

International Telecommunication Union

**ITU-T**

TELECOMMUNICATION  
STANDARDIZATION SECTOR  
OF ITU

**X.691**

**Amendment 2**

(06/2006)

SERIES X: DATA NETWORKS, OPEN SYSTEM  
COMMUNICATIONS AND SECURITY

OSI networking and system aspects – Abstract Syntax  
Notation One (ASN.1)

---

Information technology – ASN.1 encoding rules:  
Specification of Packed Encoding Rules (PER)

**Amendment 2: Time type support**

ITU-T Recommendation X.691 (2002) – Amendment 2



ITU-T X-SERIES RECOMMENDATIONS  
DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

<b>PUBLIC DATA NETWORKS</b>	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
<b>OPEN SYSTEMS INTERCONNECTION</b>	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
<b>INTERWORKING BETWEEN NETWORKS</b>	
General	X.300–X.349
Satellite data transmission systems	X.350–X.369
IP-based networks	X.370–X.379
<b>MESSAGE HANDLING SYSTEMS</b>	X.400–X.499
<b>DIRECTORY</b>	X.500–X.599
<b>OSI NETWORKING AND SYSTEM ASPECTS</b>	
Networking	X.600–X.629
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
<b>Abstract Syntax Notation One (ASN.1)</b>	<b>X.680–X.699</b>
<b>OSI MANAGEMENT</b>	
Systems Management framework and architecture	X.700–X.709
Management Communication Service and Protocol	X.710–X.719
Structure of Management Information	X.720–X.729
Management functions and ODMA functions	X.730–X.799
<b>SECURITY</b>	X.800–X.849
<b>OSI APPLICATIONS</b>	
Commitment, Concurrency and Recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.889
Generic applications of ASN.1	X.890–X.899
<b>OPEN DISTRIBUTED PROCESSING</b>	X.900–X.999
<b>TELECOMMUNICATION SECURITY</b>	X.1000–

*For further details, please refer to the list of ITU-T Recommendations.*

**Information technology – ASN.1 encoding rules:  
Specification of Packed Encoding Rules (PER)**

**Amendment 2**

**Time type support**

**Summary**

This amendment specifies the PER encodings for the **TIME** type, for the useful time types (**DATE**, **TIME-OF-DAY**, **DATE-TIME**, and **DURATION**) and for the defined time types specified in the ASN.1 module **DefinedTimeTypes** of ITU-T Rec. X.680/Amd.3 | ISO/IEC 8824-1/Amd.3, Annex A *bis*.

Property settings for the abstract values of the useful and defined types are specified in ITU-T Rec. X.680/Amd.3 | ISO/IEC 8824-1/Amd.3, Annex A *bis*. Property settings for the additional time types are defined by an application designer using the property setting subtype notation (which is PER visible). These property settings are used to determine optimal PER encodings. In particular, a **DATE** will normally encode into 15 bits, a **TIME-OF-DAY** into 17 bits, a **DATE-TIME** into 32 bits (four octets) and a **DURATION** will frequently encode into 16 bits or less (two octets).

**Source**

Amendment 2 to ITU-T Recommendation X.691 (2002) was approved on 13 June 2006 by ITU-T Study Group 17 (2005-2008) under the ITU-T Recommendation A.8 procedure. An identical text is also published as ISO/IEC 8825-2, Amendment 2.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

## NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 2006

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

## CONTENTS

	<i>Page</i>
1) Contents .....	1
2) New clause 9.3.11 <i>bis</i> .....	1
3) New clause 28 <i>bis</i> .....	2



**INTERNATIONAL STANDARD  
ITU-T RECOMMENDATION**

**Information technology – ASN.1 encoding rules:  
Specification of Packed Encoding Rules (PER)**

**Amendment 2**

**Time type support**

**1) Contents**

*Update the Contents as follows:*

- 28 *bis* Encoding the time type, the useful time types, the defined time types and the additional time types
- 28 *bis.1* General
- 28 *bis.2* Encoding subtypes with the "Basic=Date" property setting
- 28 *bis.3* Encoding subtypes with the "Basic=Time" property setting
- 28 *bis.4* Encoding subtypes with the "Basic=Date-Time" property setting
- 28 *bis.5* Encoding subtypes with the "Basic=Interval Interval-type=SE" property setting
- 28 *bis.6* Encoding subtypes with the "Basic=Interval Interval-type=D" property setting
- 28 *bis.7* Encoding subtypes with the "Basic=Interval Interval-type=SD" or "Basic=Interval Interval-type=DE" property setting
- 28 *bis.8* Encoding subtypes with the "Basic=Rec-Interval Interval-type=SE" property setting
- 28 *bis.9* Encoding subtypes with the "Basic=Rec-Interval Interval-type=D" property setting
- 28 *bis.10* Encoding subtypes with the "Basic=Rec-Interval Interval-type=SD" or "Basic=Rec-Interval Interval-type=DE" property setting
- 28 *bis.11* Encoding subtypes with mixed settings of the Basic property

**2) New clause 9.3.11 *bis***

*Insert clause 9.3.11 *bis* after clause 9.3.11:*

**9.3.11 *bis*** Property setting constraints on the time type (or on the useful and defined time types) which are not extensible after the application of ITU-T Rec. X.680 | ISO/IEC 8824-1, 48.3 to 48.5, are PER-visible. Property setting constraints which are extensible are not PER-visible.

**3) New clause 28 bis**

*Insert clause 28 bis before clause 29:*

**28 bis Encoding the time type, the useful time types, the defined time types and the additional time types**

**28 bis.1 General**

**28 bis.1.1** The encoding of the useful time types, the defined time types and the additional time types shall be determined by the property settings of the abstract values of these types. Property settings for the abstract values of the useful and defined time types are specified in ITU-T Rec. X.680 | ISO/IEC 8824-1, 34 bis.4 and Annex A bis, respectively. Property settings for the abstract values of additional time types are determined by the property settings of the parent type, restricted by any PER-visible constraints that apply (see 9.3.11 bis).

**28 bis.1.2** If all the abstract values of the type to be encoded have one of the property settings listed in a row of column 2 of Table 2, then that type shall be encoded as if the type with its constraints (if any) had been replaced by the type specified in the corresponding row of column 3 of Table 2. Otherwise, it shall be encoded as specified in 28 bis.11.

NOTE – If a time property (for example **Midnight**) is not listed in Table 2 for a particular row, there is no constraint on its setting.

**28 bis.1.3** For rows 24 to 32 to be applicable, all abstract values of the type are required to have the same value of **n** in **F<sub>n</sub>**.

**28 bis.1.4** The types specified in column 3 of Table 2 are defined (using the ASN.1 notation) in 28 bis.2 to 28 bis.10, and are assumed to be defined in an environment of **AUTOMATIC TAGS**.

NOTE 1 – The use of these type reference names in the specification of PER encodings does not make them available for use by an application designer in an ASN.1 specification, nor are they reserved words in such a specification. However, with the removal of **-ENCODING**, they correspond to the names of the useful time types or defined time types specified in ITU-T Rec. X.680 | ISO/IEC 8824-1, 34 bis.4 and Annex A bis.

NOTE 2 – All the useful and defined time types satisfy the conditions for one of the rows of Table 2, and hence have optimized encodings. Additional time types may satisfy the conditions for one of the rows, but are otherwise encoded as specified in 28 bis.11. The unconstrained **TIME** type is always encoded as specified in 28 bis.11.

**Table 2 – Encoding of a time subtype with all abstract values having specified property settings**

Row number	Property settings	ASN.1 type to be encoded
1	"Basic=Date Date=C Year=Basic" or "Basic=Date Date=C Year=Proleptic"	<b>CENTURY-ENCODING</b> (see 28 bis.2.1)
2	"Basic=Date Date=C Year=Negative" or "Basic=Date Date=C Year=Ln" (for any n)	<b>ANY-CENTURY-ENCODING</b> (see 28 bis.2.2)
3	"Basic=Date Date=Y Year=Basic" or "Basic=Date Date=Y Year=Proleptic"	<b>YEAR-ENCODING</b> (see 28 bis.2.3)
4	"Basic=Date Date=Y Year=Negative" or "Basic=Date Date=Y Year=Ln" (for any n)	<b>ANY-YEAR-ENCODING</b> (see 28 bis.2.4)
5	"Basic=Date Date=YM Year=Basic" or "Basic=Date Date=YM Year=Proleptic"	<b>YEAR-MONTH-ENCODING</b> (see 28 bis.2.5)
6	"Basic=Date Date=YM Year=Negative" or "Basic=Date Date=YM Year=Ln" (for any n)	<b>ANY-YEAR-MONTH-ENCODING</b> (see 28 bis.2.6)
7	"Basic=Date Date=YMD Year=Basic" or "Basic=Date Date=YMD Year=Proleptic"	<b>DATE-ENCODING</b> (see 28 bis.2.7)
8	"Basic=Date Date=YMD Year=Negative" or "Basic=Date Date=YMD Year=Ln" (for any n)	<b>ANY-DATE-ENCODING</b> (see 28 bis.2.8)

Table 2 – Encoding of a time subtype with all abstract values having specified property settings

Row number	Property settings	ASN.1 type to be encoded
9	"Basic=Date Date=YD Year=Basic" or "Basic=Date Date=YD Year=Proleptic"	<b>YEAR-DAY-ENCODING</b> (see 28 bis.2.9)
10	"Basic=Date Date=YD Year=Negative" or "Basic=Date Date=YD Year=Ln" (for any <i>n</i> )	<b>ANY-YEAR-DAY-ENCODING</b> (see 28 bis.2.10)
11	"Basic=Date Date=YW Year=Basic" or "Basic=Date Date=YW Year=Proleptic"	<b>YEAR-WEEK-ENCODING</b> (see 28 bis.2.11)
12	"Basic=Date Date=YW Year=Negative" or "Basic=Date Date=YW Year=Ln" (for any <i>n</i> )	<b>ANY-YEAR-WEEK-ENCODING</b> (see 28 bis.2.12)
13	"Basic=Date Date=YWD Year=Basic" or "Basic=Date Date=YWD Year=Proleptic"	<b>YEAR-WEEK-DAY-ENCODING</b> (see 28 bis.2.13)
14	"Basic=Date Date=YWD Year=Negative" or "Basic=Date Date=YWD Year=Ln" (for any <i>n</i> )	<b>ANY-YEAR-WEEK-DAY-ENCODING</b> (see 28 bis.2.14)
15	"Basic=Time Time=H Local-or-UTC=L"	<b>HOURS-ENCODING</b> (see 28 bis.3.1)
16	"Basic=Time Time=H Local-or-UTC=Z"	<b>HOURS-UTC-ENCODING</b> (see 28 bis.3.2)
17	"Basic=Time Time=H Local-or-UTC=LD"	<b>HOURS-AND-DIFF-ENCODING</b> (see 28 bis.3.3)
18	"Basic=Time Time=HM Local-or-UTC=L"	<b>MINUTES-ENCODING</b> (see 28 bis.3.4)
19	"Basic=Time Time=HM Local-or-UTC=Z"	<b>MINUTES-UTC-ENCODING</b> (see 28 bis.3.5)
20	"Basic=Time Time=HM Local-or-UTC=LD"	<b>MINUTES-AND-DIFF-ENCODING</b> (see 28 bis.3.6)
21	"Basic=Time Time=HMS Local-or-UTC=L"	<b>TIME-OF-DAY-ENCODING</b> (see 28 bis.3.7)
22	"Basic=Time Time=HMS Local-or-UTC=Z"	<b>TIME-OF-DAY-UTC-ENCODING</b> (see 28 bis.3.8)
23	"Basic=Time Time=HMS Local-or-UTC=LD"	<b>TIME-OF-DAY-AND-DIFF-ENCODING</b> (see 28 bis.3.9)
24	"Basic=Time Time=HF <sub>n</sub> Local-or-UTC=L" (but see 28 bis.1.3)	<b>HOURS-AND-FRACTION-ENCODING</b> (see 28 bis.3.10)
25	"Basic=Time Time=HF <sub>n</sub> Local-or-UTC=Z" (but see 28 bis.1.3)	<b>HOURS-UTC-AND-FRACTION-ENCODING</b> (see 28 bis.3.11)
26	"Basic=Time Time=HF <sub>n</sub> Local-or-UTC=LD" (but see 28 bis.1.3)	<b>HOURS-AND-DIFF-AND-FRACTION-ENCODING</b> (see 28 bis.3.12)
27	"Basic=Time Time=HMF <sub>n</sub> Local-or-UTC=L" (but see 28 bis.1.3)	<b>MINUTES-AND-FRACTION-ENCODING</b> (see 28 bis.3.13)
28	"Basic=Time Time=HMF <sub>n</sub> Local-or-UTC=Z" (but see 28 bis.1.3)	<b>MINUTES-UTC-AND-FRACTION-ENCODING</b> (see 28 bis.3.14)
29	"Basic=Time Time=HMF <sub>n</sub> Local-or-UTC=LD" (but see 28 bis.1.3)	<b>MINUTES-AND-DIFF-AND-FRACTION-ENCODING</b> (see 28 bis.3.15)
30	"Basic=Time Time=HMSF <sub>n</sub> Local-or-UTC=L" (but see 28 bis.1.3)	<b>TIME-OF-DAY-AND-FRACTION-ENCODING</b> (see 28 bis.3.16)

Table 2 – Encoding of a time subtype with all abstract values having specified property settings

Row number	Property settings	ASN.1 type to be encoded
31	"Basic=Time Time=HMSFn Local-or-UTC=Z" (but see 28 bis.1.3)	TIME-OF-DAY-UTC-AND-FRACTION-ENCODING (see 28 bis.3.17)
32	"Basic=Time Time=HMSFn Local-or-UTC=LD" (but see 28 bis.1.3)	TIME-OF-DAY-AND-DIFF-AND-FRACTION-ENCODING (see 28 bis.3.18)
33	"Basic=Date-Time" All abstract values are required to have the same additional property settings specified in one of rows 1 to 14 for "Basic=Date" together with the same additional property settings specified in one of the rows 15 to 32 for "Basic=Time".	DATE-TIME-ENCODING {Date-Type, Time-Type} (instantiated as specified in 28 bis.4.1)
34	"Basic=Interval Interval-type=SE SE-point=Date" All abstract values are required to have the same additional property settings specified in one of rows 1 to 14 for "Basic=Date".	START-END-DATE-INTERVAL-ENCODING {Date-Type} (see 28 bis.5.1)
35	"Basic=Interval Interval-type=SE SE-point=Time" All abstract values are required to have the same additional property settings specified in one of rows 15 to 32 for "Basic=Time".	START-END-TIME-INTERVAL-ENCODING {Time-Type} (see 28 bis.5.2)
36	"Basic=Interval Interval-type=SE SE-point=Date-Time" All abstract values are required to have the same additional property settings specified in one of rows 1 to 14 for "Basic=Date" together with the same additional property settings specified in one of rows 15 to 32 for "Basic=Time".	START-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} (see 28 bis.5.3)
37	"Basic=Interval Interval-type=D"	DURATION-INTERVAL-ENCODING (see 28 bis.6.1)
38	"Basic=Interval Interval-type=SD SE-point=Date" All abstract values are required to have the same additional property settings specified in one of rows 1 to 14 for "Basic=Date".	START-DATE-DURATION-INTERVAL-ENCODING {Date-Type} (see 28 bis.7.1)
39	"Basic=Interval Interval-type=SD SE-point=Time" All abstract values are required to have the same additional property settings specified in one of rows 15 to 32 for "Basic=Time".	START-TIME-DURATION-INTERVAL-ENCODING {Time-Type} (see 28 bis.7.2)
40	"Basic=Interval Interval-type=SD SE-point=Date-Time" All abstract values are required to have the same additional property settings specified in one of rows 1 to 14 for "Basic=Date" together with the same additional property settings specified in one of rows 15 to 32 for "Basic=Time".	START-DATE-TIME-DURATION-INTERVAL-ENCODING {Date-Type, Time-Type} (see 28 bis.7.3)
41	"Basic=Interval Interval-type=DE SE-point=Date" All abstract values are required to have the same additional properties specified in one of rows 1 to 14 for "Basic=Date".	DURATION-END-DATE-INTERVAL-ENCODING {Date-Type} (see 28 bis.7.4)

Table 2 – Encoding of a time subtype with all abstract values having specified property settings

Row number	Property settings	ASN.1 type to be encoded
42	"Basic=Interval Interval-type=DE SE-point=Time" All abstract values are required to have the same additional properties specified in one of rows 15 to 32 for "Basic=Time".	DURATION-END-TIME-INTERVAL-ENCODING {Time-Type} (see 28 bis.7.5)
43	"Basic=Interval Interval-type=DE SE-point=Date-Time" All abstract values are required to have the same additional properties specified in one of rows 1 to 14 for "Basic=Date" together with the same additional property settings specified in one of rows 15 to 32 for "Basic=Time".	DURATION-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} (see 28 bis.7.6)
44	"Basic=Rec-Interval Interval-type=SE SE-point=Date" All abstract values are required to have the same additional property settings specified in one of rows 1 to 14 for "Basic=Date".	REC-START-END-DATE-INTERVAL-ENCODING {Date-Type} (see 28 bis.8.1)
45	"Basic=Rec-Interval Interval-type=SE SE-point=Time" All abstract values are required to have the same additional property settings specified in one of rows 15 to 32 for "Basic=Time".	REC-START-END-TIME-INTERVAL-ENCODING {Time-Type} (see 28 bis.8.2)
46	"Basic=Rec-Interval Interval-type=SE SE-point=Date-Time" All abstract values are required to have the same additional property settings specified in one of rows 1 to 14 for "Basic=Date" together with the same additional property settings specified in one of rows 15 to 32 for "Basic=Time".	REC-START-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} (see 28 bis.8.3)
47	"Basic=Rec-Interval Interval-type=D"	REC-DURATION-INTERVAL-ENCODING (see 28 bis.9.1)
48	"Basic=Rec-Interval Interval-type=SD SE-point=Date" All abstract values are required to have the same additional property settings specified in one of rows 1 to 14 for "Basic=Date".	REC-START-DATE-DURATION-INTERVAL-ENCODING {Date-Type} (see 28 bis.10.1)
49	"Basic=Rec-Interval Interval-type=SD SE-point=Time" All abstract values are required to have the same additional property settings specified in one of rows 15 to 32 for "Basic=Time".	REC-START-TIME-DURATION-INTERVAL-ENCODING {Time-Type} (see 28 bis.10.2)
50	"Basic=Rec-Interval Interval-type=SD SE-point=Date-Time" All abstract values are required to have the same additional property settings specified in one of rows 1 to 14 for "Basic=Date" together with the same additional property settings specified in one of rows 15 to 32 for "Basic=Time".	REC-START-DATE-TIME-DURATION-INTERVAL-ENCODING {Date-Type, Time-Type} (see 28 bis.10.3)
51	"Basic=Rec-Interval Interval-type=DE SE-point=Date" All abstract values are required to have the same additional properties specified in one of rows 1 to 14 for "Basic=Date".	REC-DURATION-END-DATE-INTERVAL-ENCODING {Date-Type} (see 28 bis.10.4)

Table 2 – Encoding of a time subtype with all abstract values having specified property settings

Row number	Property settings	ASN.1 type to be encoded
52	"Basic=Rec-Interval Interval-type=DE SE-point=Time" All abstract values are required to have the same additional properties specified in one of rows 15 to 32 for "Basic=Time".	REC-DURATION-END-TIME-INTERVAL-ENCODING {Time-Type} (see 28 bis.10.5)
53	"Basic=Rec-Interval Interval-type=DE SE-point=Date-Time" All abstract values are required to have the same additional properties specified in one of rows 1 to 14 for "Basic=Date" together with the same additional property settings specified in one of rows 15 to 32 for "Basic=Time".	REC-DURATION-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} (see 28 bis.10.6)

### 28 bis.2 Encoding subtypes with the "Basic=Date" property setting

This subclause defines the ASN.1 types referenced in Table 2, column 3 for types where all the abstract values of the type have the "Basic=Date" property setting.

**28 bis.2.1** The CENTURY-ENCODING type is:

```
CENTURY-ENCODING ::= INTEGER(0..99) -- 7 bits
```

with the integer value set to the value specified by the first two digits of the year component of the abstract value.

**28 bis.2.2** The ANY-CENTURY-ENCODING type is:

```
ANY-CENTURY-ENCODING ::= INTEGER(MIN..MAX)
```

with the integer value set to the value specified by the year component of the abstract value, ignoring the last two digits.

**28 bis.2.3** The YEAR-ENCODING type is:

```
YEAR-ENCODING ::= CHOICE { -- 2 bits for choice determinant
    immediate    INTEGER (2005..2020), -- 4 bits
    near-future  INTEGER (2021..2276), -- 8 bits
    near-past    INTEGER (1749..2004), -- 8 bits
    remainder    INTEGER (MIN..1748 | 2277..MAX) }
```

with the integer value set to the year component of the abstract value.

NOTE – This has been optimized to provide a 6-bit or a 10-bit encoding in common cases.

**28 bis.2.4** The ANY-YEAR-ENCODING type is:

```
ANY-YEAR-ENCODING ::= INTEGER(MIN..MAX)
```

with the integer value set to the year component of the abstract value.

**28 bis.2.5** The YEAR-MONTH-ENCODING type is:

```
YEAR-MONTH-ENCODING ::= SEQUENCE {
    year        YEAR-ENCODING,
    month       INTEGER (1..12) -- 4 bits -- }
```

with the YEAR-ENCODING set according to 28 bis.2.3 and the month integer value set to the month component of the abstract value.

NOTE – This has been optimized to provide a 10-bit or a 14-bit encoding in common cases.

**28 bis.2.6** The ANY-YEAR-MONTH-ENCODING type is:

```
ANY-YEAR-MONTH-ENCODING ::= SEQUENCE {
    year        ANY-YEAR-ENCODING,
    month       INTEGER (1..12) }
```

with the ANY-YEAR-ENCODING set according to 28 bis.2.4 and the month integer value set to the month component of the abstract value.

**28 bis.2.7** The **DATE-ENCODING** type is:

```
DATE-ENCODING ::= SEQUENCE {
    year          YEAR-ENCODING,
    month         INTEGER (1..12), -- 4 bits
    day           INTEGER (1..31) -- 5 bits -- }
```

with the **YEAR-ENCODING** set according to 28 bis.2.3, the **month** integer value set to the month component of the abstract value and the **day** integer value set to the day component of the abstract value.

NOTE – This has been optimized to provide a 15-bit or a 19-bit encoding in common cases.

**28 bis.2.8** The **ANY-DATE-ENCODING** type is:

```
ANY-DATE-ENCODING ::= SEQUENCE {
    year          ANY-YEAR-ENCODING,
    month         INTEGER (1..12),
    day           INTEGER (1..31)}
```

with the **ANY-YEAR-ENCODING** set according to 28 bis.2.4, the **month** integer value set to the month component of the abstract value and the **day** integer value set to the day component of the abstract value.

**28 bis.2.9** The **YEAR-DAY-ENCODING** type is:

```
YEAR-DAY-ENCODING ::= SEQUENCE {
    year          YEAR-ENCODING,
    day           INTEGER (1..366)}
```

with the **YEAR-ENCODING** set according to 28 bis.2.3 and the **day** integer value set to the day component of the abstract value.

**28 bis.2.10** The **ANY-YEAR-DAY-ENCODING** type is:

```
ANY-YEAR-DAY-ENCODING ::= SEQUENCE {
    year          ANY-YEAR-ENCODING,
    day           INTEGER (1..366)}
```

with the **ANY-YEAR-ENCODING** set according to 28 bis.2.4 and the **day** integer value set to the day component of the abstract value.

**28 bis.2.11** The **YEAR-WEEK-ENCODING** type is:

```
YEAR-WEEK-ENCODING ::= SEQUENCE {
    year          YEAR-ENCODING,
    week          INTEGER (1..53) -- 6 bits --}
```

with the **YEAR-ENCODING** set according to 28 bis.2.3 and the **week** integer value set to the week component of the abstract value.

NOTE – This has been optimized to provide a 12-bit or a 16-bit encoding in common cases.

**28 bis.2.12** The **ANY-YEAR-WEEK-ENCODING** type is:

```
ANY-YEAR-WEEK-ENCODING ::= SEQUENCE {
    year          ANY-YEAR-ENCODING,
    week          INTEGER (1..53)}
```

with the **ANY-YEAR-ENCODING** set according to 28 bis.2.4 and the **week** integer value set to the week component of the abstract value.

**28 bis.2.13** The **YEAR-WEEK-DAY-ENCODING** type is:

```
YEAR-WEEK-DAY-ENCODING ::= SEQUENCE {
    year          YEAR-ENCODING,
    week          INTEGER (1..53), -- 6 bits
    day           INTEGER (1..7) -- 3 bits -- }
```

with the **YEAR-ENCODING** set according to 28 bis.2.3, the **week** integer value set to the week component of the abstract value and the **day** integer value set to the day component of the abstract value.

NOTE – This has been optimized to provide a 15-bit or a 19-bit encoding in common cases.

**28 bis.2.14** The **ANY-YEAR-WEEK-DAY-ENCODING** type is:

```
ANY-YEAR-WEEK-DAY-ENCODING ::= SEQUENCE {
    year          ANY-YEAR-ENCODING,
    week          INTEGER (1..53),
    day           INTEGER (1..7)}
```

with the **ANY-YEAR-ENCODING** set according to 28 bis.2.4, the **week** integer value set to the week component of the abstract value and the **day** integer value set to the day component of the abstract value.

### 28 bis.3 Encoding subtypes with the "Basic=Time" property setting

This subclause defines the ASN.1 types referenced in Table 2, column 3 for types where all the abstract values of the type have the **Basic=Time** property setting.

**28 bis.3.1** The **HOURS-ENCODING** type is:

```
HOURS-ENCODING ::= INTEGER(0..24) -- 5 bits
```

with the integer value set to the hours component of the abstract value.

NOTE – This has been optimized to provide a 5-bit encoding.

**28 bis.3.2** The **HOURS-UTC-ENCODING** type is:

```
HOURS-UTC-ENCODING ::= INTEGER(0..24) -- 5 bits
```

with the integer value set to the hours component of the abstract value.

NOTE – This has been optimized to provide a 5-bit encoding.

**28 bis.3.3** The **HOURS-AND-DIFF-ENCODING** type is:

```
HOURS-AND-DIFF-ENCODING ::= SEQUENCE {
    local-hours    INTEGER (0..24),
    time-difference TIME-DIFFERENCE }
```

where:

```
TIME-DIFFERENCE ::= SEQUENCE {
    hours          INTEGER (-15..16),
    minutes        INTEGER (1..59) OPTIONAL }
```

with the **local-hours** integer value set to the hours component of the local time of the abstract value and the **time-difference** set to the hours and the minutes of the time-difference component of the abstract value. If the minutes component of the time-difference is zero, the **TIME-DIFFERENCE minutes** shall be omitted.

**28 bis.3.4** The **MINUTES-ENCODING** type is:

```
MINUTES-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59) -- 5 bits -- }
```

with the **hours** integer value set to the hours component of the abstract value and the **minutes** integer value set to the minutes component.

NOTE – This has been optimized to provide a 10-bit encoding.

**28 bis.3.5** The **MINUTES-UTC-ENCODING** type is:

```
MINUTES-UTC-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59) -- 5 bits -- }
```

with the **hours** integer value set to the hours component of the abstract value and the **minutes** integer value set to the minutes component.

NOTE – This has been optimized to provide a 10-bit encoding.

**28 bis.3.6** The **MINUTES-AND-DIFF-ENCODING** type is:

```
MINUTES-AND-DIFF-ENCODING ::= SEQUENCE {
    local-time      SEQUENCE {
        hours       INTEGER (0..24),
        minutes     INTEGER (0..59) },
    time-difference TIME-DIFFERENCE }
```

with the **local-time** set to the hours and minutes component of the local time of the abstract value and the **time-difference** set to the hours and minutes of the time-difference component of the abstract value as specified in 28 bis.3.3.

**28 bis.3.7** The **TIME-OF-DAY-ENCODING** type is:

```
TIME-OF-DAY-ENCODING ::= SEQUENCE {
    hours           INTEGER (0..24), -- 5 bits
    minutes         INTEGER (0..59), -- 5 bits
    seconds         INTEGER (0..60) -- 5 bits -- }
```

with the **hours** integer value set to the hours component of the abstract value, the **minutes** integer value set to the minutes component, and the **seconds** integer value set to the seconds component.

NOTE – This has been optimized to provide a 15-bit encoding.

**28 bis.3.8** The **TIME-OF-DAY-UTC-ENCODING** type is:

```
TIME-OF-DAY-UTC-ENCODING ::= SEQUENCE {
    hours           INTEGER (0..24), -- 5 bits
    minutes         INTEGER (0..59), -- 5 bits
    seconds         INTEGER (0..60) -- 5 bits -- }
```

with the **hours** integer value set to the hours component of the abstract value, the **minutes** integer value set to the minutes component, and the **seconds** integer value set to the seconds component.

NOTE – This has been optimized to provide a 15-bit encoding.

**28 bis.3.9** The **TIME-OF-DAY-AND-DIFF-ENCODING** type is:

```
TIME-OF-DAY-AND-DIFF-ENCODING ::= SEQUENCE {
    local-time      SEQUENCE {
        hours       INTEGER (0..24),
        minutes     INTEGER (0..59),
        seconds     INTEGER (0..60) },
    time-difference TIME-DIFFERENCE }
```

with the **local-time** set to the hours, minutes and seconds components of the local time of the abstract value and the **time-difference** set to the hours and minutes of the time-difference component of the abstract value as specified in 28 bis.3.3.

**28 bis.3.10** The **HOURS-AND-FRACTION-ENCODING** type is:

```
HOURS-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours           INTEGER (0..24), -- 5 bits
    fraction         INTEGER (0..999, ..., 1000..MAX)
    -- 11 bits for up to three digits accuracy -- }
```

with the **hours** integer value set to the hours component of the abstract value and the **fraction** integer value set to the fractional hours multiplied by ten-to-the-power-N, where N is the specified number of digits in the fractional part.

NOTE – This has been optimized to provide a 16-bit encoding for up to 3-digit accuracy.

**28 bis.3.11** The **HOURS-UTC-AND-FRACTION-ENCODING** type is:

```
HOURS-UTC-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours           INTEGER (0..24), -- 5 bits
    fraction         INTEGER (0..999, ..., 1000..MAX)
    -- 11 bits for up to three digits accuracy -- }
```

with the **hours** integer value set to the hours component of the abstract value and the **fraction** integer value set to the fractional hours multiplied by ten-to-the-power-N, where N is the specified number of digits in the fractional part.

NOTE – This has been optimized to provide a 16-bit encoding for up to 3-digit accuracy.

**28 bis.3.12** The **HOURS-AND-DIFF-AND-FRACTION-ENCODING** type is:

```
HOURS-AND-DIFF-AND-FRACTION-ENCODING ::= SEQUENCE {
    local-hours      INTEGER (0..24), -- 5 bits
    fraction         INTEGER (0..999, ..., 1000..MAX)
                    -- 11 bits for up to three digits accuracy -- ,
    time-difference TIME-DIFFERENCE }
```

with the **local-hours** integer value set to the hours component of the local time of the abstract value, the **fraction** integer value set to the fractional hours multiplied by ten-to-the-power-N (where N is the specified number of digits in the fractional part) and the **time-difference** set to the hours and the minutes of the time-difference component of the abstract value as specified in 28 bis.3.3.

**28 bis.3.13** The **MINUTES-AND-FRACTION-ENCODING** type is:

```
MINUTES-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours           INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59), -- 5 bits
    fraction       INTEGER (0..999, ..., 1000..MAX)
                    -- 11 bits for up to three digits accuracy -- }
```

with the **hours** integer value set to the hours component of the abstract value, the **minutes** integer value set to the minutes component and the **fraction** integer value set to the fractional hours multiplied by ten-to-the-power-N, where N is the specified number of digits in the fractional part.

NOTE – This has been optimized to provide a 21-bit encoding for up to 3-digit accuracy.

**28 bis.3.14** The **MINUTES-UTC-AND-FRACTION-ENCODING** type is:

```
MINUTES-UTC-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours           INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59), -- 5 bits
    fraction       INTEGER (0..999, ..., 1000..MAX)
                    -- 11 bits for up to three digits accuracy -- }
```

with the **hours** integer value set to the hours component of the abstract value, the **minutes** integer value set to the minutes component and the **fraction** integer value set to the fractional hours multiplied by ten-to-the-power-N (where N is the specified number of digits in the fractional part).

NOTE – This has been optimized to provide a 21-bit encoding for up to 3-digit accuracy.

**28 bis.3.15** The **MINUTES-AND-DIFF-AND-FRACTION-ENCODING** type is:

```
MINUTES-AND-DIFF-AND-FRACTION-ENCODING ::= SEQUENCE {
    local-time      SEQUENCE {
        hours       INTEGER (0..24),
        minutes    INTEGER (0..59),
        fraction    INTEGER (0..999, ..., 1000..MAX)},
    time-difference TIME-DIFFERENCE }
```

with the **local-time** set to the hours and minutes component of the local time of the abstract value, the **fraction** integer value set to the fractional minutes multiplied by ten-to-the-power-N (where N is the specified number of digits in the fractional part) and the **time-difference** set to the hours and minutes of the time-difference component of the abstract value as specified in 28 bis.3.3.

**28 bis.3.16** The **TIME-OF-DAY-AND-FRACTION-ENCODING** type is:

```
TIME-OF-DAY-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours           INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59), -- 5 bits
    seconds       INTEGER (0..60), -- 5 bits --
    fraction       INTEGER (0..999, ..., 1000..MAX)
                    -- 11 bits for up to three digits accuracy -- }
```

with the **hours** integer value set to the hours component of the abstract value, the **minutes** integer value set to the minutes component, the **seconds** integer value set to the seconds component and **fraction** integer value set to the fractional seconds multiplied by ten-to-the-power-N, where N is the specified number of digits in the fractional part.

NOTE – This has been optimized to provide a 26-bit encoding.

**28 bis.3.17** The **TIME-OF-DAY-UTC-AND-FRACTION-ENCODING** type is:

```

TIME-OF-DAY-UTC-AND-FRACTION-ENCODING ::= SEQUENCE {
    hours          INTEGER (0..24), -- 5 bits
    minutes        INTEGER (0..59), -- 5 bits
    seconds        INTEGER (0..60), -- 5 bits --
    fraction       INTEGER (0..999, ..., 1000..MAX)
    -- 11 bits for up to three digits accuracy -- }

```

with the **hours** integer value set to the hours component of the abstract value, the **minutes** integer value set to the minutes component, the **seconds** integer value set to the seconds component and **fraction** integer value set to the fractional seconds multiplied by ten-to-the-power-N, where N is the specified number of digits in the fractional part.

NOTE – This has been optimized to provide a 26-bit encoding.

**28 bis.3.18** The **TIME-OF-DAY-AND-DIFF-AND-FRACTION-ENCODING** type is:

```

TIME-OF-DAY-AND-DIFF-AND-FRACTION-ENCODING ::= SEQUENCE {
    local-time     SEQUENCE {
        hours      INTEGER (0..24),
        minutes    INTEGER (0..59),
        seconds    INTEGER (0..60),
        fraction   INTEGER (0..999, ..., 1000..MAX)},
    time-difference TIME-DIFFERENCE }

```

with the **local-time** set to the hours, minutes, seconds and fractional part components of the local time of the abstract value and the **time-difference** set to the hours and minutes of the time-difference component of the abstract value as specified in 28 bis.3.3.

#### **28 bis.4 Encoding subtypes with the "Basic=Date-Time" property setting**

This subclause defines the ASN.1 type referenced in Table 2, column 3 for types where all the abstract values of the type have the "Basic=Date-Time" property setting.

**28 bis.4.1** The **DATE-TIME-ENCODING** type is:

```

DATE-TIME-ENCODING {Date-Type, Time-Type} ::= SEQUENCE {
    date      Date-Type,
    time      Time-Type}

```

**28 bis.4.2** The encoding shall be the encoding of an instantiation of this type with the **Date-Type** and **Time-Type** actual parameters set to the types specified in Table 2 column 3 of the "Basic=Date" and "Basic=Time" rows (respectively) that specify the additional property settings of all the abstract values of the type.

NOTE – This has been optimized to provide a 32-bit encoding in common cases.

#### **28 bis.5 Encoding subtypes with the "Basic=Interval Interval-type=SE" property setting**

This subclause defines the ASN.1 types referenced in Table 2, column 3 for types where all the abstract values of the type have the "Basic=Interval Interval-type=SE" property setting.

**28 bis.5.1** The **START-END-DATE-INTERVAL-ENCODING** type is:

```

START-END-DATE-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    start      Date-Type,
    end        Date-Type}

```

and the encoding shall be the encoding of an instantiation of this type with the **Date-Type** actual parameter set to the type specified in Table 2 column 3 of the "Basic=Date" row that specifies the additional property settings of all the abstract values of the type. The **start** component shall be set to the start date and the **end** component shall be set to the end date of the interval.

**28 bis.5.2** The **START-END-TIME-INTERVAL-ENCODING** type is:

```

START-END-TIME-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    start      Time-Type,
    end        Time-Type}

```

and the encoding shall be the encoding of an instantiation of this type with the **Time-Type** actual parameter set to the type specified in Table 2 column 3 of the "Basic=Time" row that specifies the additional property settings of all the abstract values of the type. The **start** component shall be set to the start time and the **end** component shall be set to the end time of the interval.

**28 bis.5.3** The `START-END-DATE-TIME-INTERVAL-ENCODING` type is:

```
START-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} ::=
    SEQUENCE {
        start      DATE-TIME-ENCODING {Date-Type, Time-Type},
        end        DATE-TIME-ENCODING {Date-Type, Time-Type}}
```

and the encoding shall be the encoding of an instantiation of this type with the `Date-Type` and `Time-Type` actual parameters set to the types specified in Table 2 column 3 of the "Basic=Date" and "Basic=Time" rows (respectively) that specify the additional property settings of all the abstract values of the type. The `start` component shall be set (as specified in 28 bis.4) to the start date-time and the `end` component shall be set to the end date-time of the interval.

### 28 bis.6 Encoding subtypes with the "Basic=Interval Interval-type=D" property setting

This subclause defines the ASN.1 type referenced in Table 2, column 3 for types where all the abstract values of the type have the "Basic=Interval Interval-type=D" property setting.

**28 bis.6.1** The `DURATION-INTERVAL-ENCODING` type is:

```
DURATION-INTERVAL-ENCODING ::= SEQUENCE { -- 8 bits for optionality
    years      INTEGER (0..31, ..., 32..MAX) OPTIONAL,
               -- 5 bits for up to 31 years
    months     INTEGER (0..15, ..., 16..MAX) OPTIONAL,
               -- 4 bits for up to 15 months
    weeks      INTEGER (0..63, ..., 64..MAX) OPTIONAL,
               -- 6 bits for up to 63 weeks
    days       INTEGER (0..31, ..., 32..MAX) OPTIONAL,
               -- 5 bits for up to 31 days
    hours      INTEGER (0..31, ..., 32..MAX) OPTIONAL,
               -- 5 bits for up to 31 hours
    minutes    INTEGER (0..63, ..., 64..MAX) OPTIONAL,
               -- 6 bits for up to 63 minutes
    seconds    INTEGER (0..63, ..., 64..MAX) OPTIONAL,
               -- 6 bits for up to 63 seconds
    fractional-part SEQUENCE {
        number-of-digits    INTEGER(1..3, ..., 4..MAX),
                           -- 3 bits for up to three digits accuracy
        fractional-value    INTEGER(1..999, ..., 1000..MAX)
                           -- 11 bits for up to three digits accuracy
    } OPTIONAL }
```

**28 bis.6.2** The `weeks` component shall be present if, and only if, the `years`, `months`, `days`, `hours`, `minutes`, and `seconds` components are all absent.

NOTE – This reflects restrictions that are present for the use of time elements in the definition of the `DURATION` abstract value.

**28 bis.6.3** (Canonicalization) If a time element component of the abstract value is zero, and does not have a fractional part, then the corresponding component of `DURATION-INTERVAL-ENCODING` shall be absent unless this time element is the least significant time element in the abstract value. If a time element of the abstract value has the value zero, and is the least significant time element in the abstract value, or has a fractional part, then the corresponding component shall be present in `DURATION-INTERVAL-ENCODING` with the value zero.

**28 bis.6.4** The `fractional-part` of `DURATION-INTERVAL-ENCODING` shall be absent if there is no fractional part of any time element, otherwise it shall be set to the fractional part (of the least significant time element) as specified in 28 bis.6.5.

**28 bis.6.5** The number of digits in the fractional part shall be placed in `number-of-digits`. If the number of digits is N, then the value of the fractional part shall be multiplied by ten-to-the-power-N and the resulting integer value placed in `fractional-value`.

NOTE 1 – Decoders can recover the original fractional part from these encodings, including any trailing zeros.

NOTE 2 – This encoding has been optimized for the cases where there are only a few non-zero time elements in the abstract value, and where the values of the time elements are small. Encodings of less than 16 bits occur in simple cases.

### 28 bis.7 Encoding subtypes with the "Basic=Interval Interval-type=SD" or "Basic=Interval Interval-type=DE" property setting

This subclause defines the ASN.1 types referenced in Table 2, column 3 for types where all the abstract values of the type have the "Basic=Interval Interval-type=SD" or "Basic=Interval Interval-type=DE" property setting.

**28 bis.7.1** The `START-DATE-DURATION-INTERVAL-ENCODING` type is:

```
START-DATE-DURATION-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    start          Date-Type,
    duration       DURATION-INTERVAL-ENCODING }
```

and the encoding shall be the encoding of an instantiation of this type with the `Date-Type` actual parameter set to the type specified in Table 2 column 3 of the "Basic=Date" row that specifies the additional property settings of all the abstract values of the type. The `start` component shall be set to the start date and the `duration` component shall be set (as specified in 28 bis.6) to the duration of the interval.

**28 bis.7.2** The `START-TIME-DURATION-INTERVAL-ENCODING` type is:

```
START-TIME-DURATION-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    start          Time-Type,
    duration       DURATION-INTERVAL-ENCODING }
```

and the encoding shall be the encoding of an instantiation of this type with the `Time-Type` actual parameter set to the type specified in Table 2 column 3 of the "Basic=Time" row that specifies the additional property settings of all the abstract values of the type. The `start` component shall be set to the start time and the `duration` component shall be set (as specified in 28 bis.6) to the duration of the interval.

**28 bis.7.3** The `START-DATE-TIME-DURATION-INTERVAL-ENCODING` type is:

```
START-DATE-TIME-DURATION-INTERVAL-ENCODING {Date-Type, Time-Type} ::=
SEQUENCE {
    start          DATE-TIME-ENCODING {Date-Type, Time-Type},
    duration       DURATION-INTERVAL-ENCODING }
```

and the encoding shall be the encoding of an instantiation of this type with the `Date-Type` and `Time-Type` actual parameters set to the types specified in Table 2 column 3 of the "Basic=Date" and "Basic=Time" rows (respectively) that specify the additional property settings of all the abstract values of the type. The `start` component shall be set (as specified in 28 bis.4) to the start date-time and the `duration` component shall be set (as specified in 28 bis.6) to the duration of the interval.

**28 bis.7.4** The `DURATION-END-DATE-INTERVAL-ENCODING` type is:

```
DURATION-END-DATE-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    duration       DURATION-INTERVAL-ENCODING,
    end           Date-Type }
```

and the encoding shall be the encoding of an instantiation of this type with the `Date-Type` actual parameter set to the type specified in Table 2 column 3 of the "Basic=Date" row that specifies the additional property settings of all the abstract values of the type. The `duration` component shall be set (as specified in 28 bis.6) to the duration of the interval and the `end` component shall be set to the end date.

**28 bis.7.5** The `DURATION-END-TIME-INTERVAL-ENCODING` type is:

```
DURATION-END-TIME-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    duration       DURATION-INTERVAL-ENCODING,
    end           Time-Type }
```

and the encoding shall be the encoding of an instantiation of this type with the `Time-Type` actual parameter set to the type specified in Table 2 column 3 of the "Basic=Time" row that specifies the additional property settings of all the abstract values of the type. The `duration` component shall be set (as specified in 28 bis.6) to the duration of the interval and the `end` component shall be set to the end time.

**28 bis.7.6** The DURATION-END-DATE-TIME-INTERVAL-ENCODING type is:

```
DURATION-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} ::= SEQUENCE {
    duration          DURATION-INTERVAL-ENCODING,
    end               DATE-TIME-ENCODING {Date-Type, Time-Type}}
```

and the encoding shall be the encoding of an instantiation of this type with the **Date-Type** and **Time-Type** actual parameters set to the types specified in Table 2 column 3 of the "**Basic=Date**" and "**Basic=Time**" rows (respectively) that specify the additional property settings of all the abstract values of the type. The **duration** component shall be set (as specified in 28 bis.6) to the duration of the interval and the **end** component shall be set (as specified in 28 bis.4) to the end date-time.

### **28 bis.8 Encoding subtypes with the "Basic=Rec-Interval Interval-type=SE" property setting**

This subclause defines the ASN.1 types referenced in Table 2, column 3 for types where all the abstract values of the type have the "**Basic=Rec-Interval Interval-type=SE**" property setting.

**28 bis.8.1** The REC-START-END-DATE-INTERVAL-ENCODING type is:

```
REC-START-END-DATE-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    recurrence        INTEGER OPTIONAL,
    start             Date-Type,
    end               Date-Type}
```

and the encoding shall be the encoding of an instantiation of this type with the **Date-Type** actual parameter set to the type specified in Table 2 column 3 of the "**Basic=Date**" row that specifies the additional property settings of all the abstract values of the type. The **recurrence** component shall be absent for an unlimited number of recurrences in the abstract value, and shall otherwise be set to the number of recurrences. The **start** component shall be set to the start date and the **end** component shall be set to the end date of the interval.

**28 bis.8.2** The REC-START-END-TIME-INTERVAL-ENCODING type is:

```
REC-START-END-TIME-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    recurrence        INTEGER OPTIONAL,
    start             Time-Type,
    end               Time-Type}
```

and the encoding shall be the encoding of an instantiation of this type with the **Time-Type** actual parameter set to the type specified in Table 2 column 3 of the "**Basic=Time**" row that specifies the additional property settings of all the abstract values of the type. The **recurrence** component shall be absent for an unlimited number of recurrences in the abstract value, and shall otherwise be set to the number of recurrences. The **start** component shall be set to the start time and the **end** component shall be set to the end time of the interval.

**28 bis.8.3** The REC-START-END-DATE-TIME-INTERVAL-ENCODING type is:

```
REC-START-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} ::=
SEQUENCE {
    recurrence        INTEGER OPTIONAL,
    start             DATE-TIME-ENCODING {Date-Type, Time-Type},
    end               DATE-TIME-ENCODING {Date-Type, Time-Type}}
```

and the encoding shall be the encoding of an instantiation of this type with the **Date-Type** and **Time-Type** actual parameters set to the types specified in Table 2 column 3 of the "**Basic=Date**" and "**Basic=Time**" rows (respectively) that specify the additional property settings of all the abstract values of the type. The **recurrence** component shall be absent for an unlimited number of recurrences in the abstract value, and shall otherwise be set to the number of recurrences. The **start** component shall be set (as specified in 28 bis.4) to the start date-time and the **end** component shall be set to the end date-time of the recurring interval.

**28 bis.9 Encoding subtypes with the "Basic=Rec-Interval Interval-type=D" property setting**

This subclause defines the ASN.1 type referenced in Table 2, column 3 for types where all the abstract values of the type have the "Basic=Rec-Interval Interval-type=D" property setting.

**28 bis.9.1** The REC-DURATION-INTERVAL-ENCODING type is:

```
REC-DURATION-INTERVAL-ENCODING ::= SEQUENCE {
    recurrence      INTEGER OPTIONAL,
    duration        DURATION-INTERVAL-ENCODING}
```

**28 bis.9.2** The **recurrence** component shall be absent for an unlimited number of recurrences in the abstract value, and shall otherwise be set to the number of recurrences. The **duration** component shall be set (as specified in 28 bis.6) to the duration of the recurring interval.

**28 bis.10 Encoding subtypes with the "Basic=Rec-Interval Interval-type=SD" or "Basic=Rec-Interval Interval-type=DE" property setting**

This subclause defines the ASN.1 types referenced in Table 2, column 3 for types where all the abstract values of the type have the "Basic=Rec-Interval Interval-type=SD" or "Basic=Rec-Interval Interval-type=DE" property setting.

**28 bis.10.1** The REC-START-DATE-DURATION-INTERVAL-ENCODING type is:

```
REC-START-DATE-DURATION-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    recurrence      INTEGER OPTIONAL,
    start           Date-Type,
    duration        DURATION-INTERVAL-ENCODING}
```

and the encoding shall be the encoding of an instantiation of this type with the **Date-Type** actual parameter set to the type specified in Table 2 column 3 of the "Basic=Date" row that specifies the additional property settings of all the abstract values of the type. The **recurrence** component shall be absent for an unlimited number of recurrences in the abstract value, and shall otherwise be set to the number of recurrences. The **start** component shall be set to the start date and the **duration** component shall be set (as specified in 28 bis.6) to the duration of the interval.

**28 bis.10.2** The REC-START-TIME-DURATION-INTERVAL-ENCODING type is:

```
REC-START-TIME-DURATION-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    recurrence      INTEGER OPTIONAL,
    start           Time-Type,
    duration        DURATION-INTERVAL-ENCODING }
```

and the encoding shall be the encoding of an instantiation of this type with the **Time-Type** actual parameter set to the type specified in Table 2 column 3 of the "Basic=Time" row that specifies the additional property settings of all the abstract values of the type. The **recurrence** component shall be absent for an unlimited number of recurrences in the abstract value, and shall otherwise be set to the number of recurrences. The **start** component shall be set to the start time and the **duration** component shall be set (as specified in 28 bis.6) to the duration of the interval.

**28 bis.10.3** The REC-START-DATE-TIME-DURATION-INTERVAL-ENCODING type is:

```
REC-START-DATE-TIME-DURATION-INTERVAL-ENCODING {Date-Type, Time-Type} ::=
SEQUENCE {
    recurrence      INTEGER OPTIONAL,
    start           DATE-TIME-ENCODING {Date-Type, Time-Type},
    duration        DURATION-INTERVAL-ENCODING }
```

and the encoding shall be the encoding of an instantiation of this type with the **Date-Type** and **Time-Type** actual parameters set to the types specified in Table 2 column 3 of the "Basic=Date" and "Basic=Time" rows (respectively) that specify the additional property settings of all the abstract values of the type. The **recurrence** component shall be absent for an unlimited number of recurrences in the abstract value, and shall otherwise be set to the number of recurrences. The **start** component shall be set (as specified in 28 bis.4) to the start date-time and the **duration** component shall be set (as specified in 28 bis.6) to the duration of the recurring interval.

**28 bis.10.4** The REC-DURATION-END-DATE-INTERVAL-ENCODING type is:

```
REC-DURATION-END-DATE-INTERVAL-ENCODING {Date-Type} ::= SEQUENCE {
    recurrence      INTEGER OPTIONAL,
    duration        DURATION-INTERVAL-ENCODING,
    end             Date-Type }
```

and the encoding shall be the encoding of an instantiation of this type with the **Date-Type** actual parameter set to the type specified in Table 2 column 3 of the "**Basic=Date**" row that specifies the additional property settings of all the abstract values of the type. The **recurrence** component shall be absent for an unlimited number of recurrences in the abstract value, and shall otherwise be set to the number of recurrences. The **duration** component shall be set (as specified in 28 bis.6) to the duration of the interval and the **end** component shall be set to the end date.

**28 bis.10.5** The REC-DURATION-END-TIME-INTERVAL-ENCODING type is:

```
REC-DURATION-END-TIME-INTERVAL-ENCODING {Time-Type} ::= SEQUENCE {
    recurrence      INTEGER OPTIONAL,
    duration        DURATION-INTERVAL-ENCODING,
    end             Time-Type }
```

and the encoding shall be the encoding of an instantiation of this type with the **Time-Type** actual parameter set to the type specified in Table 2 column 3 of the "**Basic=Time**" row that specifies the additional property settings of all the abstract values of the type. The **recurrence** component shall be absent for an unlimited number of recurrences in the abstract value, and shall otherwise be set to the number of recurrences. The **duration** component shall be set (as specified in 28 bis.6) to the duration of the interval and the **end** component shall be set to the end time.

**28 bis.10.6** The REC-DURATION-END-DATE-TIME-INTERVAL-ENCODING type is:

```
REC-DURATION-END-DATE-TIME-INTERVAL-ENCODING {Date-Type, Time-Type} ::=
SEQUENCE {
    recurrence      INTEGER OPTIONAL,
    duration        DURATION-INTERVAL-ENCODING,
    end             DATE-TIME-ENCODING {Date-Type, Time-Type}}
```

and the encoding shall be the encoding of an instantiation of this type with the **Date-Type** and **Time-Type** actual parameters set to the types specified in Table 2 column 3 of the "**Basic=Date**" and "**Basic=Time**" rows (respectively) that specify the additional property settings of all the abstract values of the type. The **recurrence** component shall be absent for an unlimited number of recurrences in the abstract value, and shall otherwise be set to the number of recurrences. The **duration** component shall be set (as specified in 28 bis.6) to the duration of the interval and the **end** component shall be set (as specified in 28 bis.4) to the end date-time.

### **28 bis.11 Encoding subtypes with mixed settings of the Basic property**

This subclause specifies the encoding for the **TIME** type and subsets of that type whose abstract values do not all have the same setting of the **Basic** property or for which there is no applicable row in Table 2 (for example, because of the use of multiple accuracies – see 28 bis.1.3). It defines and uses the types **DATE-TYPE**, **TIME-TYPE**, and **MIXED-ENCODING** (see 28 bis.11.5 to 28 bis.11.6). These types are defined using the ASN.1 types defined in earlier subclauses.

**28 bis.11.1** For all abstract values of the **TIME** type, there is exactly one row of Table 2 for which the property settings specified in column 2 match the property settings of the abstract value, for all of those property settings that are listed in column 2. (The abstract value may have additional property settings not listed in column 2.) This is called the main determining row.

**28 bis.11.2** If the main determining row is row 33, 34, 36, 38, 40, 41, 43, 44, 46, 48, 50, 51, or 53, there is a requirement that the additional properties match those specified in one of rows 1 to 14. The applicable row 1 to 14 is called the date determining row.

**28 bis.11.3** If the main determining row is row 33, 35, 36, 39, 40, 42, 43, 45, 46, 49, 50, 52 or 53, there is a requirement that the additional properties match those specified in one of rows 15 to 32. The applicable row 15 to 32 is called the time determining row.

**28 bis.11.4** In the **DATE-TYPE**, **TIME-TYPE** and **MIXED-ENCODING** type, the **row-n** alternative shall be selected if the date determining row, the time determining row, or the main determining row (respectively) is row *n*.

28 bis.11.5 The encoding of the abstract value shall be the encoding of the MIXED-ENCODING type:

```

MIXED-ENCODING ::= CHOICE {
  row-1    CENTURY-ENCODING,
  row-2    ANY-CENTURY-ENCODING,
  row-3    YEAR-ENCODING,
  row-4    ANY-YEAR-ENCODING,
  row-5    YEAR-MONTH-ENCODING,
  row-6    ANY-YEAR-MONTH-ENCODING,
  row-7    DATE-ENCODING,
  row-8    ANY-DATE-ENCODING,
  row-9    YEAR-DAY-ENCODING,
  row-10   ANY-YEAR-DAY-ENCODING,
  row-11   YEAR-WEEK-ENCODING,
  row-12   ANY-YEAR-WEEK-ENCODING,
  row-13   YEAR-WEEK-DAY-ENCODING,
  row-14   ANY-YEAR-WEEK-DAY-ENCODING,
  row-15   HOURS-ENCODING,
  row-16   HOURS-UTC-ENCODING,
  row-17   HOURS-AND-DIFF-ENCODING,
  row-18   MINUTES-ENCODING,
  row-19   MINUTES-UTC-ENCODING,
  row-20   MINUTES-AND-DIFF-ENCODING,
  row-21   TIME-OF-DAY-ENCODING,
  row-22   TIME-OF-DAY-UTC-ENCODING,
  row-23   TIME-OF-DAY-AND-DIFF-ENCODING,
  row-24   HOURS-AND-FRACTION-ENCODING,
  row-25   HOURS-UTC-AND-FRACTION-ENCODING,
  row-26   HOURS-AND-DIFF-AND-FRACTION-ENCODING,
  row-27   MINUTES-AND-FRACTION-ENCODING,
  row-28   MINUTES-UTC-AND-FRACTION-ENCODING,
  row-29   MINUTES-AND-DIFF-AND-FRACTION-ENCODING,
  row-30   TIME-OF-DAY-AND-FRACTION-ENCODING,
  row-31   TIME-OF-DAY-UTC-AND-FRACTION-ENCODING,
  row-32   TIME-OF-DAY-AND-DIFF-AND-FRACTION-ENCODING,
  row-33   DATE-TIME-ENCODING {DATE-TYPE, TIME-TYPE},
  row-34   START-END-DATE-INTERVAL-ENCODING {DATE-TYPE},
  row-35   START-END-TIME-INTERVAL-ENCODING {TIME-TYPE},
  row-36   START-END-DATE-TIME-INTERVAL-ENCODING {DATE-TYPE, TIME-TYPE},
  row-37   DURATION-INTERVAL-ENCODING,
  row-38   START-DATE-DURATION-INTERVAL-ENCODING {DATE-TYPE},
  row-39   START-TIME-DURATION-INTERVAL-ENCODING {TIME-TYPE},
  row-40   START-DATE-TIME-DURATION-INTERVAL-ENCODING {DATE-TYPE, TIME-TYPE},
  row-41   DURATION-END-DATE-INTERVAL-ENCODING {DATE-TYPE},
  row-42   DURATION-END-TIME-INTERVAL-ENCODING {TIME-TYPE},
  row-43   DURATION-END-DATE-TIME-INTERVAL-ENCODING {DATE-TYPE, TIME-TYPE},
  row-44   REC-START-END-DATE-INTERVAL-ENCODING {DATE-TYPE},
  row-45   REC-START-END-TIME-INTERVAL-ENCODING {TIME-TYPE},
  row-46   REC-START-END-DATE-TIME-INTERVAL-ENCODING {DATE-TYPE, TIME-TYPE},
  row-47   REC-DURATION-INTERVAL-ENCODING,
  row-48   REC-START-DATE-DURATION-INTERVAL-ENCODING {DATE-TYPE},
  row-49   REC-START-TIME-DURATION-INTERVAL-ENCODING {TIME-TYPE},
  row-50   REC-START-DATE-TIME-DURATION-INTERVAL-ENCODING
            {DATE-TYPE, TIME-TYPE},
  row-51   REC-DURATION-END-DATE-INTERVAL-ENCODING {DATE-TYPE},
  row-52   REC-DURATION-END-TIME-INTERVAL-ENCODING {TIME-TYPE},
  row-53   REC-DURATION-END-DATE-TIME-INTERVAL-ENCODING
            {DATE-TYPE, TIME-TYPE} }

```

where the encoding of the type of each alternative shall be as specified in the subclause identified in Table 2, column 3 of the main determining row.

**28 bis.11.6** The **DATE-TYPE** is:

```
DATE-TYPE ::= CHOICE {
    row-1    CENTURY-ENCODING,
    row-2    ANY-CENTURY-ENCODING,
    row-3    YEAR-ENCODING,
    row-4    ANY-YEAR-ENCODING,
    row-5    YEAR-MONTH-ENCODING,
    row-6    ANY-YEAR-MONTH-ENCODING,
    row-7    DATE-ENCODING,
    row-8    ANY-DATE-ENCODING,
    row-9    YEAR-DAY-ENCODING,
    row-10   ANY-YEAR-DAY-ENCODING,
    row-11   YEAR-WEEK-ENCODING,
    row-12   ANY-YEAR-WEEK-ENCODING,
    row-13   YEAR-WEEK-DAY-ENCODING,
    row-14   ANY-YEAR-WEEK-DAY-ENCODING }
```

where the encoding of the type of each alternative shall be as specified in the subclause identified in Table 2, column 3 of the date determining row.

**28 bis.11.7** The **TIME-TYPE** is:

```
TIME-TYPE ::= SEQUENCE {
    number-of-digits INTEGER (1..MAX) OPTIONAL,
    time-type CHOICE {
        row-15   HOURS-ENCODING,
        row-16   HOURS-UTC-ENCODING,
        row-17   HOURS-AND-DIFF-ENCODING,
        row-18   MINUTES-ENCODING,
        row-19   MINUTES-UTC-ENCODING,
        row-20   MINUTES-AND-DIFF-ENCODING,
        row-21   TIME-OF-DAY-ENCODING,
        row-22   TIME-OF-DAY-UTC-ENCODING,
        row-23   TIME-OF-DAY-AND-DIFF-ENCODING,
        row-24   HOURS-AND-FRACTION-ENCODING,
        row-25   HOURS-UTC-AND-FRACTION-ENCODING,
        row-26   HOURS-AND-DIFF-AND-FRACTION-ENCODING,
        row-27   MINUTES-AND-FRACTION-ENCODING,
        row-28   MINUTES-UTC-AND-FRACTION-ENCODING,
        row-29   MINUTES-AND-DIFF-AND-FRACTION-ENCODING,
        row-30   TIME-OF-DAY-AND-FRACTION-ENCODING,
        row-31   TIME-OF-DAY-UTC-AND-FRACTION-ENCODING,
        row-32   TIME-OF-DAY-AND-DIFF-AND-FRACTION-ENCODING } }
```

where the encoding of the type of each alternative shall be as specified in the subclause identified in Table 2, column 3 of the time determining row.

**28 bis.11.8** The **number-of-digits** shall be present in the **TIME-TYPE** if and only if the **time-type** alternative is one of **row-24** to **row-32**. It shall encode the number of digits in the fractional part of the abstract value.



## SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
<b>Series X</b>	<b>Data networks, open system communications and security</b>
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems