In order to transfer the XFDU Package as a single file, the XFDU Package folder shall be compressed using one of the following two compression methods:

a) ZIP (see reference [1]); or

b) TAR (see reference [2]).

3.2.6 REQUIRED OPERATIONS

The HTTP PUT method shall be used to 'push' a file from the file source to the file recipient.

NOTE – For the TGFT, only one operation is considered essential, and that is provided by the HTTP PUT method. Essentially the PUT method enables a file or files to be pushed from the file source to the file recipient.

3.2.7 HOSTS

3.2.7.1 The hosts for both the TGFT Sender and TGFT Receiver should be transparently redundant; that is, if the prime machine fails, the backup machine should transparently replace it.

NOTE – This could be achieved by the use of dynamic DNS. It is, however, accepted that this may not be possible in all cases, and thus it may be desirable to describe the appropriate failover behavior in the service agreement; for example,

- always transfer files to the prime host unless it is unavailable, in which case the transfer should be to the backup;
- always transfer files to both prime and redundant hosts; or (less desirable)
- no backup is available.

3.2.7.2 Failover behavior should be specified in the service agreement.

3.2.7.3 The target TGFT Receiver host names shall be specified in the service agreement (see annex B below).

3.2.8 ACCOUNTS

3.2.8.1 For each service agreement, the agency acting as file recipient shall provide one login to an agency acting as file source, with appropriate permissions to write files to the target directory for the HTTPS server running on the target TGFT Receiver system(s).

3.2.8.2 Details of the account(s) shall be specified in the service agreement (see annex B below).

3.2.8.3 As a minimum, login to the account shall be by user name/password.

3.2.8.4 Login authentication should be carried out by means of public-private key pairs.

NOTE – Distribution of login information and/or required keys to the involved parties is outside the scope of this document.

3.3 FILES

3.3.1 FILE NAMING

3.3.1.1 XFDU Package File Names

3.3.1.1.1 Overview

Each XFDU Package is organized as a computer file system folder or subdirectory. The name of that folder or subdirectory (hereinafter referred to simply as *folder*) is the package folder name.

In order to transfer the XFDU Package as a single file, the XFDU Package folder is compressed using one of two compression methods: ZIP or TAR (see 3.3.2.2). The XFDU Package file name is therefore the name of the ZIP or TAR file that contains an XFDU Package folder.

Each XFDU Package file has both an original package file name and the transfer package file name.

3.3.1.1.2 Original Package File Name

3.3.1.1.2.1 The original package file name shall be the name of the file that results from the compression of the XFDU Package folder. It shall comprise two parts:

a) the *package folder name part*, as specified in 3.3.1.1.4; and

b) the *compressed folder file type extension part*, as specified in 3.3.1.1.5.

3.3.1.1.2.2 The package folder name part shall be followed by a period, followed by the compressed folder file type extension part, as shown in figure 3-1.

<package folder name part>.<compressed folder file type extension part>

Figure 3-1: Package Folder Name Syntax

3.3.1.1.3 Transferred Package File Name

3.3.1.1.3.1 To minimize the risk of overwriting files of the same name that are still present in the target directory of the TGFT Receiver, the sending side shall insert a timestamp into the file name of the compressed XFDU Package file that is transferred to the TGFT Receiver.

3.3.1.1.3.2 The sending side may also optionally sign the XFDU Package file by calculating a hash value over the compressed XFDU Package file, as specified in 3.4.3.6. The resulting hash value is inserted into the file name of the compressed XFDU Package file that is transferred to the TGFT Receiver.

3.3.1.1.3.3 The name of the XFDU Package file that is transferred by TGFT shall comprise three mandatory parts and one optional part:

a) the *package folder name part* (mandatory), as specified in 3.3.1.1.4;

b) the *timestamp part* (mandatory), as specified in 3.3.1.1.6;

c) the *hash value part* (optional), the calculated hash value represented as a hexadecimal number, as specified in 3.4.3.6; and

d) the *compressed folder file type extension part* (mandatory), as specified in 3.3.1.1.5.

3.3.1.1.3.4 The package folder name part shall be

a) followed by a hyphen (dash);

b) followed by the timestamp part;

c) followed by a period;

d) optionally followed by the hash value part followed by a period; and

e) followed by the compressed folder file type extension part.

NOTE – The syntax of the signed and unsigned XFDU Package file name is shown in figures 3-2 and 3-3, respectively.

<package folder name part>-<timestamp part>.<hash value part>.<compressed folder file type extension part>

Figure 3-2: Syntax of Signed XFDU Package File Name

<package folder name part>-<timestamp part>.<compressed folder file type extension part>

Figure 3-3: Syntax of Unsigned XFDU Package File Name

3.3.1.1.4 Package Folder Name Part

3.3.1.1.4.1 The name of the XFDU Package folder or subdirectory (hereinafter referred to simply as *folder*) shall be the package folder name part.

3.3.1.1.4.2 The rules for forming the package folder name part of the XFDU Package file name shall be defined by the application using TGFT, with the constraint that the package folder name part shall be limited to the following character set:

– [a-z] : lowercase alphabetic characters;

– [0-9] : numeric characters;

– '-' : the 'dash' (or minus) character;

– '_' : the underscore character;

– '.' : the dot character.

NOTE – The above restricted character set is intended to reduce the scope of problems when files are transferred between different operating systems. In particular, the restriction to lowercase alphabetic characters is intended to remove any ambiguity between Windows and Unix; for example, on Windows the file name *file_name.file_type* is equivalent to *FILE_NAME.FILE_TYPE*, while on Unix these are treated as distinct files.

3.3.1.1.5 Compressed Folder File Type Extension Part

The compressed folder file type extension part shall be either 'zip' or 'tar', according to the method used to compress the XFDU Package folder.

3.3.1.1.6 Timestamp Part

3.3.1.1.6.1 The timestamp part shall contain the Zulu time at which the transfer of the file from the TGFT Sender to the TGFT Receiver was started.

3.3.1.1.6.2 The format of the timestamp part shall be a modified version of the CCSDS ASCII Time Code B that is specified in reference [5]:

YYYY-DDDThh-mm-ss-d→dZ,

where (as per reference [5])

YYYY Year in four-character subfield with values 0001-9999;

DDD Day of year in three-character subfield with values 001-365 or -366;

T Calendar-Time separator;

hh Hour in two-character subfield with values 00-23;

mm Minute in two-character subfield with values 00-59;

ss Second in two-character subfield with values 00-59 (-58 or -60 during leap seconds);

d→d Decimal fraction of second in one- to n-character subfield, where each d has values 0-9. As many 'd' characters to the right of the period as required may be used to ensure no risk of overwriting files;

Z time code terminator;

but contrary to using

a) colons (':') to delimit the *hh*, *mm*, and *ss* fields (as per reference [5]), the modified time code shall use dashes ('-') because colons are illegal file name characters in many file systems;

b) a dot ('.') to separate the *ss* and *d→d* fields (as per reference [5]), the modified time code shall use a dash ('-') because a dot could cause problems on some file systems, as this is typically used to delimitate the file name from the file type.

NOTE – For example, for an XFDU Package with an original package file name (see 3.3.1.1.2),

<package folder name part>.<compressed folder file type extension part>,

the name of the XFDU Package that is transferred is

<package folder name part>-YYYY-DDDThh-mm-ss-d→dZ.<compressed folder file type extension part>.

3.3.1.2 XFDU Manifest File Names

The structure of the XFDU Manifest file name is unspecified beyond the following constraints:

a) The file name shall conform to the character set limitation specified in 3.3.1.1.4.2; and

b) The file type (extension) '.xfdu' shall be used to uniquely identify it as the XFDU Manifest and to differentiate it from any other XML-formatted file that might be in XFDU Package.

NOTE – Use of the '.xfdu' extension instead of '.xml' will require XFDU Package parsers to recognize files with the .xfdu extension as XML files and perhaps to convert the extension before submitting the file to a standard XML parser.

3.3.1.3 Payload Data File Names

The rules for forming names of XFDU Package–enclosed payload data files shall be defined by the application using TGFT.

3.3.1.4 Metadata File Names

The rules for forming names of XFDU Package–enclosed metadata files shall be defined by the application using TGFT, or by the authorities that control the creation and storage of external metadata files, as appropriate.

3.3.2 FILE PACKAGING

3.3.2.1 A file, or related group of files, shall be packaged as one physical XFDU Package.

NOTE – Construction of the XFDU Package in the context of TGFT is defined in section 4.

3.3.2.2 The physical XFDU Package shall be constructed as a computer file system folder/subdirectory, compressed to a single file using either TAR or ZIP.

NOTE – The specification of whether TAR and/or ZIP is used is deferred to the application that uses TGFT, or to the service agreement, as appropriate.

3.3.2.3 The XFDU Manifest within the (zipped/tarred) XFDU Package shall be an XML formatted document, in accordance with 4.3.1.2.

3.3.3 DIRECTORY STRUCTURE

3.3.3.1 There shall be one account per service agreement.

3.3.3.2 There shall be one directory for the account that serves as the *in-tray*.

NOTE – This directory is used as the target directory for files pushed to the TGFT Receiver. It is assumed that once a file has been found in the *in-tray*, it will be moved to another location for further processing by the file recipient.

3.4 DISCUSSION

3.4.1 PROCESSING

Any processing that is required to be carried out on any of the transferred files is defined by the file-recipient application process and is beyond the scope of this Recommended Standard.

It is, however, intended that the associated metadata that is delivered with a payload data file or group of payload data files (or metadata that is referenced by the XFDU Manifest in the XFDU Package that contains those payload data files) be sufficient to allow any required processing to be carried out. With this in mind, an extensible approach for specifying parameters in the XFDU Manifest metadata is defined in section 4.

3.4.2 MISSING FILES

No mechanism is defined in TGFT to detect missing files. However, the ability to check for missing files is supported by the metadata contained in the XFDU Manifest.

3.4.3 SECURITY

3.4.3.1 General

This TGFT Recommended Standard does not deal explicitly with mechanisms for encrypting or signing (checksum) the files transferred. It does, however, provide certain ‘hooks’ where these can be added. Full details of what is possible via the XFDU mechanism in terms of checksums and encryption is contained in reference [3].

A brief overview of encryption and signing possibilities is given below.

3.4.3.2 Signing of Encapsulated Files

The signing of enclosed files with a checksum is supported via the XFDU Data Object Section (see reference [3] and annex subsections F6.4.2 and F6.4.3 of this document).

NOTE – Any information related to the hash functions used, etc., is assumed to be defined either

a) in the definition of any service requiring that signing of encapsulated files be supported, or

b) by the parties involved in the file transfers by means of an ICD or similar mechanism.

3.4.3.3 Encryption of Encapsulated Files

The encryption of enclosed files is supported via the XFDU Data Object Section (see reference [3] and annex subsections F6.4.2 and F6.4.3 of this document).

NOTE – Any information related to the encryption and/or compression used, etc., is assumed to be defined either

a) in the definition of any service requiring that encryption and/or compression of encapsulated files be supported, or

b) by the parties involved in the file transfers by means of an ICD or similar mechanism.

3.4.3.4 Signing of Manifest File

The signing of the manifest file with a hash value is not supported.

3.4.3.5 Encryption of Manifest Files

The encryption of the manifest file is not supported.

3.4.3.6 Signing of XFDU Package Files

Signing of the XFDU Package files with hash value is supported. The mechanism for delivering the hash value of the XFDU Package file is to incorporate it into the file name as a hexadecimal value, as specified in 3.3.1.1.3.

NOTE – Any information related to the hash functions used, etc., is assumed to be defined either

a) in the definition of any service requiring that signing of XFDU Package files be supported, or

b) by the parties involved in the file transfers by means of an ICD or similar mechanism.

3.4.3.7 Encryption of XFDU Package Files

3.4.3.7.1 Encryption of an XFDU Package file is not explicitly covered by this Recommended Standard but can readily be achieved by simply encrypting the completed XFDU Package file.

3.4.3.7.2 If this is done, it is recommended that the extension of the encrypted file should make it clear to the involved parties that it is an encrypted file (in some cases encryption software will, when encrypting a file, produce a file name with a different file extension).

3.4.3.7.3 If it is required that an XFDU Package file be both signed and encrypted, it is recommended that the file first be signed (as per 3.4.3.6) and then encrypted.

NOTE – Any information related to the encryption and/or compression used, etc., is assumed to be defined either

a) in the definition of any service requiring that encryption and/or compression of XFDU Package files be supported, or

b) by the parties involved in the file transfers by means of an ICD or similar mechanism.

4 COMPOSITION RULES FOR XFDU PACKAGES FOR TGFT

4.1 OVERVIEW

This section contains the specification of the composition rules for XFDU Packages for TGFT.

For clarity, the TGFT XFDU composition rules are expressed in terms of the XFDU Package, the XFDU Manifest and its XML entities (elements, attributes, etc.), payload data files, and metadata files.

NOTE – The TGFT XFDU composition rules ignore components of the CCSDS-standard XFDU that are not part of the TGFT XFDU (e.g., the Behavior Section). Developers of a TGFT-using application that may wish to extend the definition of the TGFT XFDU for the purposes of that application should do so in conformance with the provisions, capabilities, and composition rules specified in the CCSDS-standard XFDU Blue Book (reference [3]).

For TGFT, a payload data file is a file that contains essential data for the XFDU Package, that is, data that is the motivation for creation of the XFDU Package. Each payload data file contained in the XFDU Package is represented by one **dataObject** element in the XFDU Manifest (see 4.3.7). By default, every file in an XFDU Package is a payload data file unless it is the Manifest file or a metadata file.

For TGFT, a metadata file is a file that is included in the XFDU Package (or referenced by the XFDU Manifest as an external metadata file) only because it contains data that is necessary the interpretation or processing of a payload data file within the XFDU Package. Each metadata file contained in the XFDU Package is represented by one **metadataObject** element in the XFDU Manifest (see 4.3.6).

4.2 GENERAL

Other than the XFDU Manifest file itself, every file in the XFDU Package must be represented by either a **dataObject** element or a **metadataObject** element in the XFDU Manifest.

4.3 XFDU MANIFEST

4.3.1 GENERAL

4.3.1.1 The rules for naming of XFDU Manifest files are specified in 3.3.1.2.

4.3.1.2 The XFDU Manifest shall conform to the XFDU XML schema specified in section 11 of reference [3], with further restrictions and refinements as specified in 4.3.2 through 4.3.7, below.

NOTE – Optional elements/attributes of the CCSDS-standard XFDUType that have no anticipated use for TGFT-using applications are ignored.

4.3.2 XFDU MANIFEST ATTRIBUTES

4.3.2.1 The optional textInfo attribute of the XFDU element may be used to carry TGFT-using application-specific information.

4.3.2.2 The optional version attribute shall be absent when the XFDU schema defined in Issue 1 (September 2008) of reference [3] is used as the schema for TGFT XFDUs.

NOTE – As of this writing, there is only one version of the XFDU XML schema, that which is defined in Issue 1 (September 2008) of reference [3]. If and when additional versions of the XML schema become available in the future, and assuming that by then schema naming and versioning rules will be in place to unambiguously define the corresponding contents of the version attribute, the version attribute will also be required in TGFT XFDUs to specify which version is being used.

4.3.3 XFDU MANIFEST CONTENTS

The XFDU Manifest shall contain

a) one instance of the packageHeader element;

b) one instance of the informationPackageMap element;

c) zero or one instance of the metadataSection element: the metadataSection element shall be present only if there are one or more metadataObject elements associated with the payload data file(s) being transferred in the XFDU Package; and

d) one instance of the dataObjectSection element.

4.3.4 packageHeader ELEMENT

4.3.4.1 packageHeader Attributes

The ID attribute of the packageHeader element shall have either

a) a value that is relevant to the TGFT-using application; or

b) the value ‘pkgHdr’.

4.3.4.2 packageHeader Contents

4.3.4.2.1 General

The packageHeader element shall contain one volumeInfo element and one or more environmentInfo elements.

4.3.4.2.2 volumeInfo Element

The volumeInfo element shall contain one specificationVersion element and zero or one sequenceInformation elements:

a) The specificationVersion element shall have the value '1.0'.

b) The sequenceInformation element carries no value itself; all information is carried by its attributes:

   1) The sequencePosition attribute is intended to be used to contain the numerical position of the XFDU in the XFDU sequence. The semantics of this attribute are deferred to the specification of the application.

   2) The sequenceSize attribute is intended to be used to contain size of the XFDU sequence to which this XFDU belongs. The semantics of this attribute are deferred to the specification of the application.

4.3.4.3 environmentInfo Element(s)

One **environmentInfo** element shall contain one **extension** element for the TGFT XFDU extension parameters that are applicable to the XFDU Package as a whole. This **environmentInfo** element shall not contain any **xmlData** elements:

NOTE – The CCSDS-standard **environmentInfo** element contains zero or one extension elements and zero to unbounded **xmlData** elements. Although other **environmentInfo** elements in the TGFT XFDU Manifest may contain TGFT-using-application-specific metadata in **xmlData** elements (see NOTE under b)), **xmlData** elements are prohibited in the **environmentInfo** element that contains the mandatory TGFT XFDU extension parameters.

a) The **extension** element for the TGFT XFDU extension parameters shall contain one **tgftXfduExtension** element, cast as an instance of the **TgftXfduExtensionType** complex schema type, which is registered in the 'urn:ccsds:schema:tgft:xfdu_extensions' namespace in the file TgftXfduExtensionParameters.xsd.

NOTE – The **TgftXfduExtensionType** complex schema type contains two metadata parameters: originator and recipient.

b) If a TGFT-using application has application-specific metadata that applies to the XFDU Package as a whole, zero or more environmentInfo elements, each containing an extension of the XFDU schema in the extension component element, should be used to contain the application-specific metadata.

NOTE – A TGFT-using application may also include application-specific metadata in zero or more **environmentInfo** elements**,** each of which contains one or more **xmlData** elements containing freeform XML. The specification of application-specific **environmentInfo** content, whether in the form of schema types for the **extension** element, freeform XML for the **xmlData** element, or some combination of both, is deferred to the specifications of those applications.

4.3.5 informationPackageMap ELEMENT

4.3.5.1 informationPackageMap Attributes

4.3.5.1.1 The informationPackageMap element shall contain the identifier of the TGFT-using application with which the XFDU is associated (e.g., 'ForwardFileService'—see reference [H8] and annex E):

4.3.5.1.2 For CCSDS-standard applications, the identifier is defined by the specification of TGFT-using application and registered in packageType SANA Registry (see C3.2).

4.3.5.1.3 For locally defined TGFT-using applications, the identifiers are specified in the appropriate service-agreement-level documentation.

4.3.5.2 informationPackageMap Contents

4.3.5.2.1 General

The informationPackageMap element shall contain one contentUnit element for each Data Object (i.e., payload data file) contained within the XFDU Package:

4.3.5.2.2 contentUnit Attributes

4.3.5.2.2.1 The optional ID attribute shall be present in each contentUnit element of an XFDU Manifest with more than one contentUnit elements. Each contentUnit ID attribute shall have a unique value within the scope of the XFDU Manifest.

4.3.5.2.2.2 The optional order attribute may be used to represent the order in which the contentUnit element occurs in a sequence of multiple contentUnit elements within the XFDU Manifest. The default syntax for this attribute is the string representation of the sequence number of the content unit, starting with '1' and incrementing by single digits (e.g., '2', '3', etc.) such that the order attribute of the last content unit in the sequence has the string value

equivalent of the number of payload data files in the XFDU Package. However, a TGFT-using application may specify and use an alternative syntax.

4.3.5.2.2.3 The optional unitType attribute may be used to carry TGFT-using application specific information.

4.3.5.2.2.4 The optional textInfo attribute of the contentUnit element may be used to carry TGFT-using application specific information.

4.3.5.2.2.5 The optional anyMdID attribute of the contentUnit element shall be present if the XFDU Manifest contains a metadataSection element (4.3.6) that contains any metadataObject elements that correspond to metadata files that are needed to interpret or process the payload data file that is represented by the contentUnit. If present, this attribute shall contain the IDREF values of the ID attributes of the metadataObject elements (see 4.3.6.2.1 [4.3.6.2.1.1]) that correspond to those related metadata files.

4.3.5.2.3 contentUnit Contents

4.3.5.2.3.1 General

The contentUnit element shall contain one dataObjectPointer element, and zero or one extension element.

4.3.5.2.3.2 dataObjectPointer Element

4.3.5.2.3.2.1 dataObjectPointer Attribute

The **dataObjectID** attribute shall contain the value of the **ID** attribute of the **dataObject** element (4.3.7.2.1 [4.3.7.2.1.1]) to which the **contentUnit** corresponds.

4.3.5.2.3.2.2 extension Element

NOTE – The **extension** element of the **contentUnit** is used in TGFT XFDUs to contain TGFT-specific metadata specific to the individual Content Unit.

The extension element of the contentUnit shall contain one tgftContentUnitExtension element.

4.3.5.3 tgftContentUnitExtension Element

4.3.5.3.1 The **tgftContentUnitExtension** element shall be cast as an instance of the **TgftContentUnitExtensionType** complex schema type, which is registered in the ‘urn:ccsds:schema:tgft:xfdu_extensions’ namespace in the file TgftXfduExtensionParameters.xsd.

NOTE – The **TgftContentUnitExtensionType** complex schema type contains one metadata parameter: creationDate.

4.3.5.3.2 If the TGFT-using application has application-specific metadata that is applicable to individual Content Units, the serviceSpecficContentUnitExt extension element of the tgftContentUnitExtension element shall contain an instance of a data type that contains that metadata, cast as a data type that is registered in a namespace that is associated with that application. The specification of serviceSpecficContentUnitExt content is deferred to those applications.

4.3.6 metadataSection ELEMENT

4.3.6.1 General

The metadataSection element shall contain zero or more metadataObject elements.

NOTE – Each **metadataObject** element defines and describes a metadata file that is associated with one or more of the payload data file(s) being transferred in the XFDU Package.

4.3.6.2 metadataObject Element

4.3.6.2.1 metadataObject Attributes

4.3.6.2.1.1 The ID attribute shall contain the unique ID of the metadataObject element.

4.3.6.2.1.2 The ID attribute shall be referred to by at least one anyMdID attribute in a contentUnit element (see 4.3.5.2.2, 4.3.5.2.2.5) within the same XFDU Manifest.

4.3.6.2.1.3 The category attribute may be used to specify the category of the metadata contained in the metadataObject element as suits the needs of the application using TGFT. If used, the value of this attribute shall be ‘OTHER’.

4.3.6.2.1.4 The classification attribute may be used to specify the classification of the metadataObject as suits the needs of the application using TGFT. If used, the value of this attribute shall be ‘OTHER’.

4.3.6.2.1.5 The optional otherClass attribute shall be present if and only if the classification attribute is present. The attribute shall contain the classification of the metadata represented by the metadataObject element. The definition of allowed classifications of metadata with respect to a given TGFT-using application is deferred to the specification of that application.

4.3.6.2.1.6 The optional otherCategory attribute shall be present if and only if the category attribute is present. This attribute shall contain the category of the metadata represented by the metadataObject element. The definition of allowed categories of metadata with respect to a given TGFT-using application is deferred to the specification of that application.

4.3.6.2.2 metadataObject Contents

4.3.6.2.2.1 General

The **metadataObject** element shall contain a **metadataReference** element that identifies the location of the corresponding metadata file.

4.3.6.2.2.2 metadataReference Attributes

4.3.6.2.2.2.1 The optional **textInfo** attribute of the **metadataReference** element may be used to carry TGFT-using application-specific information.

4.3.6.2.2.2.2 The **locatorType** attribute shall be set to 'URL'.

4.3.6.2.2.2.3 The **href** attribute shall contain the URL of the file:

a) If the metadata file is contained within the same XFDU Package as the XFDU Manifest, the URL of the file shall be of the form

'file:'+< package folder name part>
+ '/' + <metadata file name (with file type extension>,

where the *package folder name part* of the XFDU file name is defined in 3.3.1.1.4.

NOTES

1 The package folder name part of the XFDU Package file name is the name of the folder/subdirectory that contains the various files that constitute the XFDU Package. The package folder name part is formed according to naming rules defined by the application that generates the XFDU Package file.

2 The file type extension on the metadata file name is the file type of the metadata file itself. This is not to be confused with the file type extension ('.zip' or '.tar') of XFDU Package file name (see 3.3.1.1).

b) If the metadata file is external to the XFDU Package, the URL shall point to the external location of the file.

4.3.6.2.2.2.4 The optional **vocabularyName** attribute may be used if appropriate to the metadata type(s) used by a TGFT-using application. Identification/definition of appropriate metadata vocabulary names is deferred to the specification of the TGFT-using application.

4.3.6.2.2.2.5 The optional **mimeType** attribute may be used if appropriate to the metadata type(s) used by a TGFT-using application. Identification/definition of appropriate metadata MIME types is deferred to the specification of the TGFT-using application.

4.3.7 dataObjectSection ELEMENT

4.3.7.1 General

The **dataObjectSection** element shall contain one or more dataObject elements.

NOTE – Each **dataObject** element defines and describes a payload data file in the XFDU Package.

4.3.7.2 dataObject Element

4.3.7.2.1 dataObject Attributes

4.3.7.2.1.1 The **ID** attribute shall contain the ID of the Data Object.

4.3.7.2.1.2 The optional **mimeType** attribute may be present if appropriate to the Data Object that is transferred by the individual TGFT-using application, as specified for that application.

4.3.7.2.1.3 The **size** attribute shall be present and contain the size of the Data Object.

4.3.7.2.2 dataObject Contents

4.3.7.2.2.1 The dataObject element shall contain one byteStream element.

4.3.7.2.2.2 The dataObject element may contain a checksum element.

4.3.7.3 byteStream Element

The byteStream element shall contain a fileLocation element.

4.3.7.4 fileLocation Element

4.3.7.4.1 fileLocation Attributes

4.3.7.4.1.1 The optional **textInfo** attribute of the **fileLocation** element may be used to carry TGFT-using application specific information.

4.3.7.4.1.2 The **locatorType** attribute shall be set to ‘URL’.

4.3.7.4.1.3 The **href** attribute shall contain the URL of the file, which shall be of the form

‘file:’ + <package folder name part>

+ ‘/’ + <payload file name (with file type extension)>,

where the *package folder name part* of the XFDU Package file name is defined in 3.3.1.1.4.

NOTES

1 The package folder name part of the XFDU Package file name is the name of the folder/subdirectory that contains the various files that constitute the XFDU Package. The package folder name part is formed according to naming rules defined by the application that generates the XFDU Package file.

2 The file type extension on the payload file name is the file type of the payload file itself. This is not to be confused with the file type extension ('.zip' or '.tar') of XFDU Package file name (see 3.3.1.1).

4.4 PAYLOAD DATA FILES ENCLOSED WITHIN THE XFDU PACKAGE

4.4.1 The content of the payload data file(s) shall be defined by the application using TGFT.

4.4.2 The rules for forming names of XFDU Package-enclosed payload data files are specified in 3.3.1.3.

4.5 METADATA FILE ENCLOSED WITHIN THE XFDU PACKAGE

4.5.1 The content of the metadata data file(s) shall be defined by the application using TGFT.

4.5.2 The rules for forming names of XFDU Package-enclosed metadata files are specified in 3.3.1.4.

4.6 TGFT XFDU PACKAGE

4.6.1 Each TGFT XFDU Package shall contain one XFDU Manifest XML document, as defined in 4.3.

4.6.2 The TGFT XFDU Package shall contain one file for every dataObject element in the dataObjectSection of the XFDU Manifest.

4.6.3 The TGFT XFDU Package shall contain one file for every metadataObject element in the metadataSection of the XFDU Manifest for which the metadataReference element has an href value that begins, 'file:'.

4.6.4 Other than the XFDU Manifest, each file in the TGFT XFDU Package shall correspond to one and only one of either (a) a dataObject element in the XFDU Manifest or

(b) a metadataObject element in the XFDU Manifest for which the metadataReference element has an href value that begins, ‘file:’.

4.6.5 The rules for forming the names of XFDU Packages are specified in 3.3.1.1.

ANNEX A

IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

This annex provides the Implementation Conformance Statement (ICS) Requirements List (RL) for an implementation of the *Terrestrial Generic File Transfer*. The ICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation shall satisfy the mandatory conformance requirements referenced in the RL.

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

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

A1.2 ABBREVIATIONS AND CONVENTIONS

A1.2.1 General

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

A1.2.2 Item Column

The item column contains sequential numbers for items in the table.

A1.2.3 Feature Column

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

NOTE – The features itemized in the RL are elements of the TGFT. Therefore support for a mandatory feature indicates that this feature will be used, and support for an optional feature indicates that this feature can be used.

A1.2.4 Reference Column

The reference column indicates the relevant subsection or table in the *Terrestrial Generic File Transfer* Recommended Standard (this document).

A1.2.5 Status Column

The status column uses the following notations:

M mandatory.

O optional.

It should be noted that a parameter may be marked as M(andatory), while the class that contains it is marked O(ptional). This should be interpreted to mean that while the class is optional, if it is present, then the parameter must be present.

A1.2.6 Support Column Symbols

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

Y Yes, supported by the implementation.

N No, not supported by the implementation.

n/a Not applicable.

A1.3 INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the Recommended Standard by completing the RL; that is, the state of compliance with all mandatory requirements and the options supported shown. The resulting completed RL is called an ICS. The implementer shall complete the RL by entering appropriate responses in the support or values supported column, using the notation described in A1.2. If a conditional requirement is inapplicable, n/a should be used. If a mandatory requirement is not satisfied, exception information must

be supplied by entering a reference *Xi*, where *i* is a unique identifier, to an accompanying rationale for the noncompliance.

A2 ICS PROFORMA FOR TERRESTRIAL GENERIC FILE TRANSFER

A2.1 GENERAL INFORMATION

A2.1.1 Identification of ICS

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

A2.1.2 Identification of Implementation Under Test

Implementation name	
Implementation version	
Special Configuration	
Other Information	

A2.1.3 Identification of Supplier

Supplier	
Contact Point for Queries	
Implementation Name(s) and Versions	
Other Information necessary for full identification; for example, names(s) and version(s) for machines and/or operating systems;	

A2.1.4 Document Version

CCSDS 927.1-R-1, Issue 1	
Have any exceptions been required? (Note: A YES answer means that the implementation does not conform to the Recommended Standard. Non-supported mandatory capabilities are to be identified in the ICS, with an explanation of why the implementation is non-conforming	Yes _____ No _____

A2.1.5 Requirements List

A2.1.5.1 File Packaging

A2.1.5.1.1 General

Item	Feature	Ref.	Status	Support
1.	TAR	3.2.5	O	
2.	ZIP	3.2.5	O	

A2.1.5.2 Signing and Encryption

A2.1.5.2.1 General

Item	Feature	Ref.	Status	Support
3.	Signing of Encapsulated Files	3.4.3.2	O	
4.	Encryption of Encapsulated files	3.4.3.3	O	
5.	Signing of XFDU Package Files	3.4.3.6	O	
6.	Encryption of XFDU Package Files	3.4.3.7	O	

ANNEX B

SERVICE AGREEMENT

(NORMATIVE)

B1 SERVICE AGREEMENT

B1.1 GENERAL

The following constitutes the service agreement for the TGFT.

NOTE – It is assumed that the Service Agreement will be specific to a mission.

Table B-1: Service Agreement

Agency 1 Details	
Agency Name	
Service Agreement Points of Contact (see B1.2 below)	
Troubleshooting Points of Contact (see B1.2 below)	
Hosts names and roles (TGFT Receiver only) (see B1.3 below)	
Login Information (see B1.4 below)	
Data Volume (see B1.5 below)	
Retention Time (see B1.6 below)	
Availability (see B1.7 below)	
Transfer Rate (see B1.8 below)	
Delivery Latency (see B1.9 below)	

Agency 2 Details	
Agency Name	
Service Agreement Points of Contact (see B1.2 below)	
Troubleshooting Points of Contact (see B1.2 below)	
Hosts names and roles (TGFT Receiver only) (see B1.3 below)	
Login Information (see B1.4 below)	
Data Volume (see B1.5 below)	
Retention Time (see B1.6 below)	
Availability (see B1.7 below)	
Transfer Rate (see B1.8 below)	
Delivery Latency (see B1.9 below)	

B1.2 POINTS OF CONTACT

Points of contact would be needed potentially at 2 levels:

a) to identify the personnel at both sides who are responsible for establishing the service agreement;

b) to identify the personnel who should be notified in the event of problems arising.

B1.3 HOSTS

This should identify the machine(s) that the agency will host the TGFT Receiver(s) by name or IP address and describe the failover options available, that is, dynamic DNS, Prime and Backup (in this case, stating whether files should always be copied to the backup as well as the prime or only copied to the backup when the prime as failed, no backup, etc.).

B1.4 LOGIN INFORMATION

This should be used to specify the user account to be used and specify the mechanism by which the required authentication details (i.e., password or keys) should be distributed.

B1.5 DATA VOLUME

This should specify the total data volume that is available to the remote agency, that is, the total storage space available in the *in-tray*.

B1.6 RETENTION TIME

This should specify how long files will be retained before being deleted in the event that it is not possible to deliver the file to the TGFT Receiver. This could be global or per file type.

B1.7 AVAILABILITY

This should specify the availability of the hosts to be used for the file transfers and indicate times during which support for troubleshooting will be available.

B1.8 TRANSFER RATE

This should specify the transfer rates that can be supported between the 2 parties involved. May not be applicable in all cases, for example, where the transfer takes place over the internet.

B1.9 DELIVERY LATENCY

This should specify the expected latency between a file's becoming available at the TGFT Sender and it's being transferred to the TGFT Receiver.

ANNEX C

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

C1 SECURITY CONSIDERATIONS

C1.1 OVERVIEW

This annex presents the results of an analysis of security considerations applied to the technologies specified in this Recommended Standard.

C1.2 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

The consequences of not applying security to the systems and networks on which this Recommended Standard is implemented could include potential loss, corruption, and theft of data. Since it is possible to utilize these messages in preparing and disseminating schedules relating to the availability of communications and tracking resources for spacecraft, the consequences of not applying security to the systems and networks on which this Recommended Standard is implemented could include compromise or loss of the mission if malicious tampering of a particularly severe nature occurs.

C1.3 POTENTIAL THREATS AND ATTACK SCENARIOS

Potential threats or attack scenarios include, but are not limited to, (a) unauthorized access to the programs/processes that generate and interpret the messages and (b) unauthorized access to the messages during transmission between exchange partners. Protection from unauthorized access during transmission is especially important if the mission utilizes open ground networks such as the Internet to provide ground station connectivity for the exchange of data formatted in compliance with this Recommended Standard. It is strongly recommended that potential threats or attack scenarios applicable to the systems and networks on which this Recommended Standard is implemented be addressed by the management of those systems and networks and the utilization of adequate authentication, suitable protocols, and secured interfaces for the exchange of this information.

C1.4 SECURITY CONCERNS RELATED TO THIS RECOMMENDED STANDARD

C1.4.1 Data Privacy

Privacy of data formatted in compliance with the specifications of this Recommended Standard should be assured by the systems and networks on which this Recommended Standard is implemented.

C1.4.2 Data Integrity

Integrity of data formatted in compliance with the specifications of this Recommended Standard should be assured by the systems and networks on which this Recommended Standard is implemented.

C1.4.3 Authentication of Communicating Entities

Authentication of communicating entities involved in the transport of data that complies with the specifications of this Recommended Standard should be provided by the systems and networks on which this Recommended Standard is implemented.

C1.4.4 DATA TRANSFER BETWEEN COMMUNICATING ENTITIES

The transfer of data formatted in compliance with this Recommended Standard between communicating entities should be accomplished via secure mechanisms approved by the Information Technology Security functionaries of exchange participants.

The use of the HTTPS protocol will require that a suitable HTTPS server is installed on the TGFT Receiver system. Some Web servers allow the disabling of HTTP methods. Therefore depending on the security needs of the agencies involved, it may be desirable to configure the Web server used such that only the required HTTP methods are available.

C1.4.5 Control of Access to Resources

Control of access to resources should be managed by the systems upon which provider formatting and recipient processing are performed.

C1.4.6 Auditing of Resource Usage

Auditing of resource usage should be handled by the management of systems and networks on which this Recommended Standard is implemented.

C2 UNAUTHORIZED ACCESS

Unauthorized access to the programs/processes that generate and interpret the messages should be prohibited in order to minimize potential threats and attack scenarios.

C2.1 DATA SECURITY IMPLEMENTATION SPECIFICS

Specific information-security interoperability provisions that apply between agencies and other independent users involved in an exchange of data formatted in compliance with this Recommended Standard should be specified in an ICD.

C3 SANA CONSIDERATIONS

C3.1 GENERAL

The recommendations of this document rely on the SANA registries described below. New assignments in these registries, in conformance with the policies identified, will be available at the SANA registry Web site: http://sanaregistry.org. Therefore the reader shall look at the SANA Web site for all the assignments contained in these registries.

C3.2 REGISTRY FOR `packageTypes`

SANA is requested to register and maintain `packageTypes` values in a registry located at:

http://sanaregistry.org/r/[location to be determined].

The structure of this registry is as shown in table C-1.

Table C-1: packageTypes Registry Structure

`Field`	Type	Description
PackageType	string	Specifies the package type
Description	String	Description of the package file

In accordance with reference [H7], subsection 3.11 c), updates to this registry are to be at the discretion of CCSDS member agencies or registered organizations, via the registered agency or organization representative.

C4 PATENT CONSIDERATIONS

No patent rights are known to adhere to any of the specifications of the Recommended Standard.

ANNEX D

EXAMPLE XFDU MANIFEST

(INFORMATIVE)

In this example, a hypothetical Validated Radiometric Data cross support service generates XML-formatted Tracking Data Messages (TDMs) in conformance with the XML Specification for Navigation Data Messages Blue Book (reference [7]). This hypothetical service collects all tracking data for a space link session, performs validation processing, and transfers the resulting TDM XML document as a file over TGFT. (This hypothetical service is different from the Tracking Data Cross Support Transfer Service, which transfer samples of tracking data in near-real time.)

The Validated Radiometric Data service generates an XFDU Package ZIP file containing an XFDU Manifest file (XML document) and a separate XML document containing the TDM itself.

The metadata for this XML document consists of the XML schema file for the TDM, which is located in the SANA Registry at

http://www.sanaregistry.org/r/ndmxml/ndmxml-1.0-tdm-1.0.xsd.

The XFDU Package is represented as a folder containing the XFDU Manifest and the TDM data file. This folder has the name 'dss_25_validated_tdm_xfdu_package'. When the folder is zipped to form the resultant XFDU Package file, that file has the name 'dss_25_validated_tdm_xfdu_package.zip'.

When the TGFT Sender begins transferring the file to the TGFT Receiver on 27 February 2017 at 23:15:46Z, the TGFT Sender inserts the modified CCSDS Time Code-B formatted timestamp into the file name to uniquely identify this transmitted file:

dss_25_validated_tdm_xfdu_package-2017-058T23-15-46Z.zip.

NOTE – The file naming convention used in this example for files and packages generated by the Validated Radiometric Data service has been contrived for the purposes of providing example file names.

The XML TDM file contained within the XFDU Package is named

dss_25_validated_tdm-2017-02-27T19-35-24Z.xml,

where timestamp substring of the file name indicates the date (27 February 2017) and time (19:35:24Z) at which the tracking data stopped being acquired. It should be noted that this timestamp follows the `XSD:datetime` format and is presumed (for example purposes) to be part of the hypothetical Validated Radiometric Data service-specific file naming convention.

Thus the **href** attribute of the dataObject fileLocation element is

file:dss_25_validated_tdm_xfdu_package/dss_25_validated_tdm-2017-02-27T19-35-24Z.xml,

which indicates that the file is located within the dss_25_validated_tdm_xfdu_package folder.

Although the raw data stopped being acquired at 2017-058T19-35-24Z, the validation process took several hours; the *validated* TDM file was created at 2017-058T22:09:37. This is the **creationDate** for the validated TDM file contained in the XFDU Package.

NOTE – The Validated Radiometric Data service could also add application-specific metadata by extending the **serviceSpecificContentUnitExt** element of the **TgftContentUnitExtensionType** schema type. Examples of such metadata could be the beginning and end times of the acquisition of the raw radiometric data that was subsequently processed to create the validated data file.

Figure D-1 is the XFDU Manifest XML document for this example XFDU Package.

```
<?xml version="1.0" encoding="UTF-8"?>
<xfdu:XFDU textInfo="Example of TGFT XFDU Manifest" xsi:schemaLocation="urn:ccsds:schema:xfdu:1 TGFTXFDUschema.xsd"
xmlns:tgft="urn:ccsds:schema:tgft:xfdu_extensions" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xfdu="urn:ccsds:schema:xfdu:1">
  <packageHeader ID="PkgHdr">
    <volumeInfo>
      <specificationVersion>1.0</specificationVersion>
    </volumeInfo>
    <environmentInfo>
      <extension>
        <tgft:tgftXfduExtension>
          <originator>dss-25</originator>
          <recipient>[Xenosat MOC identifier]</recipient>

        </tgft:tgftXfduExtension>
      </extension>
    </environmentInfo>
  </packageHeader>
  <informationPackageMap textInfo="Validated Radiometric Data Info Pkg Map"  packageType="ValidatedRadiometricData">
    <xfdu:contentUnit ID="TDM_1_ContentUnit" textInfo="content unit for TDM XML document" anyMdID="TDM_Schema"  unitType="TDM">
      <extension>
        <tgft:tgftContentUnitExtension>
          <creationDate>2017-058T22:09:37 </creationDate>
        </tgft:tgftContentUnitExtension>
      </extension>
      <dataObjectPointer dataObjectID="TDM_1"/>
    </xfdu:contentUnit>
  </informationPackageMap>
  <metadataSection>
    <metadataObject ID="TDM_Schema" otherClass="Schema" classification="OTHER">
      <metadataReference textInfo="Reference to repository of TDM schema" locatorType="URL" href="http://www.sanaregistry.org/r/ndmxml/ndmxml-1.0-tdm-1.0.xsd"
mimeType="application/xml"/>
    </metadataObject>
  </metadataSection>
  <dataObjectSection>
    <dataObject ID="TDM_1" size="214748" mimeType="application/xml">
      <byteStream>
        <fileLocation locatorType="URL" href="file:dss_25_validated_tdm_xfdu_package-2017-058T23-15-46Z/dss_25_validated_tdm-2017-02-27T19-35-24Z"/>
      </byteStream>
      <checksum checksumName="payload checksum">[payload checksum value]</checksum>
    </dataObject>
  </dataObjectSection>
</xfdu:XFDU>
```

Figure D-1: Example XFDU Manifest

Figure D-2 is a graphical view of the example XFDU Manifest.

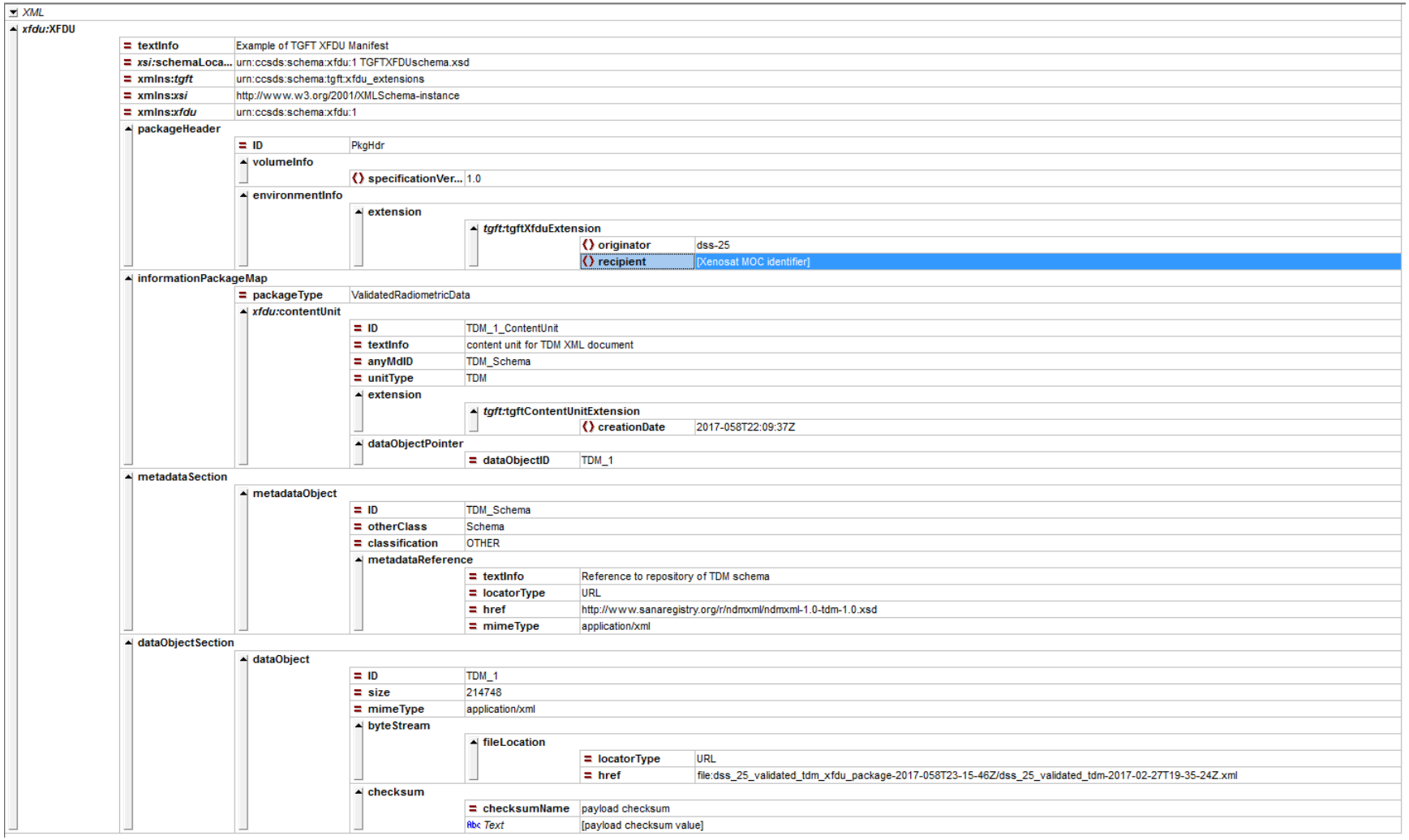

Figure D-2: Example XFDU Manifest—Grid View

ANNEX E

TGFT RELATION TO OTHER FILE RELATED SERVICES

(INFORMATIVE)

E1 TGFT IN RELATION TO OTHER FILE RELATED SERVICES

E1.1 INTRODUCTION

As defined in the normative sections of this Recommended Standard, TGFT is intended only for point-to-point file delivery in the terrestrial context. This, however, does not preclude the case in which the use of the TGFT is only part of the delivery chain, for example, when the final destination (or initial origin) of a file is the space segment.

E1.2 RELATION TO CFDP

One possible use of the TGFT is as part of a file delivery chain in which either the original source or final destination of a file is the space segment. In this case, what is required of TGFT is that it permits the specification of metadata such that information required to uplink the file to the space segment can be fully defined and, similarly, that the metadata associated with a file downlinked via CFDP can be encapsulated in the TGFT metadata and thus delivered along with the file.

It should be noted that this standard does not specify in any way how the process of injecting a file delivered over TGFT into CFDP should be achieved or vice versa. It is solely intended that the specification of the metadata is sufficiently flexible to permit this to be carried out by some mechanism.

E1.3 RELATION TO FILE-TRANSFER-BASED CROSS SUPPORT SERVICES

TGFT is used to exchange files between ground node application processes. These file-exchanging applications may be specific to a given agency or mission, or they may be the Provider and User entities of file-transfer-based Cross Support Services (CSSes) such as the proposed Forward File and Return File services (see reference [H8]).

When used in the context of CSSes, TGFT is not exposed directly to the end-users of the CSSes, but is rather an underlying protocol of those CSSes. It is left to the specification of each such CSS to define how the features and capabilities of TGFT (as specified in this Recommended Standard) are used by that service. This includes, but is not limited to, the specification of service-specific metadata and how that metadata is applied using the XFDU-based metadata mechanisms that are defined for TGFT.

ANNEX F

CONCEPT FOR USING XFDUS IN TGFT

(INFORMATIVE)

F1 XFDU AS APPLIED TO TGFT

The CCSDS XML-Formatted Data Unit (XFDU, reference [3]) standard has been selected as the packaging mechanism for payload data and associated metadata that is to be transferred using Terrestrial Generic File.

F2 BACKGROUND

The CCSDS-standard XFDU has been designed with data archiving in mind. It therefore has features and capabilities that support transferring data into an archive, storing that data with its associated metadata for timespans up to years or even decades, and eventually making that data available to users, possibly in groupings, formatting, and encodings of the data that are different from those that were used in the acquisition of the data by the archive, and possibly drawn from multiple physical data repositories.

The needs of TGFT are only a subset of those needed for the larger data acquisition/archival/dissemination problem that the CCSDS-standard XFDU is designed to address. The need exists to trim down the full set of XFDU capabilities to only those needed for the purposes of TGFT.

F3 USE OF XFDU BY TGFT

TGFT provides an underlying service to a class of applications by which the data exchanged between data sender and data receiver is in the form of a file, accompanied by the metadata necessary to interpret that file. Each TGFT entity is capable of performing two roles: the *TGFT Sender* and the *TGFT Receiver*. Because TGFT is a push-only protocol, the transmission of the file is always initiated by a TGFT entity operating in the TGFT Sender role. The triggering mechanism for the initiation of the file transfer is defined by the file sender application that is using TGFT. In cases in which the applications that use TGFT are CSSes, sometimes the Service Provider may be the file sender, and it uses TGFT to push the XFDU Package containing that data to the Service User (the file recipient), for example, in the case of the planned Forward File service. In other CSSes, the Service User may be the file sender, and it uses TGFT to push the XFDU Package containing that data to the Service Provider (the file recipient), for example, in the case of the planned Forward File service.

Figure F-1 illustrates the relationships among the various types of file-source and file-recipient applications, TGFT Sender and TGFT Receiver, and the in-tray directories with respect to two agencies that participate in bidirectional file transfer using TGFT.

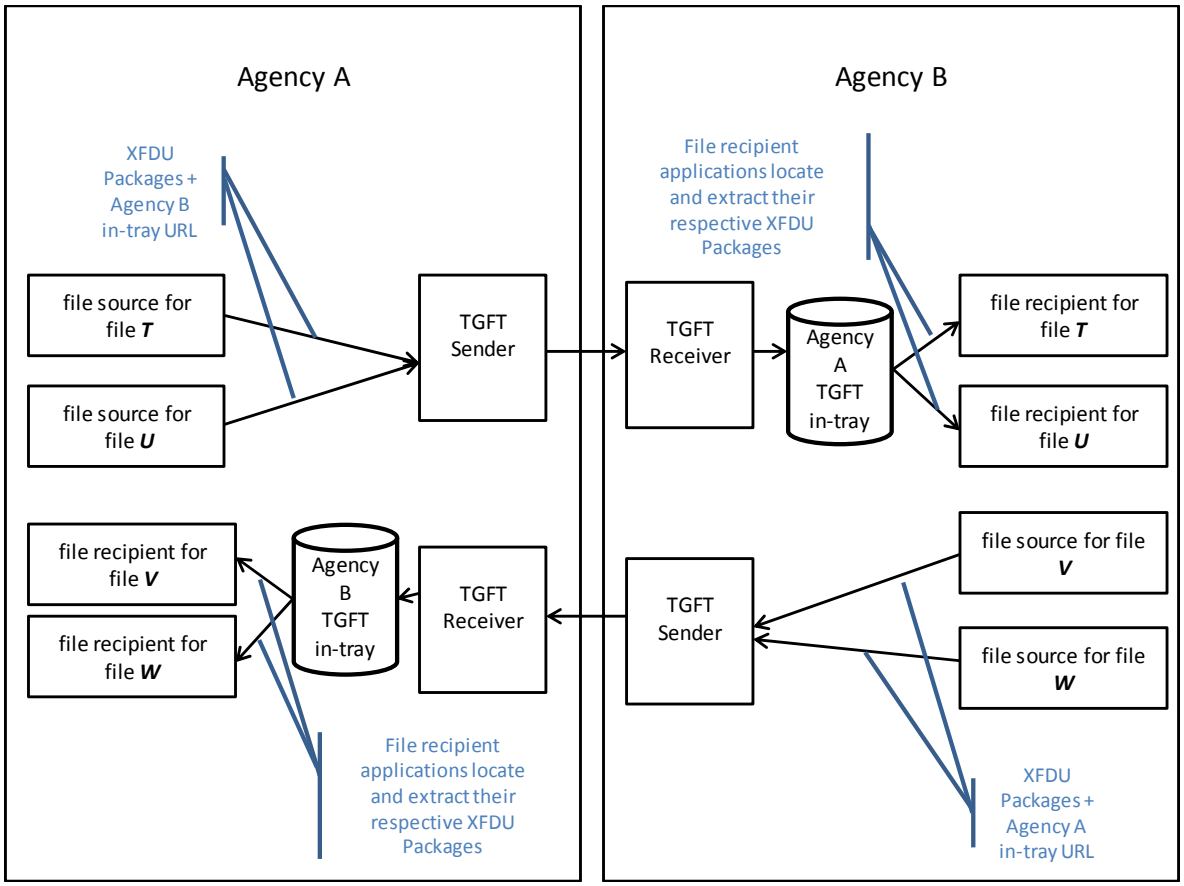

Figure F-1: TGFT-Related Entity Relationships

As illustrated in the figure, the file source applications create the XFDU Packages and submit them and their associated target in-tray URLs to their designated TFGT Sender. It should be noted that although the figure depicts only one TGFT Sender per agency, the deployment of the TGFT Sender functionality is not defined by this specification – it may be centralized per agency (as illustrated in the figure), or it may be distributed (e.g., an agency may have multiple TGFT Senders that are dedicated to individual file sources or shared by subsets of the file sources of the agency. Note, too, that the use of 'agency' in this context is a formalism; it is likely that, for practical reasons, an agency may have multiple facilities, each of which may host its own TGFT Receiver and one (or more) TGFT Sender entities.

In the destination node, the TGFT Receiver places those XFDU Packages in the designated in-tray (e.g., the agency-A TGFT Receiver puts all XFDU Packages received from agency B into the agency-B in-tray, and the agency-B TGFT Receiver puts all XFDU Packages received from agency A into the agency-A in-tray). The file recipient applications use the Package Type metadata (see 4.3.5.1, 4.3.5.1.1, and F6.3.2.1 b)) embedded in each XFDU Package to select which file recipient application is to process that XFDU Package.

NOTE – Although the figure illustrates the process of identifying which XFDU Packages belong to which file recipient application as being performed by the file recipient applications themselves, there is no requirement on implementations to do so in this way. For example, an implementation could include a common daemon process that searches through the in-tray and routes the XFDU Packages to their respective file recipient applications.

F4 DERIVATION OF THE TGFT XFDU FROM THE CCSDS-STANDARD XFDU

The CCSDS-standard XFDU (reference [3]) has been designed with data archiving in mind. It therefore has features and capabilities that support transferring data into an archive, storing that data with its associated metadata for timespans up to years or even decades, and eventually making that data available to users, possibly in groupings, formatting, and encodings of the data that are different from those that were used in the acquisition of the data by the archive, and possibly drawn from multiple physical data repositories.

The needs of TGFT are only a subset of those needed for the larger data acquisition/archival/dissemination problem that the CCSDS-standard XFDU is designed to address. For instance, much of the complexity of the CCSDS-standard XFDU results from maintaining data relationships of 'Data Objects' that are larger than individual files, whereas TGFT deals with individual payload data files and the metadata associated with it.

As a result, the TGFT XFDU is a subset of the CCSDS-standard XFDU, with the elements needed to support the terrestrial file transfer needs of CSSes and Provider CSSS/mission-specific applications.

The specification of XFDU capabilities for TGFT does not preclude TGFT-using applications from using XFDU mechanisms that are part of the CCSDS-standard XFDU but not formally part of the TGFT XFDU standard, either optionally or to suit additional requirements of those TGFT-using applications.

F5 SUMMARY OF XFDU CONCEPTS AND CAPABILITIES USED BY TGFT

This annex subsection provides an overview of the CCSDS-standard XFDU, the applicability of the various aspects of that XFDU, and the rationale for restrictions, modifications, and/or refinements for the purposes of TGFT.

In general, a CCSDS XFDU is a logical entity that comprises "an XFDU Manifest, all files contained in the manifest, and all files and XFDUs referenced from it. Some or all of the referenced files may be contained in an XFDU Package, such as through the use of a ZIP file. However, there may still be references in the Manifest to files outside the XFDU Package. In this case, the XFDU is a logical entity and does not exist as a single physical entity." (reference [3]).

Figure F-2 illustrates the top-level structure of the XFDU Package with the simplifications that apply to TGFT.

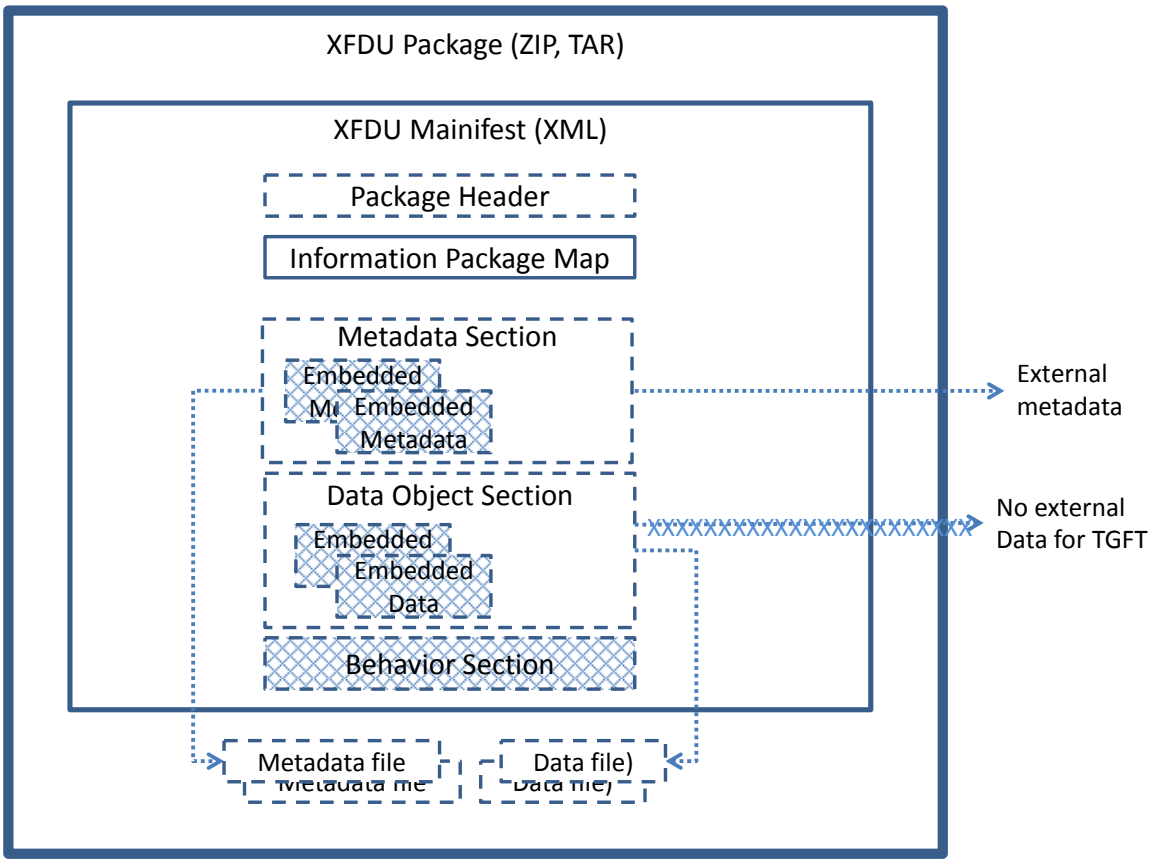

Figure F-2: Top-Level XFDU Structure with TGFT Modifications

TGFT puts a further constraint on the structure of an XFDU in that the data (hereinafter referred to as the *payload data* to differentiate it from metadata) must be carried within the XFDU Package that is being transferred via TGFT. That is, TGFT cannot be used to transfer mere pointers to payload data. This constraint is a consequence of TGFT being defined as a push-only mechanism.

The CCSDS-standard XFDU allows payload data to be carried in the XFDU Package in either of two ways: embedded in the XFDU Manifest itself, or as separate files that are contained in the ZIP/TAR XFDU Package and referenced by the XFDU Manifest. However, for TGFT, only the separate-payload-data-file(s)-within-the-XFDU-Package approach is supported.

Similar to payload data, the CCSDS-standard XFDU allows metadata to be carried in the XFDU Package in one of two ways: embedded in the XFDU Manifest itself, or as separate files that are contained in the ZIP/TAR XFDU Package and referenced by the XFDU Manifest. However, for TGFT, only the separate-metadata-file(s)-within-the-XFDU-Package approach is supported.

A TGFT-compliant XFDU Manifest may also contain references to metadata that exists outside the XFDU Package that contains the XFDU Manifest. The rationale is that some metadata (such as XML schema) may persist and be applicable to many instances of payload data, such that it is an inefficient use of resources to transfer the metadata with every instance of payload data transfer.

NOTE – The mechanism by which the file recipient accesses such external metadata is outside the scope of the TGFT specification. For some TGFT-using applications, the metadata access mechanism could simply be defined bilaterally. Other TGFT-using applications may define standard mechanisms by which metadata is made available to the file recipient, for example, by always pushing out new versions of the metadata whenever they become available. However, such application-specific mechanisms would be defined by the specification of those applications and not be part of TGFT. The specification of a TGFT-using application must address which of the two possible options (separate file(s) enclosed in the same XFDU Package or resident in an external repository) are to be used for the metadata in the XFDU Manifests and XFDU Packages for that application. Because it is possible for multiple metadata files to be associated with one payload data file, some TGFT-using applications may use both methods. For example, the payload data file may be formatted as an XML document using a standard (for that particular enterprise) XML schema, so the reference to that XML schema file would be an external reference. But the same payload file might also have metadata that applies to only that payload data file, so that metadata file would be enclosed in the same XFDU Package.

F6 ANALYSIS OF XFDU MANIFEST ELEMENT AND ATTRIBUTE APPLICABILITY TO TGFT

In this annex subsection, each of the components of the CCSDS-standard XFDU Manifest are described. For each of these components, the presence and/or use of the attributes and elements that constitute that component in TGFT XFDUs is defined. The TGFT-specific use and restrictions are highlighted by underlined text.

Figure F-3 is a graphic representation of the top-level CCSDS-standard XFDU Manifest schema. Figure F-4 is the corresponding diagram of only those top-level components that are used in TGFT.

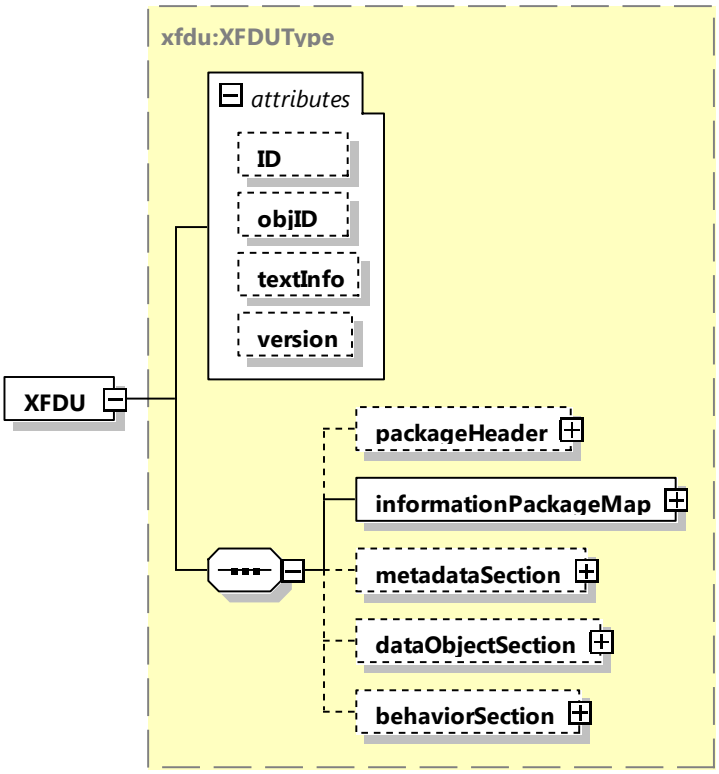

Figure F-3: CCSDS-Standard Top-Level XFDU Structure, in Graphical Representation

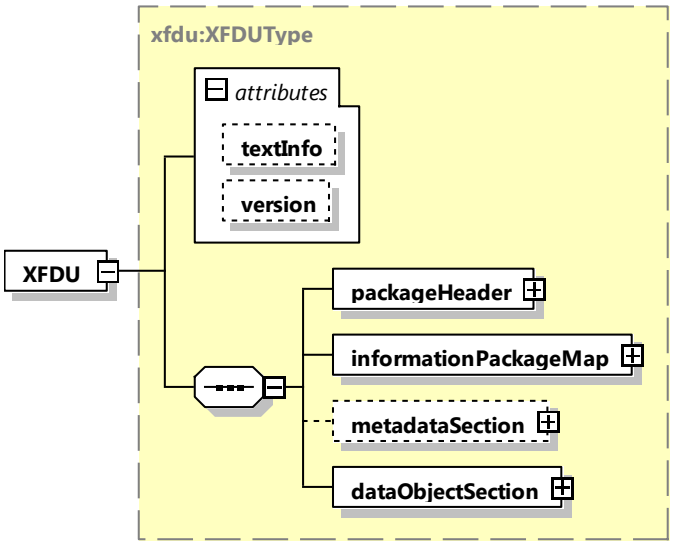

Figure F-4: Components of the Top-Level XFDU Structure Used by TGFT, in Graphical Representation

F6.1 XFDU MANIFEST ATTRIBUTES

The XFDU Manifest conforms to the XFDUType defined in reference [3]. XFDUType has four attributes:

1. **ID** (of type xsd:ID): This optional attribute is not defined in reference [3]. This attribute might be intended for use in cross referencing across different XFDU Manifests, which is outside the scope of TGFT. This attribute is not used in TGFT XFDUs.
2. **objID** (of type xsd:string): This optional attribute is defined in reference [3] as "a primary identifier assigned to this XFDU instance by the producer of the XFDU." This attribute is not used in TGFT XFDUs.
3. **textInfo** (of type xsd:string): This optional attribute is defined in reference [3] as "a title/text string identifying the document for users." For TGFT, whether the optional **textInfo** attribute is used and the value(s) that it carries are defined by the specification of the application that uses TGFT.
4. **version** (of type xsd:string): This optional attribute is defined in reference [3] as a "version of the XFDU XML Schema this XFDU should be validated against. Currently this is a string but when formal CCSDS XML Schema Naming and Versioning rules are defined it is expected that this type will be specialized to conform to those rules." Unfortunately, there is no indication in reference [3] what the current version is (presumably version 1, but is that coded as '1', '1.0', 'one', etc.?). Furthermore, as noted in the definition itself, there are not yet formal XML Schema Naming and Versioning rules, so the conditions that control the content of this attribute are currently unknown. For TGFT, the optional version attribute is absent to represent conformance to the version as specified in Issue 1 (September 2008) of reference [3]. If and when additional versions of the XML schema become available in the future, and assuming that by then schema naming and versioning rules will be in place to unambiguously define the corresponding contents of the version attribute, the version attribute will be required in TGFT XFDUs to specify which version is being used.

F6.2 PACKAGE HEADER

According to reference [3]:

> The ***Package Header*** (packageHeader element of packageHeaderType) is an XML Complex Type that contains metadata that apply to the whole XFDU Package. These metadata may include data to inform the XFDU parsing software about volume metadata (e.g., logical volume information and specification version), administrative metadata (e.g., author and creation data), and technical data (e.g., hardware and operating system).
>
> The package header type has two elements:
>
> – environmentInfo — contains application-specific information defined either by an extension of the XFDU Schema or by freeform XML;
>
> – volumeInfo — contains XFDU volume-related metadata such as XFDU specification version and logical volume sequence information.

Figure F-5 is a graphical representation of the CCSDS-standard InformationPackageMapType schema type.

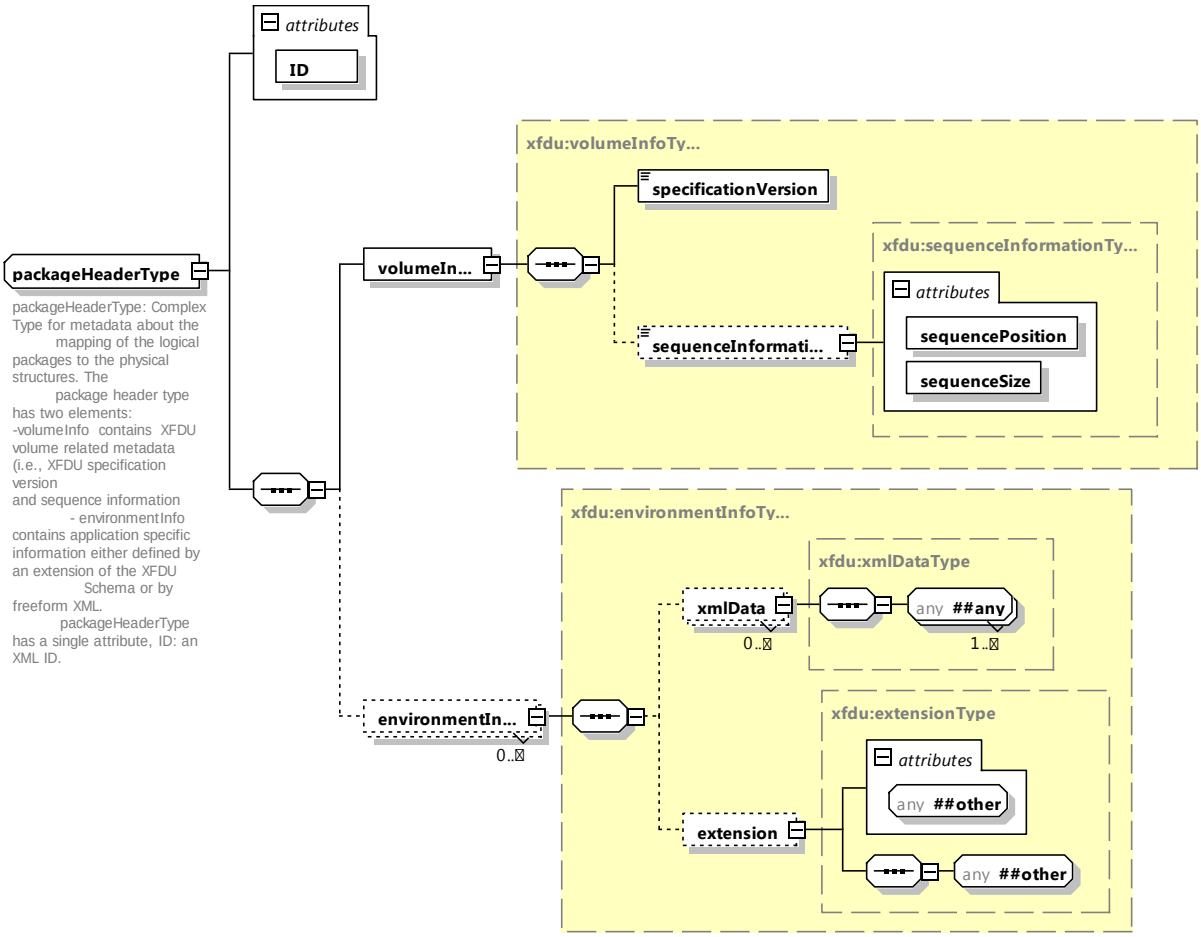

Figure F-5: CCSDS-Standard PackageHeaderType, in Graphical Representation

The packageHeader element is optional for CCSDS-standard XFDUs in general, according to reference [3]. However, for TGFT the packageHeader element is required.

The packageHeader has one attribute, **ID** (of type xsd:ID): This is a required attribute so it must be present in the packageHeader, but its usefulness is minimal since there is only one Package Header in the XFDU Manifest. The application using TGFT may specify semantics for this attribute, but the default value is 'PkgHdr'.

F6.2.1 volumeInfo element

The volumeInfo element has two component elements: specificationVersion and sequenceInformation.

F6.2.1.1 specificationVersion element

The required specificationVersion element "specifies the version of the XFDU specification to which this manifest complies." It is defined as of type xsd:string in reference [3]. Technically, CCSDS 'specifications' (e.g., Recommended Standards) are updated by 'Issue', not 'Version', and presumably this refers to the specific Issue of the *XML Formatted Data Unit (XFDU) Structure and Construction Rules* (reference [3]). As of this writing, the current Issue of reference [3] is '1'.

NOTE – The example XML snippet in reference [3] that includes the specificationVersion element has a value of '1.0'. However, full Recommended Standards do not carry decimal places in the Issue numbers. Presumably, the existence of the decimal place is there to accommodate with intermediate draft versions of Recommended Standards.

The value of the specificationVersion element shall be '1.0' for XFDUs used in conjunction with TGFT as specified in this Issue of the TGFT specification, indicating compliance with the Issue-1 CCSDS XFDU specification (reference [3]).

NOTE – If and when a new CCSDS XFDU specification is published, and if that new version provides additional capabilities pertinent to TGFT, a new set of TGFT-specific construction rules may be generated, in which case, those new construction rules would contain a specificationVersion value that points to the new CCSDS XFDU specification.

F6.2.1.2 sequenceInformation element

The optional **sequenceInformation** element is described in reference [3] as holding "information about sequence of XDFUs and the position of the current one within it." It contains two attributes: **sequenceSize** and **sequencePosition**, both of type xsd:nonNegativeInteger.

For TGFT, whether or not to use the optional **sequenceInformation** element is deferred to the TGFT-using application. The semantics of sequence information (e.g., whether the first XFDU in the sequence has position 0 or 1) are also deferred to the TGFT-using application.

F6.2.2 environmentInfo element

The optional **environmentInfo** element has two component elements: **xmlData** and **extension**.

The optional **xmlData** element contains "application specific information … defined … by freeform XML," whereas the optional **extension** element contains "application specific information … defined … by an extension of the XFDU Schema" (reference [3]).

For TGFT, at least one **environmentInfo** element is required, and that required element must contain an **extension** element that adds TGFT-specific metadata that applies to the XFDU as a whole (e.g., the originator of the XFDU). That required **environmentInfo** element is prohibited from containing any **xmlData** elements. The optional **xmlData** elements that are present in the **environmentInfoType** schema type are available for use in TGFT XFDUs only to carry TGFT-using-application-specific metadata, and only as elements of instances of the **environmentInfo** element other than the instance that carries the TFGT-specific metadata, as described below.

The mechanism by which the TGFT-specific metadata is added is as a **tgftXfduExtension** element of the **TgftXfduExtensionType** complex schema type, which is registered in the 'urn:ccsds:schema:tgft:xfdu_extensions' namespace in the file TgftXfduExtensionParameters.xsd.

If a TGFT-using application has application-specific metadata that applies to the XFDU Package as a whole, the recommended method for including such metadata is by the inclusion of zero or more **environmentInfo** elements, each containing an extension of the XFDU schema in the **extension** component element, should be used to contain the application-specific metadata. However, A TGFT-using application may also include application-specific metadata in zero or more **environmentInfo** elements each of which contains one or more **xmlData** elements containing freeform XML. The specification of application-specific **environmentInfo** content, whether in the form of schema types for the **extension** element, freeform XML for the **xmlData** element, or some combination of both, is deferred to the specifications of those applications.

Figure F-6 shows the components of the PackageHeaderType schema type that are used by TGFT. It should be noted that the **xmlData** elements are present only for the possible use by TGFT-using applications, and are not part of the standard TGFT XFDU.

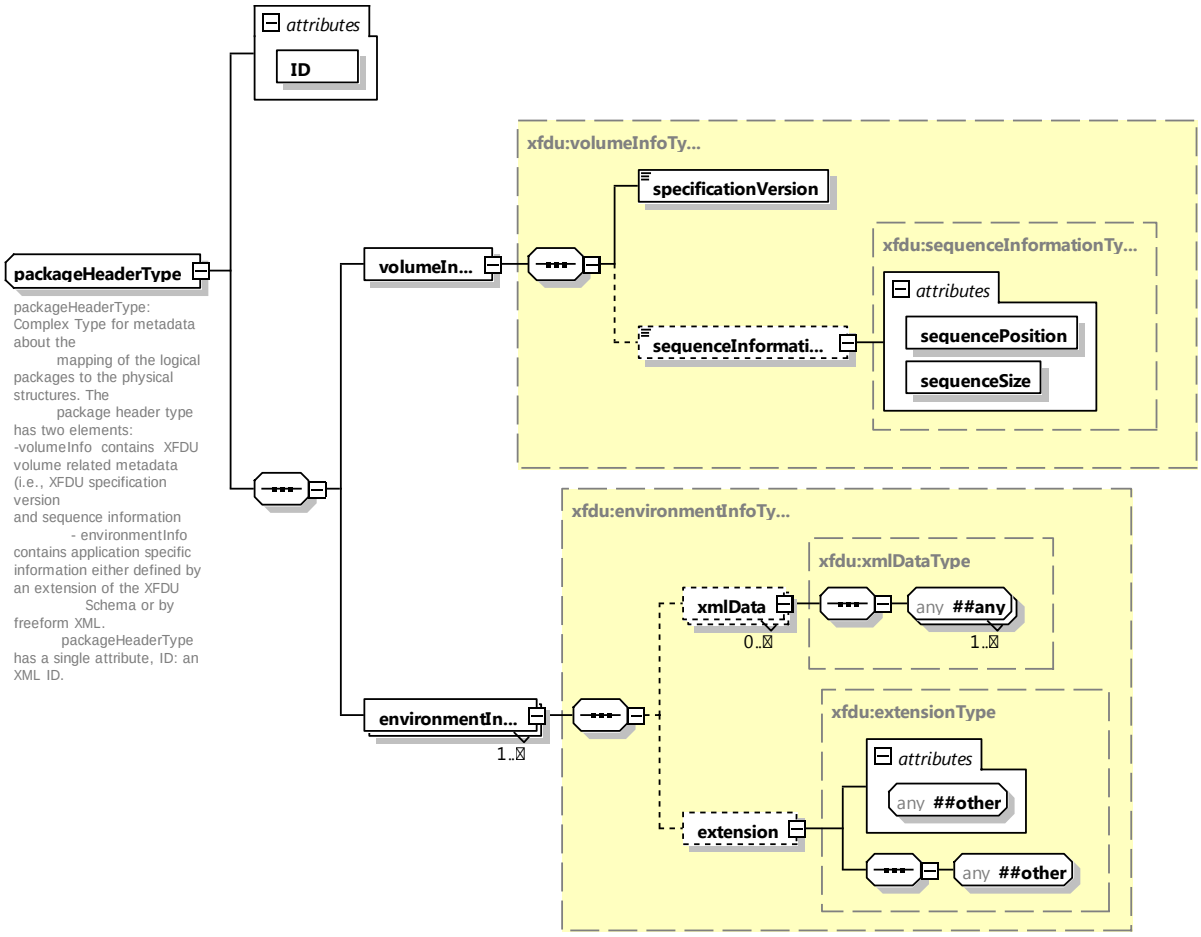

Figure F-6: PackageHeaderType Components Used by TGFT, in Graphical Representation

F6.3 INFORMATION PACKAGE MAP

According to 4.1 (b) of reference [3], the mandatory **"Information Package Map** (informationPackageMap element of informationPackageMapType): provides a hierarchical view of the content of the XFDU using a series of nested **contentUnit** elements. Content Units contain pointers to data objects and to the metadata associated with those data objects." That is, the Information Package Map is the 'table of contents' for XFDU Package.

Figure F-7 is a graphical representation of the CCSDS-standard InformationPackageMapType schema type. Figure F-8 shows the components of that schema type used by TGFT.

F6.3.1 informationPackageMap attributes.

The **informationPackageMap** element contains four attributes:

1. **ID** (of type xsd:ID): This optional attribute allows an ID to be assigned to the informationPackageMap. Because each XFDU Manifest contains only one informationPackageMap, this ID provides no additional information within the XFDU Manifest, and its purpose is not specified in reference [3]. Therefore this ID is not used in TGFT XFDUs.
2. **packageType** (of type xsd:string): According to reference [3], "a type for the object." The 'typical values' and only examples provided in the documentation for this optional attribute in reference [3] are ones that are pertinent to data archives that are defined in the context of the OAIS Reference Model (reference [6]). For TGFT, the **packageType** attribute is required, and is used to identify the application to which the XFDU Package belongs. These applications can be either CCSDS-standard services or locally defined applications. For CCSDS-standard services, the values that are used in packageType are registered in the packageTypeSANA registry (see C3.2) and indicate services that uses TGFT (e.g., 'ForwardFileService' for the Forward File service). For locally defined applications, the allowed values for the packageType element are specified in the appropriate service-agreement-level documentation.
3. **textInfo** (of type xsd:string): According to reference [3], "a string to describe the informationPackageMap to users." Because there is a one-to-one relationship between the XFDU Manifest as a whole and the informationPackageMap element, whatever textual description is needed can be included in the textInfo attribute of the XFDU Manifest itself. Therefore this textInfo attribute is not used in TGFT XFDUs.
4. **anyAttribute**: According to reference [3], a "wild-carded attribute extension point." This attribute is not used in TGFT XFDUs.

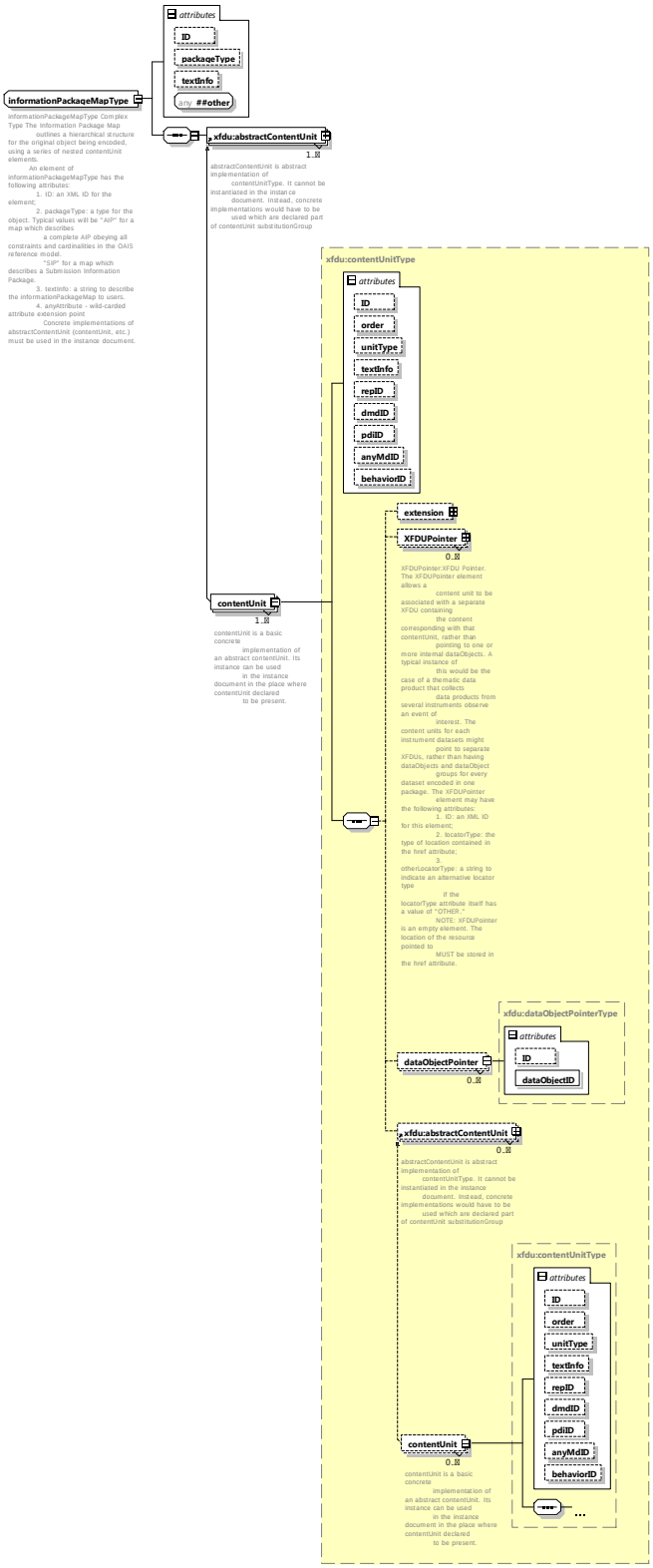

Figure F-7: CCSDS-Standard InformationPackageMapType, in Graphical Representation

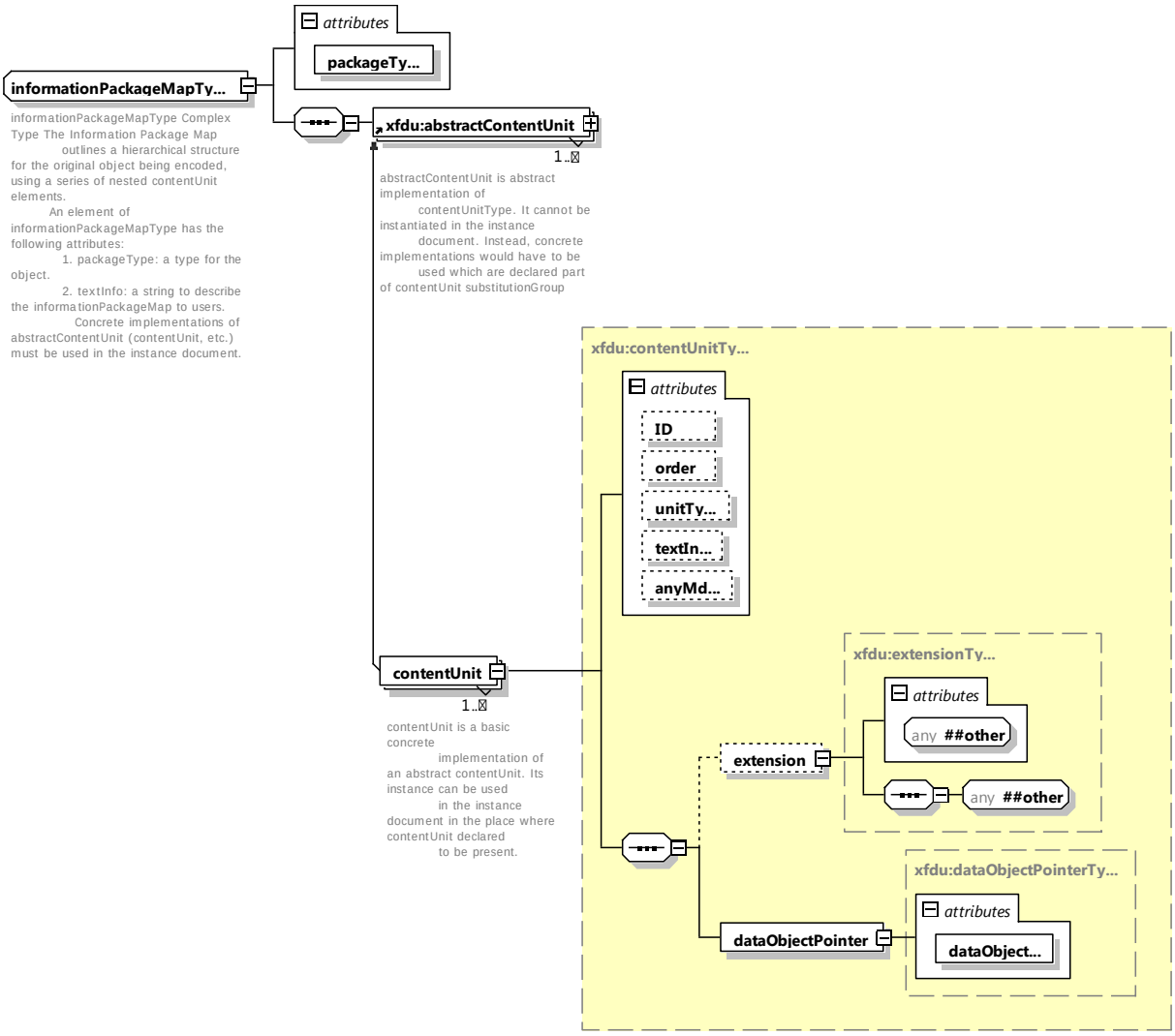

Figure F-8: InformationPackageMapType Components Used by TGFT, in Graphical Representation

F6.3.2 Content Units. In general, the CCSDS-standard XFDU Manifest Information Package Map contains one of more Content Units. As defined in 6.1 of reference [3]:

> A ***Content Unit*** (contentUnit element of contentUnitType) is the basic structural unit of the XFDU. Content Unit elements may include other Content Units, may be internal pointers to elements in the Data Object section, or may be external pointers to other XFDUs. Therefore a Content Unit can be used to associate a Data Object with one or more Metadata Objects, and multiple Content Units can present a hierarchical view of these data/metadata associations.

The CCSDS-standard XFDU Manifest allows recurrent containment of Content Units and containment of metadata in Data Objects in order to support arbitrarily complex relationships

among various aggregations of payload data and metadata. However, for TGFT, the composition of the Information Package Map and Content Unit are significantly simplified:

- The Information Package Map contains one or more Content Units, each of which contains one Data Object Pointer representing a single payload file Data Object.
- There is no recursive containment of Content Units.
- Data Objects (which Content Units point to) are not used to contain or refer to metadata.

F6.3.2.1 Content Unit attributes. The CCSDS-standard Content Unit has nine attributes:

a) **ID** (of type xsd:ID). This optional attribute is required in the Content Units of the XFDU Manifest for any XFDU Package that contains more than one payload data file, but may be included in an XFDU Manifest for an XFDU Package that contains a single payload data file.

b) **order** (of type xsd:string). This optional attribute is available to represent the order (sequence) among the Content Units in the Information Package Map. The default syntax for this attribute is the string representation of the sequence number of the content unit, starting with '1' and incrementing by single digits (e.g., '2', '3', etc.) such that the order attribute of the last content unit in the sequence has the string value equivalent of the number of payload data files in the XFDU Package.

c) **unitTyp**e (of type xsd:string). As defined in 6.2 of reference [3], 'a type of content unit (e.g., Application Data Unit, Data Description Unit, Software Installation Unit, etc.)'. There do not appear to be any pre-defined values for the content of this optional attribute. Therefore, if used, the content of this attribute must be defined by the TGFT-using application.

d) **textInfo** (of type xsd:string). As defined in 6.2 of reference [3], 'a string label to describe this contentUnit to an end user viewing the document, as per a table of contents entry'. If used, the content of this optional attribute must be defined by the TGFT-using application.

e) **repID** (of type xsd:IDREFS). This attribute is intended to contain the IDREF values of any OAIS Representation Information **metadataObject** elements in the Metadata Section (see F6.5) component of the XFDU Manifest. TGFT XFDUs do not explicitly recognize nor distinguish OAIS Representation Information metadataObject elements. Therefore this attribute is not used in TGFT XFDUs.

f) **dmdID** (of type xsd:IDREFS). This attribute is intended to contain the IDREF values of any OAIS Descriptive Information metadataObject elements in the Metadata Section (see F6.5) component of the XFDU Manifest. TGFT XFDUs do not explicitly recognize or distinguish OAIS Descriptive Information metadataObject elements. Therefore this attribute is not used in TGFT XFDUs.

g) **pdiID** (of type xsd:IDREFS). This attribute is intended to contain the IDREF values of OAIS Preservation Information metadataObject elements in the Metadata Section (see F6.5) component of the XFDU Manifest. TGFT XFDUs do not explicitly recognize or distinguish OAIS Preservation Information metadataObject elements. Therefore this attribute is not used in TGFT XFDUs.

h) **anyMdID** (of type xsd:IDREFS). For TGFT XFDUs, this attribute contains the IDREF values of the corresponding metadataObject elements in the Metadata Section (see F6.5) component of the XFDU Manifest.

NOTE – The type xsd:IDREFS is defined as a list of IDREFs, formatted as a string of whitespace-separated IDREFs. Thus a single attribute of type xsd:IDREFS can point to any number of components, each of which has an ID.

i) **behaviorID** (of type xsd:IDREF). The CCSDS-standard XFDU Manifest allows a single behaviorObject to be associated with each Content Unit, and if such an associated behaviorObject exists, the behaviorID contains the ID of that behaviorObject. However, since the Behavior Section is not used in TGFT XFDUs (see F6.6), this attribute is not used in TGFT XFDUs.

F6.3.2.2 Data Object Pointer. The Data Object Pointer (dataObjectPointer element) points to a Data Object (that is, a payload data file) within the same XFDU Package.

The CCSDS-standard XFDU Manifest allows one Content Unit to point to zero or more Data Objects (zero in the case where the Data Objects reside in different XFDU Packages—see F6.3.2.3).

For TGFT, the Content Unit points to one and only one Data Object.

The dataObjectPointer element has two attributes:

a) **ID** (of type xsd:ID): an XML ID for the element. This optional attribute is not used for TGFT XFDUs.

b) **dataObjectID** (of type xsd:IDREF): the value of the **ID** attribute of the dataObject element being pointed to. This attribute is required.

The dataObjectPointer element itself is empty; its information content is carried in the dataObjectID attribute.

F6.3.2.3 XFDU Pointer. The CCSDS-standard XFDU Manifest allows one Content Unit to point to Data Objects that reside in XFDUs outside the XFDU Package that contains the XFDU Manifest. However, for TGFT XFDU, all Data Objects (payload data) must reside within the same XFDU Package as the XFDU Manifest, so the XFDU Pointer is not used for TGFT.

F6.3.2.4 Content Unit extension element. The CCSDS-standard XFDU Manifest allows for the addition of extensions of the XFDU Content Unit from a separately controlled

namespace. For TGFT, this element is used to include TGFT-related metadata that applies on a per-Content-Unit (that is, a per-payload-data-file) basis.

The mechanism by which the TGFT-specific metadata is added is as a **tgftContentUnitExtension** element of the **TgftContentUnitExtensionType** complex schema type, which is registered in the 'urn:ccsds:schema:tgft:xfdu_extensions' namespace in the file TgftXfduExtensionParameters.xsd.

In addition to the TGFT Content Unit-specific metadata parameters, **TgftContentUnitExtensionType** also contains the **serviceSpecficContentUnitExt** extension element, which is available for use in adding application-specific metadata parameters that are applicable at the Content Unit level. The schema types that define such application-specific Content Unit metadata must be defined in a separate namespace that is associated with the TGFT-using application itself. The specification of such application-specific Content Unit metadata schema types and the namespaces in which they are registered is deferred to the TGFT-using applications.

NOTE – The **serviceSpecficContentUnitExt** extension element is included in the **TgftContentUnitExtensionType** because the **contentUnit** element contains only one extension element, so the addition of service-specific extension parameters must be 'daisy-chained' onto the TGFT extension parameters. This is in contrast to the addition of application-specific extension parameters to the **packageHeader** element, which allows multiple extensions (via multiple **environmentInfo** elements—see F6.2.2).

F6.4 DATA OBJECT SECTION

F6.4.1 General

Payload data is contained in, or referenced by, the dataObjectSection component of the XFDU Manifest.

Figure F-9 is a graphical representation of the CCSDS-standard DataObjectSectionType schema type.

F6.4.2 CCSDS-standard XFDU Data Object Section

In the CCSDS-standard XFDU Manifest, the dataObjectSection of the XFDU Manifest can contain one or more dataObject elements, each of which can contain one or more byteStream elements. The byteStream elements are a mechanism for breaking a large(r) data entity (the Data Object) into small(er) physical files (the byteStreams). The current XFDU specification defines only one combination relationship among the byteStreams that constitute a dataObject: concatenation (combinationName attribute = 'concat'). Each dataObject element may also contain an (optional) checksum over the contents of the Data Object, and one or more (optional) transformObjects (described below).

The CCSDS XFDU specification (reference [3]) allows each byteStream to contain (a) (optionally) zero or more fileLocation elements (which point to payload data files that are not contained within the Manifest), (b) (optionally) one fileContent element, which contains either base64 binaryData or xsd:any xmlData, and (c) (optionally) a checksum over the contents of the byteStream.

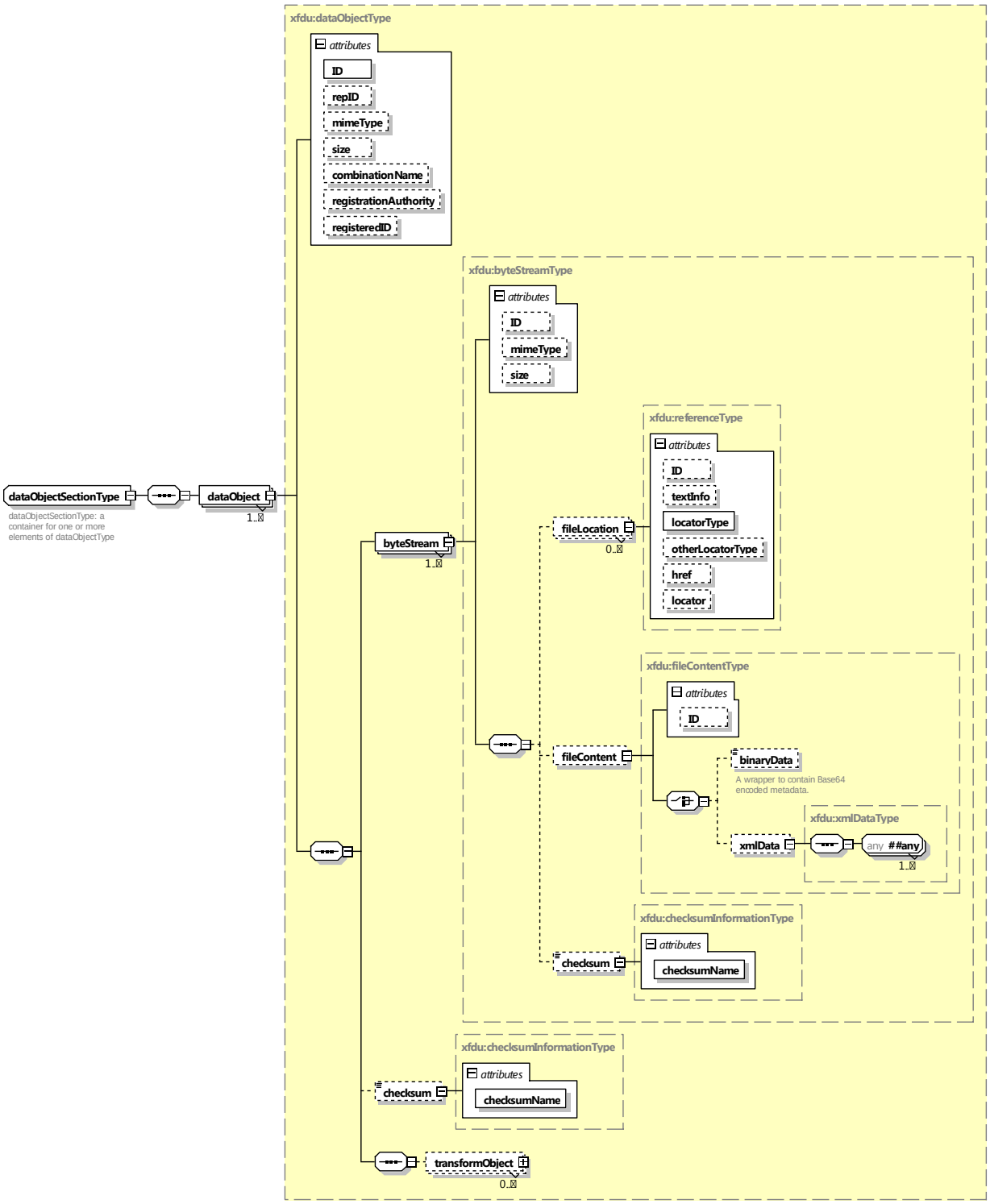

Figure F-9: CCSDS-Standard DataObjectSectionType, in Graphical Representation

Thus the CCSDS-standard XFDU specification allows multiple byteStream files to constitute a single Data Object, and the contents of those byteStreams can be (a) embedded within the XFDU Manifest itself, (b) external to the XFDU Manifest but contained within the same

XFDU Package as the XFDU Manifest, or (c) external to the XFDU Package. The flexibility to include (by reference) byteStream files that are external to the XFDU Package accommodates Data Objects that are too large to fit into a single XFDU Package, for example, a multi-year archive.

The semantics specified in 8.4 of reference [3] put constraints on the permitted combinations of fileContent and fileLocation elements:

a) A single fileLocation element may be contained by a byteStream to reference the payload data, either in the containing XFDU Package or some external source.

b) A (single) fileContent element may be contained by a byteStream to contain the payload data (in base64 binary or XML format) within the body of the XFDU Manifest itself.

c) A single fileLocation element **and** a (single) fileContent element may be contained by a byteStream, in which the fileContent data is to be a *backup* if the file referenced by fileLocation is not available; that is, fileContent and fileLocation contain/reference the same (redundant) information, but in different locations. The apparent intention here is that the data referenced by the fileLocation element is external to the XFDU Package (since having the referenced file contained within the same XFDU Package would be needlessly redundant).

d) Two fileLocation elements may be contained in a byteStream, in which one of the fileLocation elements references payload data *within the XFDU Package*, and the second fileLocation element references payload data that is *external to the XFDU Package*. In this combination, the referenced file within the XFDU Package is to be considered backup to the referenced file that is external to the XFDU Package; that is, the two sets of referenced data are the same.

e) Other forms of multiple fileLocation elements, and the presence of one fileContent element and more than one fileLocation element, are explicitly undefined, and their use is discouraged by the XFDU specification (reference [3]).

The CCSDS-standard XFDU Manifest also contains optional transformObject elements that "contain required information (e.g., algorithms and parameters) to reverse any transformations to the digital content and restore them to the original binary data object" (reference [3], subsection 8.1). The transformObject are used to (a) specify the relationship between a Data Object and the byteStreams into which it has been decomposed, and/or (b) to identify encryption and/or compression information associated with the Data Object.

F6.4.3 TGFT XFDU Data Object Section

For purposes of TGFT usage, the content of the Data Object Section is significantly simplified. Figure F-10 shows the components of the DataObjectSectionType schema type that are used by TGFT.

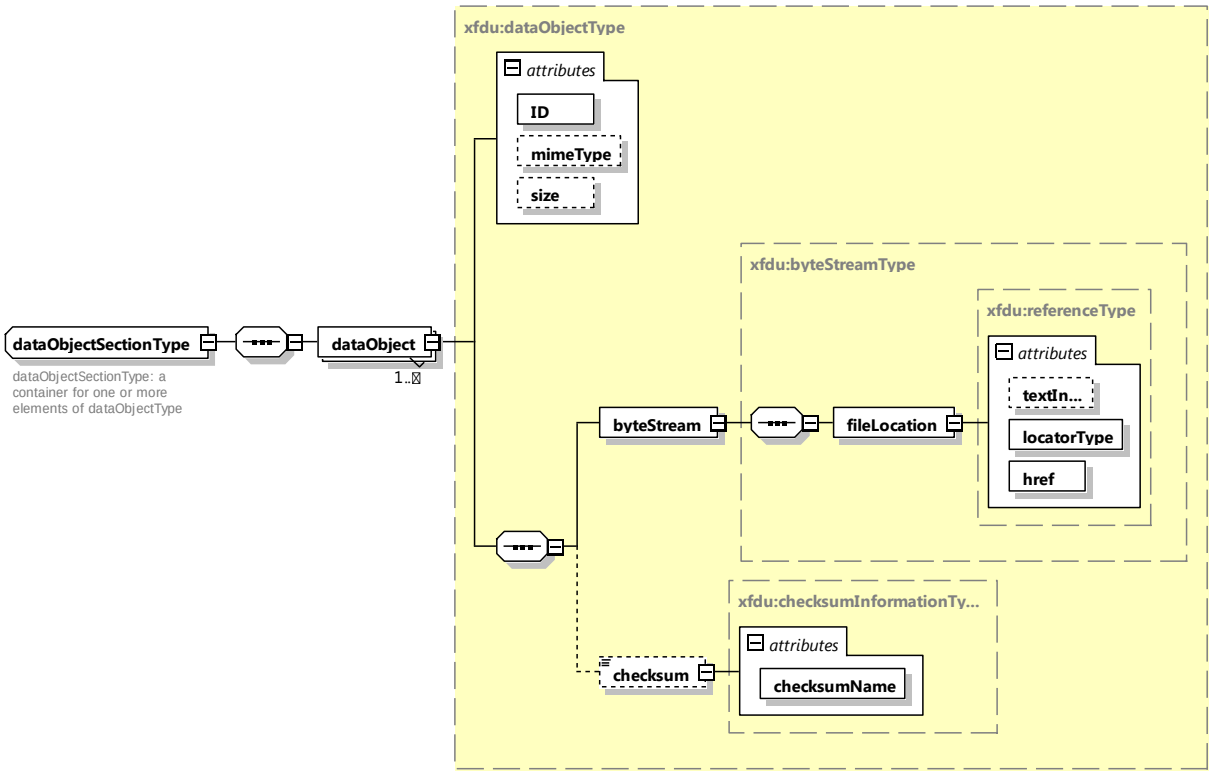

Figure F-10: DataObjectSectionType Components Used by TGFT, in Graphical Representation

F6.4.4 Data Object

F6.4.4.1 General

For TGFT, the only Data Objects that are recognized are those that can be conveyed in a single XFDU Package. Furthermore, for the purposes of TGFT, each TGFT XFDU Data Object Section contains a single Data Object.

NOTE – A TGFT-using application may still be able to use TGFT to transfer XFDU Packages that are segments of a larger XFDU, for example, ones in which a Data Object consists of a concatenation of multiple byteStream files, some of which are contained within the XFDU Packages themselves and others that are external to those XFDU Packages. However, any such usage must be defined as part of the specification of such a TGFT-using application.

Each Data Object contains seven attributes:

a) **ID** (of type xsd:ID): This mandatory attribute contains the ID of the Data Object. It is this ID value that is contained in the **dataObjectID** attribute of the dataObjectPointer element of the contentUnit element of the informationPackageMap.

b) **repID** (of type xsd:IDREFS): This optional attribute contains the "list of representation metadata IDREFs" applicable to the Data Object, and is present only if

there are applicable OAIS Representation Information metadata files. Since TGFT XFDUs neither recognize nor distinguish OAIS Representation Information metadata files, this attribute is not used in TGFT XFDUs.

c) **mimeType** (of type xfdu:mimeTypeType): This optional attribute contains the "MIME type for the dataObject." The presence and content of this attribute are controlled by the specification of the TGFT-using application.

d) **size** (of type xsd:long): This attribute contains the "size of the dataObject in bytes." This attribute is optional in the CCSDS-standard XFDU Manifest, but it is required for TGFT XFDUs.

e) **combinationName**: This optional attribute is not used in TGFT XFDUs because each Data Object consists of a single byteStream.

f) **registrationAuthority** (of type xsd:string): This optional attribute contains the name of "the authority that issued the registration." Whether or not this attribute is used, and how it is used, is deferred to the TGFT-using application.

g) **registeredID** (of type xsd:string): This optional attribute contains the name of "the ID for the registration." Whether or not this attribute is used, and how it is used, is deferred to the TGFT-using application.

For TGFT, the Data Object contains one byteStream element.

For TGFT, the Data Object may contain a checksum element.

For TGFT, the Data Object does not contain any transformObject elements. transformObject elements are unnecessary for TGFT because (a) there is a one-to-one relationship between Data Object and byteStream, so there are no transformations related to the decomposition of Data Objects into multiple byteStreams; and (b) any information related to encryption and/or compression algorithms are assumed to be defined as part of the specification of the TGFT-using application.

F6.4.4.2 byteStream Element

F6.4.4.2.1 General

The byteStream element has three attributes:

a) **ID** (of type xsd:ID): This optional attribute, if used, contains the ID of the byteStream. The byteStream **ID** attribute is not required for TGFT since there is a one-to-one relationship between the Data Object and the byteStream.

b) **mimeType** (of type xfdu:mimeTypeType): This optional attribute contains the MIME type for the byteStream. The byteStream **mimeType** attribute is not required for TGFT since there is a one-to-one relationship between the Data Object and the byteStream.

c) **size** (of type xsd:long): This attribute contains the "size of the dataObject in bytes." The byteStream **size** attribute is not used in TGFT XFDUs since there is a one-to-one relationship between the Data Object and the byteStream, and the size of the Data Object is already required to be included.

As noted above, reference [3] allows each byteStream to contain (a) (optionally) zero or more fileLocation elements (which point to payload data files that are not contained within the Manifest), (b) (optionally) one fileContent element, which contains either base64 binaryData or xsd:any xmlData, and (c) (optionally) a checksum over the contents of the byteStream.

For TGFT, the payload data file must be carried as an individual file outside of the XFDU Manifest but within the XFDU Package. Therefore the FileContent element is not used, and the fileLocation element (pointing to the payload file within the XFDU Package) must be used.

The checksum element of the byteStream is not used for TGFT because it is redundant with the dataObject checksum.

F6.4.4.2.2 fileLocation Element

The fileLocation element has six attributes:

1. **ID** (of type xsd:ID): This optional attribute is the identifier of the fileLocation element. This attribute is not used in TGFT XFDUs because TGFT constrains each byteStream to one fileLocation element, so a separate identifier is redundant.
2. **textInfo** (of type xsd:string): This optional attribute accommodates a human-readable description of the file location. This attribute is not required for TGFT; any use of this attribute will be as specified for the TGFT-using application.
3. **locatorType** (of type xsd:string, with enumerated values 'URL' and 'OTHER'): For TGFT, this attribute is required and the value is always 'URL'.
4. **otherLocatorType** (of type xsd:string): For TGFT, this optional attribute is not used because the value of locatorType in never 'OTHER'.
5. **href** (of type xsd:string): For TGFT, this attribute contains the URL of the file. The URL always begins with 'file:' because the payload file always exists within the same XFDU Package as the XFDU Manifest.
6. **locator** (of type xsd:string): For TGFT, this optional attribute is not used because the value of locatorType in never 'OTHER'.

The fileLocation element itself is empty; all information is carried in the attributes.

F6.4.4.2.3 Summary of TGFT XFDU simplified Data Object construction rules

The net effect of the constraints on the use of fileContent and fileLocation elements that are specified in reference [3], combined with the TGFT requirement that the payload data be

contained within the same XFDU Package, simplifies the allowed use of dataObject, fileContent, and fileLocation elements as follows:

- Each XFDU Package shall contain one or more Content Units.
- Each Content Unit shall contain a single dataObjectSection.
- Each dataObjectSection shall contain a single dataObject element.
- Each dataObject element shall contain a single byteStream element.
- Each byteStream element shall contain a single fileLocation element that points to a file within the same XFDU Package.

F6.5 METADATA SECTION

Metadata is contained in, or referenced by, the (optional) metadataSection component of the XFDU Manifest.

Figure F-11 is the graphical representation of the CCSDS-standard MetadataSectionType schema type. Figure F-12 shows the components of that schema type that are used by TGFT.

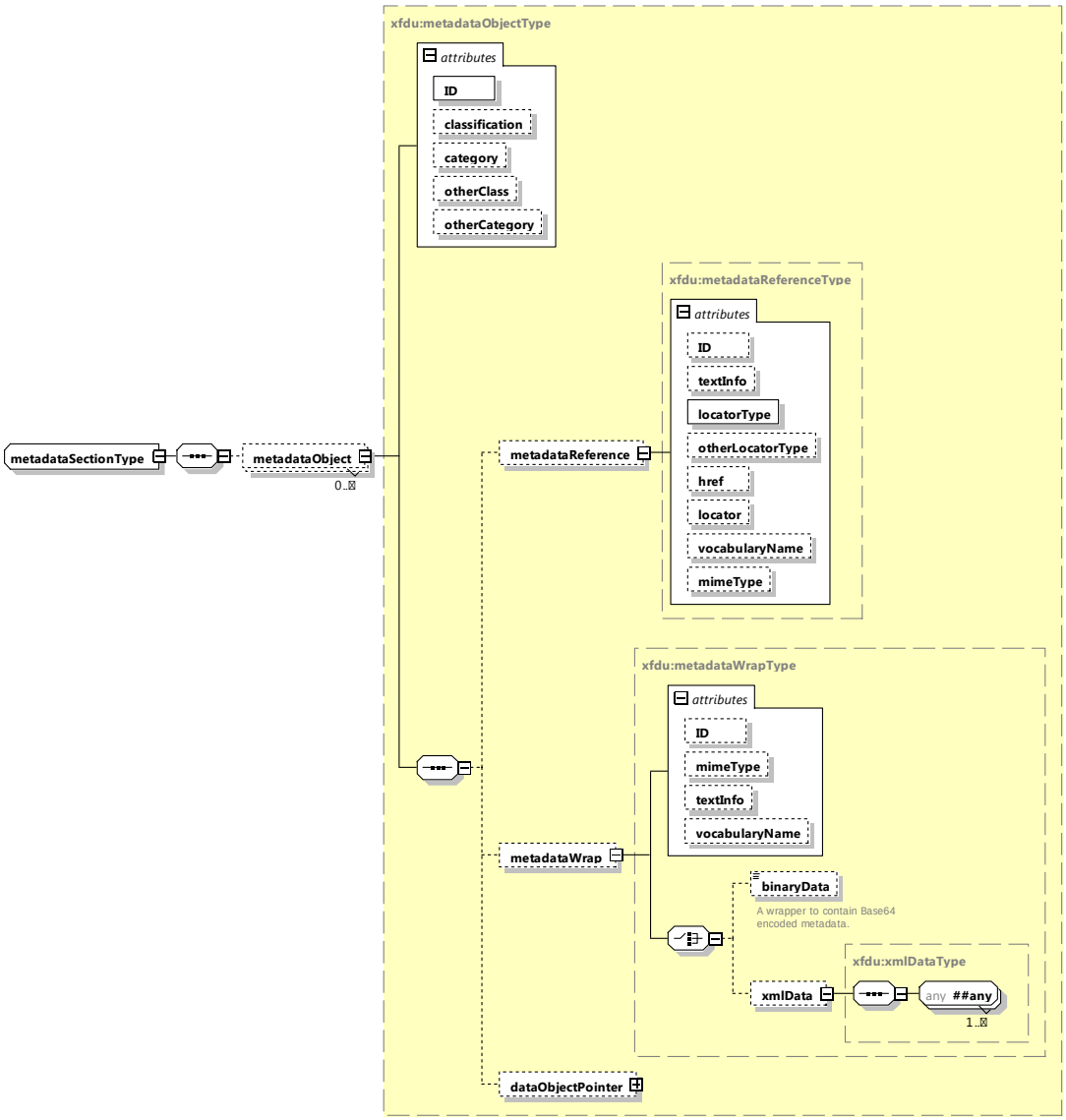

Figure F-11: CCSDS-Standard MetadataSectionType, in Graphical Representation

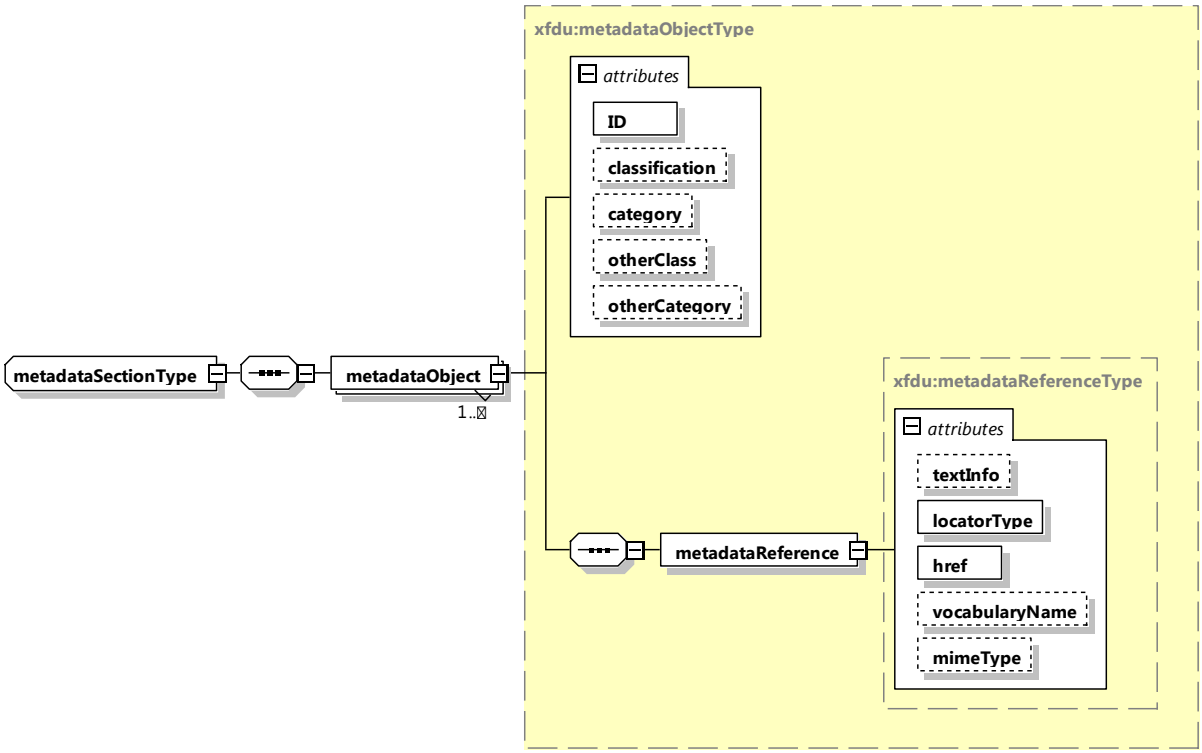

Figure F-12: MetadataSectionType Components Used by TGFT, in Graphical Representation

The XFDU specification (reference [3]) allows the metadataSection to contain zero or more metadataObject elements. However, for use in TGFT, if the optional metadataSection component is present, it is required to contain at least one metadataObject element. That is, if there are no metadataObjects, their absence in a TGFT XFDU Package is always represented by the absence of the metadataSection component.

According to reference [3], each metadataObject has the following attributes:

a) **ID** (of type xsd:ID): This required attribute contains the ID that is pointed to by the **anyMdID** attributes of the contentUnit elements associated with payload files that use those metadataObjects (see F6.3.2.1).

b) **classification** (of type xsd:string with enumerated values): This optional attribute is used to indicate the class of the metadata object. It has eight enumerated values (‘DED’, ‘SYNTAX’, ‘FIXITY’, ‘PROVENANCE’, ‘CONTEXT’, ‘REFERENCE’, ‘REFERENCE’, ‘DESCRIPTION’, and ‘OTHER’), of which seven (‘DED’, ‘SYNTAX’, ‘FIXITY’, ‘PROVENANCE’, ‘CONTEXT’, ‘REFERENCE, ‘REFERENCE’, and ‘DESCRIPTION’) have specific meaning with respect to pre-defined OAIS metadata categories. For TGFT, whether or not a class is assigned to a metadata file, and the semantics of such classification, is deferred to the individual TGFT-using applications. If classification is employed by a TGFT-using application, the value of the **classification** attribute must be ‘OTHER’ to indicate that the actual (application-defined) classification value is to be found in the **otherClass** attribute.

c) **category** (of type xsd:string with enumerated values): This optional attribute is used to indicate the category of the metadata object. It has five enumerated values ('REP', 'PDI', 'DMD', 'OTHER', and 'ANY'), of which four ('REP', 'PDI', 'DMD', and 'ANY') have specific meaning with respect to pre-defined OAIS metadata categories. For TGFT, whether or not a category is assigned to a metadata file, and the semantics of such categorization, is deferred to the individual TGFT-using applications. If categorization is employed by a TGFT-using application, the value of the **category** attribute must be 'OTHER' to indicate that the actual (application-defined) category value is to be found in the **otherCategory** attribute.

d) **otherClass** (of type xsd:string): This optional attribute if and only if the **classification** attribute is present and contains the value 'OTHER'. This attribute contains the application-specific classification value.

e) **otherCategory** (of type xsd:string): This optional attribute if and only if the **category** attribute is present and contains the value 'OTHER'. This attribute contains the application-specific category value.

The CCSDS-standard XFDU allows each metadataObject to contain (a) an optional metadataReference element, (b) an optional metadataWrap element, and (c) an optional dataObjectPointer element.

F6.5.1 metadataReference element. The metadataReference element is used to point to a metadata file that is outside of the XFDU Manifest itself. This could be either in the XFDU Package or external to the XFDU Package (e.g., on a server somewhere). Both options are supported by TGFT. Because the metadataReference element is the only element of the metadataObject, the metadataReference element must be present in each metadataObject.

The metadataReference element has eight attributes:

a) **ID** (of type xsd:ID): This optional attribute is not used for TGFT because there is a one-to-one relationship between the metadataReference and metadataObject. Therefore the required metadataObject **ID** attribute is also the ID for the metadataReference.

b) **textInfo** (of type xsd:string): This optional attribute is not required for TGFT, but it may be useful for some TGFT-using applications. If so, the semantics and allowed values must be defined as part of the application specification.

c) **locatorType** (of type xsd:string, with enumerated values 'URL' and 'OTHER'): For TGFT, this required attribute contains the value 'URL'.

d) **otherLocatorType** (of type xsd:string): optional – used when locatorType = 'OTHER'. This attribute is not used in TGFT XFDUs.

e) **href** (of type xsd:string): optional, used when locatorType = 'URL'. This attribute is required for TGFT XFDUs.

f) **locator** (of type xsd:string): optional, used when locatorType = 'OTHER'. This attribute is not used in TGFT XFDUs.

g) **vocabularyName** (of type xsd:string): According to reference [3], this optional attribute contains 'the type of metadata contained (e.g., MARC, EAD)'. This attribute is not required for TGFT, but it may be useful for some TGFT-using applications. If so, the semantics and allowed values must be defined as part of the application specification.

h) **mimeType** (of type xsd:string): This optional attribute identifies the MIME type of the metadata file. This attribute is not required for TGFT, but it may be useful for some TGFT-using applications. If so, the semantics and allowed values must be defined as part of the application specification.

The **metadataReference** element itself is empty; all information is contained in the attributes.

F6.5.2 metadataWrap element. The metadataWrap element is used to include metadata within the XFDU Manifest itself, in either base64 binary or XML format. This element is not used in TGFT XFDUs.

F6.5.3 dataObjectPointer element. The optional dataObjectPointer element points "to a Data Object in the Data Object section. This allows a Metadata Object to also be described as [a] Data Object in the Data Objects section. Since the dataObject includes an attribute that is an internal pointer to Representation Information, a Metadata Object can be associated with its own Representation Information. It should be noted that this mechanism allows the construction of OAIS-defined 'Representation Nets' when the associated Representation Metadata Objects are also held as Data Objects" (reference [3], subsection 9.1).

The dataObjectPointer type appears to exist to serve uses of XFDUs that are more esoteric than those present in simple file transfer. Therefore the dataObjectPointer element is not used in TGFT XFDUs.

F6.6 BEHAVIOR SECTION

The Behavior Section is not part of the standard TGFT XFDU Manifest.

F7 APPLICABILITY OF TGFT XFDU COMPOSITION RULES TO APPLICATIONS THAT USE TGFT

The composition rules for TGFT XFDUs define the subset of CCSDS-standard XFDU Package and XFDU Manifest elements and schema types that are standard for services that use the TGFT.

Nominally, TGFT-using applications are expected to support these composition rules in generating and processing the XFDU Packages that are transferred by those services. While

the subset of XFDU capabilities supported by the TGFT XFDU are intended to be able to support the use cases that have been identified so far (e.g., Forward File service, Delta-DOR File service, Validated Radiometric Data File service), there may arise future applications that could use TGFT as the underlying transfer mechanism but that have information content requirements that are not fully met by the standard TGFT XFDU but *are* supported by the other facets of the CCSDS-standard XFDU (e.g., Behavior data). Such applications may adopt a different set of restrictions on the XFDUs in order to support their requirements, but the redefinition of the TGFT XFDU in the context of those services must be fully defined in the specifications for those applications.

ANNEX G

ABBREVIATIONS AND ACRONYMS

(INFORMATIVE)

AD	Area Director
ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee on Space Data Systems
CFDP	CCSDS File Transfer Protocol
CSS	Cross Support Services
CSSS	Cross Support Service System
DAD	Deputy Area Director
DMZ	demilitarized zone
ICD	interface control document
ICS	implementation conformance statement
M	mandatory
N	no
n/a	not applicable
OAIS	Open Archival Information System
OMG	Object Management Group
O	optional
RL	requirements list
SANA	Space Assigned Numbers Authority
SCCS SM	Space Communication Cross Support Service Management
TDM	Tracking Data Message
TGFT	Terrestrial Generic File Transfer

UML	Unified Modelling Language
UTC	Coordinated Universal Time
VLBI	very-long-baseline interferometry
W3C	World Wide Web Consortium
XFDU	XML Formatted Data Unit
XML	Extensible Markup Language
Y	yes

ANNEX H

INFORMATIVE REFERENCES

[H1] T. Dierks and C. Allen. *The TLS Protocol*. Version 1.0. RFC . Reston, Virginia: ISOC, January 1999.

[H2] T. Dierks and E. Rescorla. *The Transport Layer Security (TLS) Protocol*. Version 1.1. RFC 4346. Reston, Virginia: ISOC, April 2006.

[H3] T. Dierks and E. Rescorla. *The Transport Layer Security (TLS) Protocol*. Version 1.2. RFC 5246. Reston, Virginia: ISOC, August 2008.[1]

[H4] R. Fielding, et al. *Hypertext Transfer Protocol -- HTTP/1.1*. RFC 2616. Reston, Virginia: ISOC, June 1999.

[H5] *Extensible Space Communication Cross Support—Service Management—Concept*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 902.0-G-1. Washington, D.C.: CCSDS, September 2014.

[H6] *Space Communications Cross Support—Architecture Description Document*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 901.0-G-1. Washington, D.C.: CCSDS, November 2013.

[H7] *Space Assigned Numbers Authority (SANA)—Role, Responsibilities, Policies, and Procedures*. Issue 2. CCSDS Record (Yellow Book), CCSDS 313.0-Y-2. Washington, D.C.: CCSDS, May 2016.

[H8] *IOAG Service Catalog #1*. Issue 2 revision 1. Washington, DC: IOAG, February 2017.

[H9] *CCSDS File Delivery Protocol (CFDP)—Part 2: Implementers Guide*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 720.2-G-3. Washington, D.C.: CCSDS, April 2007.

[H10] *CCSDS File Delivery Protocol (CFDP)*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 727.0-B-4. Washington, D.C.: CCSDS, January 2007.

[H11] *Space Communication Cross Support—Service Management—Service Specification*. Issue 1-S. Recommendation for Space Data System Standards (Historical), CCSDS 910.11-B-1-S. Washington, D.C.: CCSDS, (August 2009) June 2017.

[1] TLS 1.1 is an update from TLS version 1.0. TLS 1.2 is based on the earlier TLS 1.1 specification. All TLS versions were further refined in RFC 6176 in March 2011 removing their backward compatibility with SSL such that TLS sessions will never negotiate the use of Secure Sockets Layer (SSL) version 2.0.