



中华人民共和国国家标准

GB/T×××—××××

信息技术 学习、教育和培训 基于规则的 XML 绑定技术

Information technology - Learning, education and training -
Rule-based XML binding techniques

(报批稿)

××××-××-××发布

××××-××-××实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会

发布

目 次

| | |
|---------------------|-----|
| 前 言 | II |
| 引 言 | III |
| 1 范围 | 1 |
| 2 规范性引用文件 | 1 |
| 3 术语和定义 | 1 |
| 4 缩略语 | 8 |
| 5 XML 编码绑定模板 | 8 |
| 5.1 生成和产生 XML | 8 |
| 5.2 消费和解释 XML | 12 |
| 5.3 基本数据类型的表示 | 13 |
| 5.4 字符的编码 | 14 |
| 5.5 对异常和扩展的处理 | 14 |

前 言

本标准由教育部提出。

本标准由全国信息技术标准化委员会归口。

本标准起草单位：清华大学。

本标准主要起草人：史元春、郑莉、沈中南、向欣、郭玲、许磊、刘银涛。

引 言

在网络教育技术标准使用的过程中，通常会存在数据绑定的问题。XML 作为一种通用的可扩展置标语言，也是学习技术标准的编码绑定的首选。本标准规定了将网络教育技术标准的信息模型做 XML 编码绑定的方法。该方法能够指导标准制订者制订网络教育技术标准的相应 XML 绑定规范和实践指南。

信息技术 学习、教育和培训

基于规则的 XML 绑定技术

1 范围

本标准描述了用于数据模型的基于规则的 XML 编码绑定技术。本标准的措词可以被包含到相关标准中，用于支持 XML 绑定。该措词描述了基于规则的方法，用于描述相关标准中的词汇。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

GB/T 1988—1998 信息技术 信息交换用七位编码字符集 (eqv ISO/IEC 646: 1991)

GB/T 7408—2005 数据元和交换格式 信息交换 日期和时间表示法 (ISO 8601: 2000, MOD)

GB 13000.1—1993 信息技术 通用多八位编码字符集 (UCS) 第一部分：体系结构与基本多文种平面 (idt ISO/IEC 10646-1: 1993)

GB/T 15272—1994 程序设计语言C (idt ISO/IEC 9899: 1990)

GB/T 15273.1—1994 信息技术 八位单字节编码图形字符集 第一部分：拉丁字母 (idt ISO 8859-1: 1987)

GB/T 18221—2000 信息技术 程序设计语言 环境与系统软件接口 独立于语言的数据类型 (idt ISO/IEC 11404: 1996)

GB/T 18793—2002 信息技术 可扩展置标语言(XML)1.0

RFC 822 ARPA 互联网文本消息格式标准

IETF RFC 2068 超文本传送协议 (HTTP/1.1)

ANSI X3.42 (1990) 信息交换中字符串的表示

ANSI X.3.285 (1998) 数据表示的元数据模型

3 术语和定义

下列术语和定义适用于本标准。

3.1

聚集 (数据类型、值) aggregate (data type, value)

生成的数据类型或值。原则上，每个这样的数据类型或值都是由组件类型或值构成的。这种数据类型或值是通过将算法过程应用于组件数据类型或值而生成的。组件的值可以通过特征操作进行访问。聚集的特性与其组件的特性无关。

示例 1：数组聚集包含的组件全部具有相同的类型。特征操作使用索引（编号）来访问各个的组件。

```
my_array:
    array (0..9) of (integer), // 整数数组
my_array(4) // 访问第四个元素
```

示例 2: 记录的聚集包含若干成分, 每个成分各自有类型和标签。特征操作使用元素名 (标识符) 访问各个组件。

```
A: record
(
  B: integer,
  C: void,
  D: characterstring(GB13000-1),
),
A.B // 访问标签为 B 的元素
```

3.2

绑定 binding

从一个框架或规范到另一个框架或规范的应用或映射。

3.3

编码 coding

1) 在信息交换中, 信息的格式化或结构化表示。

另见: 信息编码 (3.10)

2) 按某一结构表示信息的过程。

3.4

条件数据元素 conditional data element

在适当的上下文中, 满足某些特定条件时, 在数据结构的实例中被定义和要求的数据结构元素。

数据元素的“条件”特征是约束属性。

另见: 扩展数据元素 (3.11)、必备数据元素 (3.19)、约束 (数据元素) (3.21)、可选数据元素 (3.23)。

3.5

消费 (数据) consume (data)

读数据并找到词法或编码的边界范围。

其它形式: 消费数据、数据消费者、数据消费。

另见: 解释 (数据) (3.16)、产生 (数据) (3.24)。

注: 数据在解释前被消费。

示例 1: 在下列字符流中:

```
<R>
  <A>123.45</A>
  <B>PQR</B>
  <C X="Y">Z</C>
</R>
```



```

<R>
  <D>JKL</D>
  <E>
    <F>XXX</F>
    <G>YYY</G>
  </E>
</R>

```

数据消费者可能识别到：

- 有两个记录，都用标签“R”标识。
- 第一个“R”记录包含三个记录，分别用标签“A”、“B”、“C”标识。
- 第二个“R”记录包含两个记录，分别用标签“D”、“E”标识。

但是，数据消费者：

- 可能不知道标签的意义，如：. . .有什么意义？
- 可能无法验证标签的合法性，如：“<C>”可以包含属性“X”吗？
- 可能无法验证记录内容的合法性，如：在记录“A”中，“123.45”是合法的值吗？
- 可能限制分析的深度，如：“R”仅仅深度分析一层，该层包含标签“D”和“E”，但对“E”的内容只是进行有限的分析（如：找到开始和结束标签），标签“F”和“G”不会被分析和发现。因此，数据消费者对信息结构可能只有部分理解。

示例 2：下面是一个 API 的例子，用以区分数据消费和数据解释，在此种情况下扩展元素被间接使用了，而实现仍是严格一致的。

```

//// 这个例子涉及到两个文件：头文件“std_data.h”和包含该头的严格一致的应用

//// 下面是被应用包含的头文件“std_data.h”的内容
struct std_data
{
    int std_element_1;    // 必备元素
    void *std_element_2; // 可选元素
    int ext_element_3;   // 扩展元素
};

//// 严格一致的应用开始
// 包括标准的头文件（内容如上所列）
#include "std_data.h"

struct std_data x; // 声明“x”作为标准数据

my_code()
{
    struct std_data y,z; // 声明“y”和“z”

    //这是严格一致的代码，
    //然而扩展元素“ext_element_3”中的内容也被复制了

```

```

memcpy(&y,&x,sizeof x);

// 将字符串赋给“std_element_2”
// 将长度值赋给“std_element_1”
y.std_element_2 = "hello there";
y.std_element_1 = strlen(y.std_element_2);

//这还是严格一致的代码,
//然而扩展元素“ext_element_3”中的内容也被复制了
memcpy(&z,&y,sizeof y);
}

```

本例是严格一致的，因为在实现过程中只解释或生成了来自标准集合的元素（即“std_element_1”和“std_element_2”）。在这个假设的API绑定中，memcpy（在内存中拷贝对象）操作与消费和产生操作是等价的，而直接访问元素（如：y.std_element_1）在这个假设的API绑定中是解释和生成操作。

3.6

数据实例 data instance

通过某种绑定规则呈现的数据集。

3.7

数据对象 data object

在针对实现时数据访问的概念模型中，一个数据处理的单元。

注1：数据对象可以是数据元素或实现中定义的对象。严格一致的实现中只使用或访问作为数据元素的数据对象。

注2：进一步通过语义进行定义和限制的数据对象的行为，就是一个数据结构。一个数据结构的实例是一个数据集。在某种绑定中进一步定义、限制和表示的数据集即是数据实例。

另见：数据实例（3.6）、数据集（3.8）、数据结构（3.9）

3.8

数据集 data set

在3.9中定义b)所指的数据结构，即：零个或多个数据元素聚集而成的实例。

注：数据集是独立于绑定的（与绑定无关）。

3.9

数据结构 data structure

有下列两个定义：

- a) 零个或多个数据元素聚集而成的数据类型。
- b) 零个或多个数据元素聚集而成的实例。

注1：在不同的上下文中，一个数据结构可能被认为是一个完整的不可分单元，即在该上下文中，一个数据结构是某更高级别数据结构中的数据元素。

注2：术语“聚集”是在GB/T 18221—2000中定义的。

示例：记录、集合、序列、列表、数组。

3.10

信息编码 encoding

信息的位或字节格式及其表示。

注：信息编码也可以简称为“编码”。

3.11

扩展数据元素 extended data element

在适当的上下文中，定义在标准之外的数据结构元素，经数据交换的参与者和实现系统允许，可以在数据结构的实例中使用。

数据元素的“扩展”属性是一种约束属性。

数据元素的“扩展”属性是一个一致性级别的特征（如：严格一致性实现和一致性实现是两个不同的级别）。

示例：必备扩展数据元素、可选扩展数据元素、条件扩展数据元素。

另见：条件数据元素（3.4）、必备数据元素（3.19）、约束（数据元素）（3.21）、可选数据元素（3.23）。

3.12

生成（数据） generate(data)

将数据的从本来的含义转换为某种适合数据交换的形式。

示例：根据某种概念模型将数据结构序列化，但不必以某种具体的编码方式表示数据。

另见：解释（数据）（3.16）、产生（数据）（3.24）。

3.13

实现行为 implementation behavior

外部观察到的现象、对外呈现的行为或动作。

另见：实现时定义的行为（3.14）、实现值（3.15）、未定义的行为/值（3.27）、未规定的行为/值（3.28）。

3.14

实现时定义的行为/值 implementation-defined behavior/value

未规定的行为或未规定的值，由实现时的文档说明如何对这些行为或值做出选择。

示例：以8为字节计算，允许一个编码时的最大容量

另见：实现行为（3.13）、未定义的行为/值（3.27）、未规定的行为/值（3.28）。

3.15

实现值 implementation value

与实现相关的可计量的对象。

另见：实现行为（3.13）、实现时定义的行为/值（3.14）、未定义的行为/值（3.27）、未规定的行为/值（3.28）。

3.16

解释（数据） interpret (data)

处理数据，在本标准规定的范围内发现数据的含义。

其它形式：解释数据，数据解释器，数据解释。

另见：生成（数据）（3.12）、消费（数据）（3.5）。

注：数据在解释前被消费。

示例：在下列字符流中：

```
<R>
  <A>123.45</A>
  <B>PQR</B>
  <C X="Y">Z</C>
</R>
<R>
  <D>JKL</D>
  <E>
    <F>XXX</F>
    <G>YYY</G>
  </E>
</R>
```

一个数据消费者能够识别：

——有两个记录，都用标签“R”标识。

——第一个“R”记录包含三个记录，分别用标签“A”、“B”、“C”标识。

——第二个“R”记录包含两个记录，分别用标签“D”、“E”标识。

因为只识别了这些标签，所以只有这些可以提供给数据解释。假设标签“E”代表一个扩展数据元素，那么一个数据解释器可能只能识别标准标签“A”、“B”、“C”和“D”。

基于翻译过程中“消费”和“解释”的分离，以及某种特殊的标准绑定（如：本例中类似 XML 的绑定）两个原因，一个应用程序可能只解释标准化的特征 A、B、C 和 D。

如上所述，一个应用程序结合了数据消费和数据解释，但只解释标准的数据元素，可以作为一个严格一致的数据阅读器。

3.17**本地特定行为 local-specific behavior**

依赖于民族、文化、语言、制度等地方约定的行为，在实现中需要用文档说明。

3.18**寿命（数据元素） longevity (data element)**

数据元素规范中的一个属性，用于说明与过去、现在或将来的标准版本进行合并的意图。

另见：约束（数据元素）（3.21）、过时数据元素（3.22）、保留数据元素（3.26）。

注：寿命属性和约束属性无关。

示例 1：过时数据元素也许已包含在本标准的过去版本中，但不会包含在本标准的将来版本中。

示例 2：保留数据元素可能没有被包含在本标准的过去版本中，而可能会包括在本标准的将来版本中。

3.19**必备数据元素 mandatory data element**

在适当的上下文中，在数据结构实例中定义和要求的元素。

“必备”是一种约束型属性。

另见：条件数据元素（3.4）、扩展数据元素（3.11）、约束（数据元素）（3.21）、可选数据元素（3.23）。

3.20

漫游（访问、系统） nomadic (access, system)

a) 服务的连续性跨越不同的通信会话和地理位置的现象。

b) 间或性地从子系统或相关系统之间的通信网络上断开。

注：也称为“间或性连通”或“间或性漫游”。

3.21

约束（数据元素） obligation (data element)

决定数据结构的合法性的，对数据元素的要求和容许。

另见：寿命（数据元素）（3.18）、条件数据元素（3.4）、扩展数据元素（3.11）、必备数据元素（3.19）、可选数据元素（3.23）。

示例：数据结构 X，有四个元素：A 和 B 是必备的、C 是可选的、D 是条件的（如果 B 的值为真，则 D 存在）。

下列是有效和无效数据结构的样例：

```
( A=123 ) // 无效，缺少必备元素 B
( A=123, B=false ) // 有效
( A=123, B=true ) // 无效，缺少条件元素 D
( A=123, B=true, D=17 ) // 有效
( A=123, B=false, D=17 ) // 有效
( A=123, B=nil, C=345 ) // 有效
```

3.22

过时数据元素 obsolete data element

在适当的上下文中，在数据结构中定义了但不宜在数据结构的实例中使用的元素。

“过时”是一种寿命属性。

另见：寿命（数据元素）（3.18）、保留数据元素（3.26）。

注：不推荐使用过时数据元素，可能从标准的将来版本中被删除。

3.23

可选数据元素 optional data element

在适当的上下文中，在数据结构中定义并允许的、但在实例中不是必须出现的元素。

另见：条件数据元素（3.4）、扩展数据元素（3.11）、必备数据元素（3.19）、约束（数据元素）（3.21）。

3.24

产生（数据） produce (data)

对数据进行处理，定义出词法或编码的边界，然后写出结果数据。

其它形式：产生数据、数据产生器、数据生产

另见：生成（数据）（3.12）、消费（数据）（3.5）。

注：数据被生成先于被产生。

3.25

信息仓库 repository

数据集合以及用于存储、索引、查找和提取信息的数据访问方法的汇集。

3.26

保留数据元素 reserved data element

在适当的下文中，在数据结构中未被定义且在实例中不允许使用的元素。

“保留”是一种寿命属性。

另见：寿命（数据元素）（3.18）、过时数据元素（3.22）。

3.27

未定义的行为/值 undefined behavior/value

标准未实施要求的实现行为或实现值。

另见：实现行为（3.13）、实现时定义的行为/值（3.14）、实现值（3.15）、未规定的行为/值（3.28）。

示例 1：可能的未定义的行为包括（但不局限于）：

- 完全忽略某种情形，
- 不可预知的结果，
- 以文档化式来表现的环境特征，
- 终止处理。

示例 2：可能的未定义的值包括无穷大、空值、非数字。

3.28

未规定的行为/值 unspecified behavior/value

实现行为或实现值，标准为之提供了两个或更多的可能性，但在未要求在任何实例中选择哪一个。

另见：实现行为（3.13）、实现时定义的行为/值（3.14）、实现值（3.15）、未定义的行为/值（3.27）。

示例 1：应用程序在创建对象的标识符时对算法的选择。

示例 2：将过程调用参数压入调用栈的次序。

4 缩略语

API Application Programming Interface 应用程序接口。

HTTP Hypertext Transfer Protocol 超文本传送协议

ICS Implementation Conformance Statement 实现一致性声明

IETF Internet Engineering Task Force 互联网工程任务组

L10N localization 本地化

LID Language Independent Datatypes 独立于语言的数据类型

MDAS API Metadata Access Service API 元数据访问服的应用程序接口

RFC Request for Comments 请求评论

SPM smallest permitted maximum 最低峰值

W3C World Wide Web Consortium 万维网集团

XML Extensible Markup Language 可扩展置标语言

5 XML 编码绑定模板

5.1 生成和产生 XML

5.1.1 规则

下列规则描述了 XXX 标准的数据元素（如同本标准与 GB/T 18221—2000 记法所描述）到 XML 记录的转换。

注：本标准中出现的“XXX 标准”应使用包含该措词的标准名称来代替。

- a) 规则 1：除规则 2 中说明的部分以外，对于 GB/T 18221—2000 记法中的每个数据元素，把各标识符映射到 XML 标签。对称的 XML 标签界定了数据元素的相关值的边界。XML 标签的嵌套表示了数据元素的结构。对于数组和顺序聚集：
 - 1) 与聚集标识符名称相同的 XML 标签名称，表示聚集组。
 - 2) 单个的数据元素用重复的 XML 标签来表示，这些 XML 标签基于聚集的标识符及后缀“_list”或“_bucket”，不用元素的索引值。
- b) 规则 2：将所有的“多语言字符串”数据类型映射到：
 - 1) 用“多语言字符串”数据类型的“语种”元素在 XML 父元素中设置 LANG 属性。
 - 2) 用“字符串”元素设置父标签的内容。
- c) 规则 3：将下面的 XML 标签（通配记法）

XXX 标准_*

转换为下列 XML 标签（通配记法）：

GBT_XXX 标准_*

通过上述规则产生的所有数据都应该是格式正确的 XML。

5.1.2 基本原理

5.1.2.1 规则 1 的基本原理

规则 1 是指从 GB/T 18221—2000 数据类型到 XML 标签约定的主要转换。下列例子使用下列定义来说明转换：

```

A: record
(
  B: integer,
  C: record
  (
    D: integer,
    E: characterstring(GB-13000-1),
  ),
  F_list: array (0..limit) of (integer),
  G: sample_mlstring_list_type,
)

```

规则 1 的第 1 句：“对于 GB/T 18221—2000 记法中的每个数据元素，把各标识符映射到 XML 标签”，转换标识符，如：“x:” ⇒ “<x>”。

规则 1 的第 2 句：“对称的 XML 标签界定了数据元素的相关值的边界”，要求

—标签是对称的

—数据元素的值在标签之间，如：“x:17” ⇒ “<x>17</x>”。

规则 1 的第 3 句：“XML 标签的嵌套表示了数据元素的结构”，要求由聚集（记录、数组、序列或列表）所隐含的嵌套得出同样的 XML 标签的嵌套。利用上述对记录 A 的定义，元素 B、C、D 和 E 隐含了下列嵌套：

```
<A>
  <B>...</B>
  <C>
    <D>...</D>
    <E>...</E>
  </C>
  ...
</A>
```

规则 1 的第 4 句：“对于数组和顺序聚集：单个的数据元素用重复的 XML 标签来表示，这些 XML 标签基于聚集的标识符及后缀“_list”或“_bucket”，不用元素的索引值”，要求数组和序列（列表）被表示为带有相同名称的多个标签——这是典型的 XML 约定。如：数据元素 F 表示为：

```
<!--对 F_list 的正确的 XML 绑定 -->
<A>
  ...
  <F_list>
    <F>...</F>
    <F>...</F>
    <F>...</F>
  </F_list>
  ...
</A>
```

但不能表示为：

```
<!--对 F_list 的不正确的 XML 绑定-->
<A>
  ...
  <F_list>
    <0>...</0>
    <1>...</1>
    <2>...</2>
  </F_list>
  ...
</A>
```

5.1.2.2 规则 2 的基本原理

XXX 标准的记录使用几种特定的数据类型，如多语言数据类型，用来描述在多语言和多文化上下

文中必须标识的某些特征类型数据元素——这些特征通常称为国际化 (I18N) 和本地化 (L10N) 特征。下面是多语言数据类型的示例，这种多语言数据类型不与本标准其他位置所定义的其他多语言数据类型相抵触。在这个示例中，数据元素 `sample_mlstring_type` 表示一个字符串对：本地化字符串和本地规范 (L10N 映射)。数据类型 `sample_mlstring_array_type` 表示一个字符串对的数组。在这个示例中，数组 `example_remarks` 包含四个元素，每个元素是一个字符串对。可以推测，应用可以根据所在的国家（地区）从 `example_remarks` 中选择适当的字符串。下面是示例的类型定义和值定义：

```

type sample_mlstring_type =
  record
  (
    L10N_string: characterstring(GB-13000-1),
    L10N_locale: string_type,
  ),

type sample_mlstring_array_type =
  array (0..limit) of (sample_mlstring_type),

value example_remarks:
  sample_mlstring_array_type =
  (
    (
      L10N_string: "abc abc abc",
      L10N_locale: "en-US",
    ),
    (
      L10N_string: "def def def",
      L10N_map: "fr-CA",
    ),
    (
      L10N_string: "ghi ghi ghi",
      L10N_map: "de-DE",
    ),
    (
      L10N_string: "您好 您好 您好",
      L10N_map: "zh-CN",
    ),
  ),

```

根据规则 2 和规则 1 中的数组处理规则，这些数据元素可以被转换成下列 XML：

```

<example_remarks LANG="en-US">abc abc abc</example_remarks>
<example_remarks LANG="fr-CA">def def def</example_remarks>
<example_remarks LANG="de-DE">ghi ghi ghi</example_remarks>

```

```
<example_remarks LANG="zh-CN">您好 您好 您好</example_remarks>
```

5.1.2.3 规则 3 的基本原理

这个规则用于重写标签，以使用某些名称空间约定（namespace conventions）。该规则可以通过选择不同的名称空间约定（前缀）来规定 XML 的名称空间。

在使用规则 3 之后，实现要确保这些转换的结果是格式正确的 XML。

5.2 消费和解释 XML

5.2.1 规则

下列规则描述了通过使用本标准和 GB/T 18221—2000 的标记法，将 XML 记录转换到 XXX 标准数据元素。

被消费的数据都应该是格式正确的 XML。

a) 规则 1：将下列 XML 标签（通配记法）：

GBT_XXX 标准_*

转换成下列 XML 标签（通配记法）：

xxx 标准_*

b) 规则 2：转换下列内容：

- 1) 用 XML 元素的 LANG 属性设置“多语言字符串”数据元素中的“语种”元素。
- 2) 用标签标记的元素中的内容设置“多语言字符串”数据元素中的“字符串”元素。

c) 规则 3：对于在本标准中每一个与 XXX 标准所定义的数据元素标识符有关系的 XML 标签，需要它对对应的开、闭标签进行匹配。对于每一个 XML 标签，除了规则 2 中修改的部分，需要把该标签映射为对应的数据元素标识符。XML 标签的嵌套结构表示了数据元素的嵌套结构，如上述 5.1 中的规则 1 的逆操作即是生成和解释 XML。每个用标签标记的元素的内容被转换成相应的数据元素的值。

5.2.2 基本原理

5.2.2.1 规则1的基本原理

在处理前，实现要保证进行消费和解释的内容是格式正确的 XML。

此规则将根据需要剥离 XML 名称空间的前缀和后缀。在规则 1 的样例中，没有使用 XML 的名称空间，而是使用了一个名称空间的前缀“GBT_”来减少名称空间冲突的可能性。

5.2.2.2 规则2的基本原理

此规则做了从 LANG 属性到“多语言字符串”数据类型的逆映射。此规则仅仅转化已知的“多语言字符串”数据元素，因为所有其它的 XML LANG 属性跟本标准中的“多语言字符串”数据元素没有对应关系。

5.2.2.3 规则3的基本原理

此规则处理 XML 标签及其内容与数据元素之间的主要转换。

规则 3 的第 1 句：“对于在本标准中每一个与 XXX 标准所定义的数据元素标识符有关系的 XML 标签，需要对它对应的开、闭标签进行匹配”

- 忽略掉所有本标准不识别的标识符
- 正确地匹配它们。

规则 3 的第 2 句：“对于每一个 XML 标签，除了规则 2 中修改的部分，需要把该标签映射为对应的数据元素标识符”，创建与数据元素之间的联系，但并不给数据元素赋值。

规则 3 的第 3 句：“XML 标签的嵌套结构表示了数据元素的嵌套结构，如上述 5.1 中的规则 1 的逆操作即是生成和解释 XML”，在本标准所要求的范围内，确保 XML 标签的内部结构与数据元素的内部结构一致。

规则 3 的第 4 句：“每个用标签标记的元素的内容被转换成相应的数据元素的值”，将 XML 标签内的内容转换成数据元素的值，即填充数据元素。

5.3 基本数据类型的表示

5.3.1 字符和字符串

字符 (character) 类型的数据元素应该根据 XML 规范进行表示。

注1：特殊字符，如：“&”、“<”、“>”、“;”，需要转换方法，尽可能无损。

注2：某些编码，如：GB/T 15273.1—1994和GB 13000.1—1993允许诸如“©”（版权符号）等字符的直接编码表示。其它的编码，如：GB/T 1988—1998，需要进行编码扩展，如：通过“©”来表示这些符号。

5.3.2 整数

整数 (integer) 类型的数据元素应根据 GB/T 15272—1994 程序设计语言 C，第 6.4.4.1 小节（整型常量）来表示，但不包括“U”、“L”、“LL”后缀及它们的小写形式。整数也可以包含可选的前导符：加号（“+”）或减号（“-”），但不能同时有两者。

示例：

```
0
23
0x17    // 16 进制的 23
027     // 8 进制的 23
-34
+34
```

5.3.3 实数

实数 (real) 类型的数据元素分为下列两种情况：

- 如果是整数，参照 5.3.2：整数。
- 如果不是整数或不表示为整数，应根据 GB/T 15272—1994 程序设计语言 C，第 6.4.4.2 小节（浮点常量）来表示，但不包括“F”和“L”后缀以及它们的小写形式。实数也可以包括一个可选的前导符：加号（“+”）或减号（“-”），但不能同时有两者。

示例：

```
0
0.0
130.0
1.3E2    // 130 的另一种表示方法
```

+1.3E2 // 130 的另一种表示方法

5.3.4 日期和时间值

时间（time）类型的数据元素的表示应遵循 GB/T 7408-2005，数据元素和交换格式——信息交换——日期和时间的表示。

示例 1：时间点

2007-06-11 // 2007 年 6 月 11 日
2007-06-16T19:20:30+01:00 // 2007 年 6 月 16 日，下午 7 点 20 分 30 秒，与 UTC 时差为+1 小时

示例 2：时间段

PT1H30M12.88S // 1 小时 30 分 12 秒 88
P1Y1M3DT1H // 1 年 1 个月 3 天 1 小时

5.3.5 空类型

空类型（void）应该没有表示和编码。

示例：下列记录：

```
A: record
(
  B: integer,
  C: void,
  D: characterstring(GB-13000-1),
)
```

用 XML 表示为：

```
<!--正确的 XML 表示 -->
<A>
  <B>17</B>
  <D>hello</D>
</A>
```

而不应该是：

```
<!--不正确的 XML 表示 -->
<A>
  <B>17</B>
  <C></C>
  <D>hello</D>
</A>
```

5.4 字符的编码

XML 编码技术规定了如何将字符映射为 8 位字节的方法。

遵循 XML 绑定的 XXX 标准的数据实例应使用以下标准中的一种进行编码：GB/T 1988—1998、GB/T 15273.1—1994、GB 13000.1—1993 UTF-8 或 GB 13000.1—1993 UTF-16。

遵循 XML 绑定的 XXX 标准的应用应该支持以下所有编码标准：GB/T 1988—1998、GB/T 15273.1

—1994、GB 13000.1—1993 UTF-8 或 GB 13000.1—1993 UTF-16。

5.5 对异常和扩展的处理

5.5.1 实现时定义的行为

除了在本标准其他地方描述的内容之外，下列是实现时定义的行为。

下列是在 XML 编码的产生和消费中，实现时定义的行为：

- 用 XML 编码并能够被成功处理的、严格一致的 XXX 标准的数据实例的最大长度（以 8 位字节表示）。
- XML 记录的最大嵌套深度。
- 未指定时区的时间（time）类型的数据元素的时区信息。

5.5.2 未规定的行为

除了在本标准其他地方描述的内容之外，下列是未规定的行为。

下列是在 XML 编码的生成和解释中，未规定的行为：

- 数据元素的处理次序。

下列是在 XML 编码的产生和消费中，未规定的行为：

- 除在数据元素值中和 XML 规范要求的以外，附加空格的使用。

5.5.3 未定义的行为

除了在本标准其他地方描述的内容之外，下列是未定义的行为。

下列是在 XML 编码的产生和消费中，未定义的行为：

- 和扩展数据元素对应的 XML 标签的使用。
 - 和保留数据元素对应的 XML 标签的使用。
 - 在本 XML 绑定中未定义的 XML 标签或属性。
 - 对本标准所描述的字汇以外的字符的使用。
-