INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR
STUDY PERIOD 1997 - 2000

COM 8-41-E
October 1997
Original: English

Question: 3/8

Texte disponible seulement en
Text available only in
Texto disponible solamente en
} **E**

STUDY GROUP 8 – CONTRIBUTION 41

SOURCE*: ASSOCIATED RAPPORTEUR

TITLE: BFT IMPLEMENTOR'S GUIDE

* **Contact:** Dr. Hans-Detlef Schulz, Germany
Tel: +49 6151 827711
Fax: +49 6151 827741
E-mail: hdschulz@t-online.de

IMPLEMENTOR'S GUIDE FOR T.434 (BFT)

1. INTRODUCTION

Recommendation T.434 has been standardized in 1992 (Version 1). Since this date, products consistent with this Recommendation have been developed and some problems have been found :

- errors or inaccuracies have been found in version 1,
- the ASN1 coding used for T.434 does not seem to be clear enough, since different BFT implementations have been unable to interwork.

In March 1996, a Version 2 of T.434 syntax has been approved, correcting the defects found in version 1 and adding some improvements such as the possibility to precisely indicate (object identifier instead of free syntax) the "application reference" or the "compression" type (T.4, T.6, JPEG...) used for the file transfer.

In order to facilitate the interworking between BFT implementations, this implementor's guide provides advice about the use of the parameters and indicates the hexadecimal coding which should be used. This implementor's guide takes into account the version 2 of T.434.

2. GENERAL PRINCIPLES

2.1 CODING OF THE LENGTH OF THE PARAMETERS

According to X.209 (§ 6.3) or X.690 (§ 8.1.3), two forms of lengths octets are specified :

- the definite form
- the indefinite form

and the following rules are defined :

"A sender shall
a) use the definite form if the encoding is primitive;
b) use either the definite form or the indefinite form, a sender's option, if the encoding is constructed and all immediately available;
c) use the indefinite form if the encoding is constructed and is not all immediately available".

For BFT all the parameters, except the file itself, are immediately available. So these parameters should be coded with the definite form, with the fewest length octets necessary. Due to the fact that, at the beginning of the coding, the sending terminal may ignore the length of the file, the only exception to this rule should be the parameter "data-file-content" the length of which shall be indefinite. To summarize, the following parameters could be coded with the indefinite form :

```
BINARY-DATA-Message ::= [Application 23] IMPLICIT SEQUENCE OF SEQUENCE
......... .......
data-file-content       [30] CHOICE ::=
                        {EXTERNAL,
                        ANY} OPTIONAL
```

See the coding examples in Tables 2 and 3

2.2 “CONSTRUCTED” OR "PRIMITIVE" FORM

X.209 or X.690 indicate whether a parameter must be coded as constructed or primitive. However, for some types of parameters, it is stated that the encoding will be "*either primitive or constructed at the option of the sender*". These types which are present in the definition of some parameters of T.434 are :

- Bitstring
- Octetstring
- GraphicString

For the simplification of the decoding of the parameters, the "Canonic encoding rules (CER)" defined in X.690 should apply :

"*Bitstring, octetstring, and restricted character string values shall be encoded with a primitive encoding if they would require no more than 1000 contents octets, and as a constructed encoding otherwise".*

Due to the fact that the parameters of the BFT header will likely contain less than 1000 octets, only the coding of the file (data-file-content) should be coded as "constructed" if the size of the file exceeds 1000 octets.

2.3 ORDER OF THE PARAMETERS

The BFT parameters are defined as a sequence :

```
BINARY-DATA-MESSAGE ::= [APPLICATION 23] IMPLICIT SEQUENCE OF SEQUENCE
```

According to X.209 (§ 14) or X.690 (§ 8.9), "*the contents octets shall consist of the complete encoding ....in the order of their appearance in the definition".*
That means that the parameters must be coded in the order defined in Annex A of T.434 and not in the numeric order of the tags assigned to the parameters. For example, the "protocol version" parameter (tag is [28]) which begins the sequence shall be the first parameter of the coded sequence and not the 28th.

2.4 RECOMMENDED PARAMETERS

All the T.434 parameters are optional except the "protocol version" parameter, which is defaultable and has a default value (version-1). In order to facilitate the processing of the transmitted file by the receiver, it is important that a minimum set of parameters be present. Taking into account the existing implementations of BFT, it is recommended that the following subset of the BFT parameters be implemented and interpreted :

- protocol-version
- contents-type
- application-reference
- filename
- date-and-time-of-creation
- date-and-time-of-last-modification
- identity-of-creator
- filesize

- operating-system
- compression (if applicable)
- user-visible-string (if applicable)
- private-use (if applicable)

However, if some other attributes are used by an implementation or some attributes listed above are missing, the receiving implementation of BFT should not react with an error, but try to operate in the best possible practise. If new parameters are added in the future, a previous version should ignore them when receiving these new parameters.

3. CLARIFICATION ON THE USE OF THE PARAMETERS

3.1 PARAMETERS DEFINED BY "SEQUENCE OF"

The syntax of T.434 allows to send several files during a BFT session, since the definition of a BFT data stream is :

BINARY-DATA-Message ::= [APPLICATION 23] IMPLICIT SEQUENCE OF SEQUENCE

The method consisting in sending one "sequence" for one file should be used rather than concatenating several files within one "data-file-content" parameter. With the first method, when several files are sent during the same session, each file is transmitted with one "sequence" and is preceded by its own header containing the different BFT parameters which applies to the file.
However, some parameters are defined with "SEQUENCE OF". That means that several occurrences of these parameters can be found for the transmission of a file. Some of these parameters are :

machine [20] IMPLICIT SEQUENCE OF GraphicString
environment [25] IMPLICIT SEQUENCE OF GraphicString
pathname [26] IMPLICIT SEQUENCE OF GraphicString

These parameters apply to only one file. So, if its agreed that a header applies to only one file, only one occurrence of these parameters should be found per file and per sequence.

3.2 "EXTERNAL” TYPE AND "ANY" TYPE FOR THE TRANSMISSION OF THE FILE

3.2.1 External Type

In the version 2 of T.434, the file is transmitted by means of the "data-file-content" parameter :

```
data-file-content [30] CHOICE ::=    {EXTERNAL,
                                      ANY} OPTIONAL
```

The "EXTERNAL" type is kept for compatibility with the version 1. When used, the coding should be consistent with X.208 § 34 or X.680 § 31 and X.690 § 8.18. However, this type is not well suitable for T.434 :

- this type usually applies to OSI environments where a context negotiation is performed.
- according to X.208 §34.5, "*when presentation layer negotiation of encoding rules is not in use* (it is the case of facsimile), *the "direct-reference OBJECT IDENTIFIER" shall be present".* But the value of this OBJECT IDENTIFIER is not defined in T.434.

Therefore the EXTERNAL type should not be used in the version 2 of T.434. Instead of "EXTERNAL", "ANY" should be used. However, if "EXTERNAL" is used, the value for the "direct-reference OBJECT IDENTIFIER" should be the one of the "Contents-Type-Attribute" or "Application-reference" attribute. As an example, if the "Contents-Type-Attribute" has the value "Unstructured-Binary" and if "EXTERNAL" is used, the "data-file content" parameter should be coded as follows :

```
    BE LL               data-file-content [30]
        28 LL           [Universal  8] Implicit sequence
             06         direct-reference OBJECT IDENTIFIER
                        05         Length
                        28 C2 7B 05 03     iso(0) standard (40) 8571(8571) document-type (5) unstructured-
binary (3)

             81 LL      octet-aligned [1] IMPLICIT OCTET STRING
                        xx  xx     content of the file
```

3.2.2 ANY Type

This type allows to choose any type among the types defined in X.208 or X.680. For the transmission of a file, it is likely that the "OCTET STRING" type be the best candidate to use. Therefore, in order to simplify the decoding of the BFT stream at the receiving side, the "OCTET STRING" type should be used.

3.3 RECOMMENDED USE OF PARAMETERS FOR FILE IDENTIFICATION

Within version 2 of T.434, it is possible to precisely specify the file type which is to be transferred. This can be accomplished by using the following two parameters of T.434:

1 - Contents-Type
2 - Application-Reference

The conventions which should be followed for precise identification of the file type are defined in the sections below.

3.3.1 “Contents-Type” Parameter

The syntax of the contents-type is defined as follows:

```
contents-type                         [2] Contents-Type-Attribute OPTIONAL,

        Contents-Type-Attribute             ::=CHOICE{
                document-type-name                  [1] OBJECT IDENTIFIER,
                parameter                           [0] ANY OPTIONAL}
```

Per the related notes within T.434 version 2, the default syntax is OBJECT IDENTIFIER. The following note is also included:

Note - The document-type-name of the Contents-Type-Attribute should be specified using the default value of UNSTRUCTURED BINARY in cases where the application-reference is being used for further identification of file contents. The object identifier for unstructured-binary is: { iso (0) standard (40) 8571 (8571) document-type (5) unstructured-binary (3) }

Therefore, for precise identification of file types, it is recommended that implementors set the value of the Contents-Type-Attribute to the default object identifier for “unstructured-binary” and then use the application-reference parameter to further identify the exact type of file.

3.3.2 “Application-Reference” Parameter

It is recommended that implementors use the Object ID syntax of the “Application-Reference” parameter to further identify the file type once the value for the contents-type has been set to use the object identifier for “unstructured-binary”.

There are Object Identifiers that have been defined by various organizations and companies. The defining entity generally has worked with the registration authorities in ISO, the ITU or national administrations to register a unique OID “arc”. From this point, the entities are able to use this “arc” as a starting point and then identify a file type by defining a unique OID sub-arc series of nodes.

An example of an organization OID arc is as follows:

{joint-iso-ccitt(2)country(16)US(840)organization(1)xyzcorp(5551212) }

An example of a OID sub-arc for a particular file type is as follows:

{files(4) wordprocessor(2)}

For this example, the full OID is:

WordProcessorStyleSheet ::=

{joint-iso-ccitt(2) country(16) US(840) Organization (1) xyzcorp(5551212) files(4) wordprocessor(2)}

An optional template that may be used by organizations that wish to develop Object IDs for specific file types is included in Appendix A of this guide. A list of currently available Object Identifiers that may be used with the Application-Reference parameter is also included in Appendix A .

When it is known, it is recommended that the exact Object Identifier for the Application Reference be selected from the table A.1 in Appendix A.

When the exact Object Identifier is not known for a binary file type or the file type is unknown, it is recommended that the following OID for an “unknown” file type be used:

```
{ joint-iso-ccitt(2) country(16) us(840) organization(1) ema(113694) objects
(2) messaging (2) attachments (1) unknown (1) }
```

When the exact Object Identifier is not known for a text file type or where no character set or code page is available for a text file, it is recommended that the following OID for an “unknown-text” file type be used:

{ joint-iso-ccitt(2) country(16) us(840) organization(1) ema(113694) objects (2) messaging (2) attachments (1) unknown-text (3) }

3.4 "PERMITTED-ACTIONS" ATTRIBUTE

It is not recommended to use this attribute for BFT.

3.5 "DATE AND TIME" ATTRIBUTES

With regard to "Date and Time", three parameters are defined in T.434 :

date-and-time-of-creation
date-and-time-of-last-modification
date-and-time-of-last-read-access

The ASN1 definition of these parameters is : "Generalized Time". However, there is a contradiction between T.434 and X.208 :

- According to X.208 § 32.3 (or X.680 § 39), the representation of the date is a string representing the calendar date, as specified in ISO 2014, with a four-digit representation of the year, a two-digit representation of the month and a two-digit representation of the day ....
- In T.434, the example in section 5.5 codes the date "2 January 1982" with only two digits (82) for the year. This coding is not consistent with "Generalized Time" but with "Universal Time".

Therefore, the example in section 5.5 must be corrected as follows :

Generalized Time	Length	Contents
18	0C	313938323031303230373030

4. HEXADECIMAL CODING OF THE BFT PARAMETERS

The following table 1 provides the hexadecimal coding to be used for the BFT parameters. This table is based on the version 2 of T.434.

The tables 2, 3 and 4 give examples of coding for a file with the attributes which will be likely used in most of the cases. The first example uses recommended parameters. The second example use ndefinite lengths for sending the content of the file whilst the third example only uses definite lengths.

Coding for BFT parameters

Notes :

1 - the parameters in *italic*s are not coded, due to the presence of the keyword "IMPLICIT".

Coding	Parameter	Parameter definition	Sub-parameter definition
77 LL (LL = length) **30 LL**	BINARY-DATA-Message	[APPLICATION 23] IMPLICIT *SEQUENCE OF* SEQUENCE {	
9C 02 **06 40**	protocol-version	[28] Protocol-Version DEFAULT {version-1},	Protocol-Version ::= IMPLICIT *BIT STRING* { version-2(1) }
A0 LL **19 LL** **xx ...yy**	filename	[0] IMPLICIT *Filename-Attribute* OPTIONAL,	Filename-Attribute ::= *SEQUENCE OF* GraphicString
81 LL **xx ...yy**	permitted-actions	[1] IMPLICIT *Permitted-Actions-Attribute* OPTIONAL,	Permitted-Actions-Attribute ::= *BIT STRING* { read (0), insert (1), replace (2), extend (3), erase (4) } (size (2...2))

A2 LL
A1 LL
06 LL
xx...yyy
or :
A2 LL
A0 LL
xx LL
xx...yyy
	contents-type	[2] Contents-Type-Attribute OPTIONAL,	Contents-Type-Attribute ::=
			CHOICE{
			document-type-name [1] OBJECT IDENTIFIER,
			parameter [0] ANY OPTIONAL}

83 LL **xx ...yy**	storage-account	[3] IMPLICIT *GraphicString* OPTIONAL,	
84 LL **xx .. yy**	date-and-time-of-creation	[4] IMPLICIT *GeneralizedTime* OPTIONAL,	
85 LL **xx .. yy**	date-and-time-of-last-modification	[5] IMPLICIT *GeneralizedTime* OPTIONAL,	
86 LL **xx .. yy**	date-and-time-of-last-read-access	[6] IMPLICIT *GeneralizedTime* OPTIONAL,	

88 LL **xx xx ...**	identity-of-creator	[8] IMPLICIT *GraphicString* OPTIONAL,	
89 LL **xx xx**	identity-of-last-modifier	[9] IMPLICIT *GraphicString* OPTIONAL,	
8A LL **xx xx ...**	identity-of-last-reader	[10] IMPLICIT *GraphicString* OPTIONAL,	
8D LL **xx xx**	filesize	[13] IMPLICIT *INTEGER* OPTIONAL,	
8E LL **xx xx**	future-filesize	[14] IMPLICIT *INTEGER* OPTIONAL,	
AF LL	access-control	[15] Access-Control-Attribute OPTIONAL,	for further study
90 LL xx xx	legal-qualifications	[16] IMPLICIT *GraphicString* OPTIONAL,	

B1 LL
30 LL
A0 LL
xx LL
xx x
	private-use	[17] Private-Use-Attribute OPTIONAL,	Private-Use-Attribute ::=
			SEQUENCE {
			manufacturer-values [0] ANY OPTIONAL }

92 LL **xx xx**	structure	[18] IMPLICIT *OBJECT IDENTIFIER* OPTIONAL,	

B3 LL
06 LL
xx xx
or
B3 LL
19 LL
xx xx
	application-reference	[19] General-Identifier OPTIONAL,	General-Identifier ::= CHOICE {
			OBJECT IDENTIFIER,
			IMPLICIT SEQUENCE OF GraphicString }

B4 LL **19 LL** **xx...yy**	machine	[20] IMPLICIT *SEQUENCE OF* GraphicString OPTIONAL,	
95 LL **xx xx**	operating-system	[21] IMPLICIT *OBJECT IDENTIFIER* OPTIONAL,	
B6 LL **19 LL** **xx...yy** **[19 LL** **xx xx]**	recipient	[22] IMPLICIT *SEQUENCE OF* GraphicString OPTIONAL,	
97 LL **xx xx**	character-set	[23] IMPLICIT *OBJECT IDENTIFIER* OPTIONAL,	

B8 LL
06 LL
xx xx
or
B8 LL
19 LL
xx xx
	compression	[24] General-Identifier OPTIONAL,	General-Identifier ::= CHOICE {
			OBJECT IDENTIFIER,
			IMPLICIT SEQUENCE OF GraphicString }

B9 LL **19 LL** **xx xx**	environment	[25] IMPLICIT *SEQUENCE OF* GraphicString OPTIONAL,	
BA LL **19 LL** **xx xx**	pathname	[26] IMPLICIT *SEQUENCE OF* GraphicString OPTIONAL,	
BD LL **19 LL** **xx xx**	user-visible-string	[29] IMPLICIT *SEQUENCE OF* GraphicString OPTIONAL,	

BE LL
xx xx
	data-file-content	[30] CHOICE ::= {
		EXTERNAL,
		ANY} OPTIONAL}	

Table 1

Example of Coding with recommended parameters

The following table gives an exemple of the BFT coding of a TIFF file named "PHOTO.TIF"

Coding	Value	Parameter	Parameter definition
77 80 **30 80**	(indefinite length) (indefinite length)	BINARY-DATA-Message	[APPLICATION 23] IMPLICIT *SEQUENCE OF* SEQUENCE {
9C 02 **06 40**	BFT version-2	protocol-version	[28] Protocol-Version {version-2},
A0 0B **19 09** **50 48 4F 54 4F 2E 54 49 46**	"PHOTO.TIF"	filename	[0] IMPLICIT *Filename-Attribute* OPTIONAL,

A2 09
A1 07
06 05
28 C2 7B 05 03
	Object identifier for unstructured binary : {iso(0) standard(40) 8571(8571) document-type(5) unstructured-binary(3)}	contents-type	[2] CHOICE{
			document-type-name [1] OBJECT IDENTIFIER,
			parameter [0] ANY OPTIONAL}

84 0C **31 39 39 36 30 36 30 38 31 31 30 35**	8 June 1996 - 11 H 05	date-and-time-of-creation	[4] IMPLICIT *GeneralizedTime* OPTIONAL,
85 0C **31 39 39 36 30 36 31 30 30 36 34 32**	10 June 1996 - 6 H 42	date-and-time-of-last-modification	[5] IMPLICIT *GeneralizedTime* OPTIONAL,
88 0A **4A 6F 68 6E 20 53 4D 49 54 48**	John SMITH	identity-of-creator	[8] IMPLICIT *GraphicString* OPTIONAL,
8D 84 **ww xx yy zz**	length of the file	filesize	[13] IMPLICIT *INTEGER* OPTIONAL,
B3 0D **06 0B** **60 86 78 01 86 F8 1E 02 02 01 05**	object identifier for a TIFF image: {joint-iso-ccitt(2) country(16) us(840) organization(1) ema(113694) ema objects(2) messaging(2) attachments(1) tiff-image(5)}	application-reference	[19] CHOICE { OBJECT IDENTIFIER, IMPLICIT SEQUENCE OF GraphicString},
	Object identifier to be defined	operating system	[21] IMPLICIT OBJECT IDENTIFIER OPTIONAL,

BE 80
24 80
04 82 xx yy
	(indefinite length)
	constructed form
		data-file-content	[30] CHOICE ::= {
			EXTERNAL,
			ANY} OPTIONAL}
			"*Octet string choosen*"

xx ..... xxxx **04 82 xx yy** **xx......xxxx**	First TIFF block of data Last TIFF block of data		
00 00 **00 00** **00 00** **00 00**	End of content (for indefinite length)		

Table 2

Example of Coding with indefinite lengths

The following table gives an example of the BFT coding of a file named "TEST.TXT", containing the 26 letters of the alphabet plus the 10 numbers and intended to be forwarded to two recipients.

Coding	Value	Parameter	Parameter definition
77 80 **30 80**	(indefinite length) (indefinite length)	BINARY-DATA-Message	[APPLICATION 23] IMPLICIT *SEQUENCE OF* SEQUENCE {
9C 02 **06 40**	BFT version-2	protocol-version	[28] Protocol-Version {version-2},
A0 0A **19 08** **54 45 53 54 2E 54 58 54**	"TEST.TXT"	filename	[0] IMPLICIT *Filename-Attribute* OPTIONAL,

A2 09
A1 07
06 05
28 C2 7B 05 03
	Object identifier for unstructured binary : {iso(0) standard(40) 8571(8571) document-type(5) unstructured-binary(3)}	contents-type	[2] CHOICE{
			document-type-name [1] OBJECT IDENTIFIER,
			parameter [0] ANY OPTIONAL}

84 0C **31 39 39 36 30 36 30 38 31 31 30 35**	8 June 1996 - 11 H 05	date-and-time-of-creation	[4] IMPLICIT *GeneralizedTime* OPTIONAL,
85 0C **31 39 39 36 30 36 31 30 30 36 34 32**	10 June 1996 - 6 H 42	date-and-time-of-last-modification	[5] IMPLICIT *GeneralizedTime* OPTIONAL,
88 0A **4A 6F 68 6E 20 53 4D 49 54 48**	John SMITH	identity-of-creator	[8] IMPLICIT *GraphicString* OPTIONAL,
89 0C **41 6C 61 69 6E 20 44 55 50 4F 4E 54**	Alain DUPONT	identity-of-last-modifier	[9] IMPLICIT *GraphicString* OPTIONAL,
8D 01 **24**	36 bytes	filesize	[13] IMPLICIT *INTEGER* OPTIONAL,
B3 0D **06 0B** **60 86 78 01 86 F8 1E 02 02 01 01**	object identifier for an unknown attachment : {joint-iso-ccitt(2) country(16) us(840) organization(1) ema(113694) ema objects(2) messaging(2) attachments(1) unknown(1)}	application-reference	[19] CHOICE { OBJECT IDENTIFIER, IMPLICIT SEQUENCE OF GraphicString},

	Object identifier to be defined	operating system	[21] IMPLICIT OBJECT IDENTIFIER OPTIONAL,
B6 21 **19 0C** **50 65 74 65 72 20 4D 41 52 54 49 4E** **19 11** **53 74 65 70 68 65 6E 20 4D 41 43 44 4F 4E 41 4C 44**	Peter MARTIN Stephen MACDONALD	recipient	[22] IMPLICIT *SEQUENCE OF* GraphicString OPTIONAL,

BE 80
24 80
04 1A
41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A
04 0A
30 31 32 33 34 35 36 37 38 39
	(indefinite length)
	constructed form
	"ABCDEFGHIJKL MNOPQRSTUVWXY Z"
	"0123456789"
		data-file-content	[30] CHOICE ::= {
			EXTERNAL,
			ANY} OPTIONAL}
			"*Octet string choosen*"

00 00 **00 00** **00 00** **00 00**	End of content (for indefinite lenth)		

Table 3

Example of Coding with definite lengths

The following table gives an exemple of the BFT coding of a file named "TEST.TXT", contaning the 26 letters of the alphabet and intended to be forwarded to two recipients.

Coding	Value	Parameter	Parameter definition
77 81 76 **30 81 73**	(definite length = 144 bytes) (definite length = 142 bytes)	BINARY-DATA-Message	[APPLICATION 23] IMPLICIT *SEQUENCE OF* SEQUENCE {
9C 02 **06 40**	BFT version-2	protocol-version	[28] Protocol-Version {version-2},
A0 0A **19 08** **54 45 53 54 2E 54 58 54**	"TEST.TXT"	filename	[0] IMPLICIT *Filename-Attribute* OPTIONAL,

A2 09
A1 07
06 05
28 C2 7B 05 03
	Object identifier for unstructured binary : {iso(0) standard(40) 8571(8571) document-type(5) unstructured-binary(3)}	contents-type	[2] CHOICE{
			document-type-name [1] OBJECT IDENTIFIER,
			parameter [0] ANY OPTIONAL}

84 0C **31 39 39 36 30 36 30 38 31 31 30 35**	8 June 1996 - 11 H 05	date-and-time-of-creation	[4] IMPLICIT *GeneralizedTime* OPTIONAL,
85 0C **31 39 39 36 30 36 31 30 30 36 34 32**	10 June 1996 - 6 H 42	date-and-time-of-last-modification	[5] IMPLICIT *GeneralizedTime* OPTIONAL,
88 0A **4A 6F 68 6E 20 53 4D 49 54 48**	John SMITH	identity-of-creator	[8] IMPLICIT *GraphicString* OPTIONAL,
8D 01 **1A**	26 bytes	filesize	[13] IMPLICIT *INTEGER* OPTIONAL,
B3 0D **06 0B** **60 86 78 01 86 F8 1E 02 02 01 01**	object identifier for an unknown attachment : {joint-iso-ccitt(2) country(16) us(840) organization(1) ema(113694) ema objects(2) messaging(2) attachments(1) unknown(1)}	application-reference	[19] CHOICE { OBJECT IDENTIFIER, IMPLICIT SEQUENCE OF GraphicString},
	Object identifier to be defined	operating system	[21] IMPLICIT OBJECT IDENTIFIER OPTIONAL,

BE1C
04 1A
41 42 43 44 45 46 47 48 49 4A
4B 4C 4D 4E 4F 50 51 52 53
54 55 56 57 58 59 5A
	(definite length)
	"ABCDEFGHIJKLMNOPQRSTUVWXYZ"
		data-file-content	[30] CHOICE ::= {
			EXTERNAL,
			ANY} OPTIONAL}
			"*Octet string choosen*"

Table 4