电力系统数据标记语言

**——E**语言规范

**Data Markup Language for Electric Power System**

**——E Language**

# 目　　录

前言 II

1 范围 1

2 符号定义 1

3 基本语法 1

4 扩展语法 4

附录A（参考性附录） E语言数据样例 8

附录B（规范性附录） 属性类型和量纲标准 10

# 前　　言

电力系统数据标记语言——E语言规范（以下简称“E语言”）是在IEC 61970–301电力系统公用数据模型CIM（Common Information Model）的面向对象抽象基础上，针对CIM在以XML方式进行描述时的效率缺陷所制定的一种新型高效的电力系统数据标记语言。

E语言的命名源于Easy（简单）、Efficiency（高效）、Electric-power（电力）3个英文词头，它将电力系统传统的面向关系的数据描述方式与面向对象的CIM相结合，既保留了面向关系方法的高效率，继承了其长期的研究成果，又吸收了面向对象方法的优点（如类的继承性等），具有简洁、高效和适用于电力系统的特点。

E语言是一种标记语言，具有标记语言的基本特点和优点，其所形成的实例数据是一种标记化的纯文本数据。E语言通过少量标记符号和描述语法，可以简洁高效地描述电力系统各种简单和复杂数据模型，数据量越大则效率越高，而且E语言比XML更符合人们使用的自然习惯，计算机处理也更简单。

E语言与XML均一致地遵循CIM基础对象类，以E语言描述的电力系统模型与以XML语言描述的电力系统模型可进行双向转换。

本标准由国家电网公司国家电力调度通信中心提出并负责解释。

本标准由国家电网公司科技部归口。

本标准起草单位：国家电网公司国家电力调度通信中心，国网电力科学研究院，江苏电力调度通信中心

本标准的主要起草人：辛耀中、林峰、陶洪铸、倪斌、彭清卿、李毅松、石俊杰、崔恒志、苏大威

# 电力系统数据标记语言——E语言规范

1　范围

本标准规定了电力系统数据标记语言——E语言的语法和语义。

E语言主要适用于以文件方式描述和交换数据的应用场合。

本标准定义的E语言版本为V1.0。

2　符号定义

|  |  |  |
| --- | --- | --- |
| 序号 | 符号 | 定 义 |
| 1 | < > | 类起始符 |
| 2 | </ > > | 类结束符 |
| 3 | <! | 系统声明起始符 |
| 4 | !> | 系统声明结束符 |
| 5 | @ | 数据块头引导符（横表式） |
| 6 | @@ | 数据块头引导符（单列式） |
| 7 | @# | 数据块头引导符（多列式） |
| 8 | # | 数据行引导符 |
| 9 | // | 注释引导符 |
| 10 | :: | 类和实体连接符 |
| 11 | = | 赋值连接符 |
| 12 | . | 名称连接符，父类与子类的连接符 |
| 13 | /@ | 属性别名行引导符 |
| 14 | % | 属性类型行引导符 |
| 15 | $ | 属性量纲行引导符 |
| 16 | : | 属性限值行引导符，限值上下限定义，父类名缩写定义 |
| 17 | \* | 指针引导符 |
| 18 | 空格 | 由一个或连续多个空格或制表符（Tab）组成 |
| 19 | ’ | 含空格的字符数据，前后加单引号 |
| 20 | - | 该项数据为空 |
| 21 | , | 对象的枚举 |
| 22 | := | 无结构数据块定义 |

说明：以上符号均为英文半角符号。

3　基本语法

E语言数据是纯文本数据。E语言主要通过对文本中每行第一个字符或前两个字符的使用，达到规范文本的目的。

E语言对于英文字母的大小写敏感。

3.1　基本结构

E语言数据文件包括注释（可选）、系统声明、数据块起始标记、数据块头定义、数据块、数据块结束标记等部分。

E语言数据有三种基本结构，即横表式结构、单列式结构和多列式结构，分别如图1、图2、图3所示：

注释

系统声明

数据块起始标记

数据块头定义

数据块

数据块结束标记

//注释

<!System=OMS Version=1.0 Code=UTF-8 Data=1.0!>

<类名::实体名>

@ 序号 属性名1 属性名2 属性名3

# 1 对象1值1 对象1值2 对象1值3

# 2 对象2值1 对象2值2 对象2值3

……

# n … …

</类名::实体名>

图1　**E**语言数据的横表式结构

注释

系统声明

数据块起始标记

数据块头定义

数据块

数据块结束标记

//注释

<!System=OMS Version=1.0 Code=UTF-8 Data=1.0!>

<类名::实体名>

@@ 序号 属性名 属性值

# 1 属性1 值1

# 2 属性2 值2

……

# n … …

</类名::实体名>

图2　**E**语言数据的单列式结构

注释

系统声明

数据块起始标记

数据块头定义

数据块

数据块结束标记

//注释

<!System=OMS Version=1.0 Code=UTF-8 Data=1.0!>

<类名::实体名>

@# 序号 属性名 对象1 对象2 …… 对象n

# 1 属性1 值11 值12 …… 值1n

# 2 属性2 值21 值22 …… 值2n

……

# n … …

</类名::实体名>

图3　**E**语言数据的多列式结构

3.2　注释

注释是E语言的可选部分，通过双斜杠“//”引导开始。注释可出现在数据文件的任意地方，既可以独立一行，也允许在行的后部。

3.3　系统声明

E语言的系统声明位于数据块前端，由左尖括号加叹号并列“<!”引导开始，由叹号加右尖括号并列 “!>”标记结束。系统声明中说明产生数据的应用系统名称、采用的E语言版本、字符集和数据版本。标记内的具体内容如下：

a） System=应用系统名称。指产生本数据的应用系统名称。

b） Version=E语言版本。指封装本数据所采用的E语言版本。由于E语言本身的发展，语法规则和符号可能会发生变化，此版本信息可用于程序识别语言本身的差别。

c） Code=字符集名称。指本文件所用的字符集。

d） Data=数据版本。指本数据所处版本。由于应用需求变化，同样类别的数据其封装结构可能会进行局部的扩充、修改或简化，此版本信息可用于程序识别数据本身的细微差别。

以上各部分之间用空格分割，并且可根据系统情况进行扩展。以下为一个系统声明的实际样例：

<! System=OMS Version=1.0 Code=UTF-8 Data=1.0 !>

3.4　数据块起始和结束

E语言数据块的起始用尖括号“<类名>”或“<类名::实体名>”表示，数据块结束用尖括号内加单斜杠“</类名>”或“</类名::实体名>”表示。

“类名::实体名”中，前者表示数据块内数据对象的类，后者表示数据块内数据的归属对象。如数据块起始符“<调度日报::华东>”指明了数据块内的数据是调度日报类，并且这些数据归属于华东电网，即这些数据为华东电网的调度日报数据。

类和实体名称之后可跟若干个标记属性名及其值，标记属性名与属性值之间用等号“=”相连，多个属性名值之间用空格分割。如Date=’YYYY-MM-DD’，表示报表数据的日期。

数据块中如果只包含一个对象也可以用一行来描述，采用如下格式：

<类名::实体名　属性1=值1　属性2=值2　/>。

3.5　数据块头定义

数据块头定义用地址符“@”标记，用于表明数据的基本结构：即横表式、单列式或多列式。数据块头定义的具体标记如下：

a） 单地址符“@”表示数据的基本结构为横表式，其语法为：

@ 序号 属性名1 属性名2 属性名3 …

“@”、序号、各属性名之间使用空格分割，序号也可用数据ID表示（下同），单列式和多列式结构对于序号的处理方式类似。横表式结构的数据块中每个对象占一行、每个属性占一列，适用于表格类数据，或属性较少且对象较多的数据，如图1所示。

b） 双地址符“@@”表示数据的基本结构为单列式，其语法为：

@@ 序号 属性名 属性值

“@@”、序号、属性名、各属性值之间通过空格分割。单列式结构的数据块中每个属性占一行，属性名和值各占一列，适合于单个对象且属性较多的数据，如图2所示。

c） 单地址符和井号“@#”表示数据的基本结构为多列式，其语法为：

@# 序号 属性名 对象名1 对象名2 对象名3 …

“@#”、序号、属性名、各对象名之间通过空格分割。多列式结构的数据块中每个属性占一行，每个对象占一列，适合于对象数相对固定且属性较多的数据，如图3所示。

3.6　数据块

数据块由多个数据行组成，每个数据行由井号“#”引导。数据行中的各数据值之间通过空格分割。数据行的表达方式与数据块头定义一一对应。

数据块的第一列总是在“#”后跟一个或多个空格，空格后加序号，如“# 10”，使序号单独成为一列。该序号表示本行数据在本数据块中的顺序，以1开始编号。序号后面跟一空格后开始数据值序列。

3.7　空格

E语言中使用空格作为分割符，如数据块中每行各项内容之间使用空格分割。空格分割符由一个或连续多个空格或制表符（Tab）组成。如果字符串数据中含有空格字符，则需在字符串数据前后加单引号“’”。

3.8　连接符

a） 类与实体连接符：用“::”表示，用于类和实体之间的连接，如：<调度日报::华东>。

b） 赋值连接符：用等号“=”表示，用于数据块起始标记内属性名与属性值之间的连接，如：<调度日报::华东 日期=‘2006-04-02’ 时间=‘23:15:00’>。

c） 名称连接符：用小数点“.”表示，用于连接层次结构的类或实体。用于实体的层次描述如：“华中.河南.郑州”；用于类及属性的层次描述如：“线路.阻抗”等。

4　扩展语法

为保持E语言的完整性和灵活性，定义下列扩展语法。

4.1　属性类型定义

在数据块头定义中，如果是横表式结构数据，可以为每个属性增加类型定义，此定义为可选。

类型定义行用百分号“%”引导，后面跟一空格及类型序列，与横表式结构属性序列一一对应，指明各属性的数据类型，数据类型长度不做限制。规定：i（int）表示整数类型，f（float）表示浮点数类型，s（string）表示字符串类型，p（pointer）表示指针类型，如图4所示。属性类型遵循附录B的规定。

4.2　属性量纲定义

在数据块头定义中，如果是横表式结构数据，可以为每个属性增加量纲定义，此定义为可选。

量纲定义行用美元符“$”引导，后面跟一空格及量纲序列，与横表式结构属性序列一一对应，指明各属性的量纲，例如：MW，kV等，若某属性没有量纲时，用“-”表示，如图4所示。量纲标准遵循附录B的规定。

4.3　属性限值定义

在数据块头定义中，如果是横表式结构的数据，可以为每个属性增加限值定义，此定义为可选。

限值定义行用冒号“：”引导，后面跟一空格及限值序列，与横表式结构属性序列一一对应，指明各属性的限值。限值用“最小允许值:最大允许值”表示，例如：“月”的限值为“1:12”。如果没有最小限值限制，左侧位置不填，如“:100”；如果没有最大限值，右侧位置不填，如“10:”；如果上下均无限值则用“-”表示，如图4所示。

//数据类型、量纲和限值描述

<Line >

@ 序号 I\_Node J\_Node R X Cx

**% i s s f f f**

**$ - - - Ω Ω Ω**

**: 1:10 - - - - -**

# 1 辛安 洹安 0.0194 0.0592 0.0264

# 2 姜家营 高岭 0.0470 0.1980 0.0219

……

# n … …

</Line>

图4　类型、量纲和限值定义

4.4　对象指针

E语言中用星号“\*”表示对象指针。当“\*”出现在属性名前面时，表示该属性为指针属性（即其值是指向对象的指针），后面紧跟指针所指向的类名；当“\*”出现在数据项前面时，后面紧跟指针所指向的对象序号（关于序号参见3.6数据块）。如果指针指向多个对象，对连续的对象序号以冒号连接起始和终止对象序号，当对象标识不连续时以逗号“，”分割各对象标识（最后一个对象标识无逗号）。

如图5中，阳城变电站的\*Line属性的值为“\*45:48,67”，表示指针指向线路类Line的第45号到第48号对象以及第67号对象。如果某数据行不存在指向该类的指针，则对应数据项填0，且前面不带“\*”号，如图5中由于三堡变电站没有发电机，相应数据项为0。

对于横表式结构，如果某指针属性只有少量数据行存在对应值（对象实例），则该属性称为“稀疏指针属性”。稀疏指针属性可以不在数据块头中定义，而是在具有稀疏指针属性值的数据行后另起一“指针行”以单独存放数据。指针行以“\*”开头，后跟指向的类名、双冒号“::”及所指对象序号。指针行位于所属对象数据行之后,在下一个对象数据行之前。在两个对象数据行之间允许连续出现多个指针行。指针行可有效避免由于稀疏指针属性引起的存储浪费，提高存储效率，如图5所示。

//指针使用举例

<Substation>

@ 序号 Name Voltage Angle P Q Type **\*Gen \*Line**

# 1 阳城 1.0600 0.0000 232.38 -16.8900 VA **\*25:31 \*45:48,67**

# 2 三堡 1.0200 0.0002 231.49 -15.0025 PQ **0** **\*49:54**

**\*串补装置::2,4**

**\*电容器::9:11,21**

# 3 东明 1.0200 0.0002 231.49 -15.0025 PQ 0 0

……

# n … …

</Substation>

图5　**E**语言对象指针

4.5　属性继承

可以在数据块起始标记和数据块头定义中使用属性继承。

可以在数据块起始标记的尖括号内列出若干个有继承关系的父类名，父类之间用空格分割，每个父类名后跟冒号“：”，冒号后跟一个单字母化的父类名缩写。

数据块头定义中可以引用上述定义，以继承父类的属性，该属性名前用父类名缩写加小数点“.”予以限定。例如在线路类Line的属性中，继承了区域类的某些属性I和J，如图6所示。这里仅继承了所用的属性，不是所有属性。

//继承机制使用举例

<Line **Area:A**>

@ 序号 I\_Node J\_Node R X Cx **A.I** **A.J**

# 1 辛安 洹安 0.0194 0.0592 0.0264 华北 华中

# 2 姜家营 高岭 0.0470 0.1980 0.0219 华北 东北

……

# n ……

</Line>

图6　**E**语言继承机制使用举例

4.6　无结构数据描述

用冒号和等号“:=”表示，如图7所示。类或实体引导符尖括号内的<类名::实体名>退化为<数据块:=Free>，省略掉以“@”开头的属性描述行和以“#”开头的数据行，以及所有结构化描述机制，结束符</类名::实体名>退化为</数据块:=Free>。可描述任何数据，包括图形、图像、文字、程序等。

//无结构数据描述方式

<数据块:=Free SIZE=数据块长度>

数据块（任意数据）

</数据块:=Free >

图7　无结构数据描述机制

4.7　嵌套描述方式

嵌套是指在一类数据块内又包括其它类数据的描述方式，这种方式与XML非常相似。为了保持E语言的高效率，位于行首的特殊字符不变，后面可用空格“ ”和跳格“TAB”表示嵌套格式，如图8所示。该方式逻辑性较好，但效率较低，简洁性差。

//嵌套描述方式

<Substation>

@ 序号 Name Voltage Angle P Q Type

# 1 阳城 1.0600 0.0000 232.38 -16.8900 VA

**<Gen>**

**@ 序号 Name V A Pg Qg**

**# 1 G1 1.0600 0.0010 232.38 -16.8900**

**</Gen>**

# 2 三堡 1.0200 0.0002 231.49 -15.0025 PQ

</Substation>

图8　嵌套描述方式

4.8　模板

E语言不仅可以描述数据内容，也可以用来对数据格式进行定义，即作为模板使用。

E语言的模板功能类似于XML语言中的大纲（Schema）功能，主要用于描述数据类包含的属性列表及每个属性的名称、类型、长度、量纲和备注等相关信息。

E语言的模板功能一方面可以用于对数据内容进行格式定义，实现数据格式自描述，便于计算机编程实现；另一方面也便于今后属性的扩展。

//模板

<线路::数据格式定义>

@ 序号 属性名称 类型 长度 量纲 备注

# 1 线路名称 s 64 无 设备的调度命名

# 2 线路长度 i - km

……

# n … …

</线路::数据格式定义>

图9　模板使用举例

4.9　属性别名

为了适应某些属性名称的中英文双重命名或者多个系统属性命名不一致的情况，属性可以使用别名机制。属性别名主要适用于横表式结构，在其数据块头的标记单地址符@前增加反斜杠“/”，以“/@”标示，如图10所示。

//华东厂站信息

<! System=OMS Version=1.0 Code=UTF-8 Data=1.0 !>

<厂站::华东>

@ 序号 名称 电压 类型 主接线

**/@ ID name voltage type mainbus**

# 1 石洞口二厂 500 火电厂 双母带旁路

# 2 清流变 500 变电站 双母带旁路

</厂站::华东>

图10　属性别名使用举例

4.10　计算公式

E语言通过“FRAME”机制支持类似于Excel的公式计算。将整个E语言文本装载到FRAME中，其中每行用“行号”（阿拉伯数字1，2，…）标识，每个属性及对应的值用“列名”（大写英文字母A，B，C，…）标识。数据块中的数据项可以为公式的计算结果，用等号“=”引导，后跟数据项名及四则运算符号，必要时可用括号“（）”。例如图11中，“=H5×E5”表示该数据项等于第5行属性H的值与第5行属性E的值之积。公式描述的语法基本等同于Excel。

行号 A B C D E F G H I

1 < Line >

2 @ ID I\_Node J\_Node R X Cx Pi Qi

3 # 1 阳城 东明 0.02 0.06 0.002 793.67 34.30

4 # 2 东明 三堡 0.03 0.09 0.001 820.43 16.30

5 # 3 绥中 姜家营 0.01 0.03 0.002 384.12 **=H5**×**E5**

6 # 4 辛安 洹安 0.02 0.06 0.001 407.67 18.30

7 </ Line >

图11　**E**语言的**FRAME**及计算公式描述

附　录　**A**  
**（**参考性附录**）**  
**E**语言数据样例

**A**.1　横表式结构

//华东厂站信息

<! System=OMS Version=1.0 Code=UTF-8 Data=1.0 !>

<厂站::华东>

@ 序号 名称 电压 类型 主接线

# 1 石洞口二厂 500 火电厂 双母带旁路

# 2 清流变 500 变电站 双母带旁路

……

# n … …

</厂站::华东>

**A**.2　单列式结构

//华东调度日报

<! System=OMS Version=1.0 Code=UTF-8 Data=1.0 !>

<调度日报::华东 Date=’2006-09-02’>

@@ 序号 数据名称 数据值

# 1 统调发电量 1856198

# 2 统调火电量 1744342

# 3 统调水电量 46119

……

# n … …

</调度日报::华东>

**A**.3　多列式结构

//华东年度电力电量平衡预测表

<! System=OMS Version=1.0 Code=UTF-8 Data=1.0 !>

<年度电力电量平衡预测表::华东 Date=’2007-01-01’>

@# 序号 属性名 1月 2月 3月 …… 12月

# 1 新增统调装机容量 237 224 190 …… 205

# 2 受电量 2436 2234 1881 …… 2132

……

# n 用电量需求 9012 8087 8767 …… 8120

</年度电力电量平衡预测表::华东>

附　录　**B**  
**（**规范性附录**）**  
属性类型和量纲标准

**B**.1　属性类型标准

|  |  |  |
| --- | --- | --- |
| 序号 | 属性类型缩写及全称 | 属性类型说明 |
| 1 | i（integer） | 整数 |
| 2 | f（float） | 浮点数 |
| 3 | s（string） | 字符串 |
| 4 | p（pointer） | 对象指针 |

**B**.2　量纲标准

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 量纲名称 | 单位名称 | 单位符号 |
| 1 | 频率 | 赫兹 | Hz |
| 2 | 电压 | 千伏 | kV |
| 3 | 电流 | 安培 | A |
| 4 | 有功功率 | 兆瓦 | MW |
| 5 | 无功功率 | 兆乏 | MVar |
| 6 | 电量 | 兆瓦时 | MWh |
| 7 | 容量 | 兆伏安 | MVA |
| 8 | 阻抗 | 欧姆 |  |
| 9 | 长度 | 千米 | km |

# 电力系统数据标记语言——E语言规范 编 制 说 明

**1.** 编写背景

电力系统数据标记语言――E语言规范（以下简称“E语言”）是在IEC 61970-301电力系统公用数据模型（CIM――Common Information Model）的面向对象抽象基础上，针对CIM在以XML方式进行描述时的效率缺陷所发展出来的一种新型高效的电力系统数据建模语言。

E语言的命名源于Easy（简单）、Efficiency（高效）、Electric-power（电力）3个英文词头，它将电力系统传统的面向关系（设备）的数据描述方式与面向对象的CIM相结合，既保留了面向设备方法的高效率，继承了其长期的研究成果，又吸收了面向对象方法的优点（如类的继承性等），具有简洁、高效和实用于电力系统的特点。

E语言是一种标记语言，具有标记语言的基本特点和优点，其所形成的实例数据是一种标记化的纯文本数据。E语言通过少量的几个标记符号和描述语法，就可以简洁高效地描述电力系统各种简单和复杂数据模型。对于大量数据的描述，E语言效率比XML高得多，而且数据量越大则效率越高。对于少量数据的描述，E语言比XML效率稍高，但E语言更符合人类的自然习惯，计算机处理也更简单。

**2.** 编制情况

E语言规范的研究始于2005年4月调度系统数据整合工作，作为《国家电网公司调度系统数据整合总体框架》的附件《调度系统报表数据交换格式》发各网、省调征求意见，同年该研究成果就应用于调度日报表的交换，实现了国、网、省三级调度机构调度日报的自动交换汇总，成为E语言规范的雏形；随后E语言规范在调度机构部分网、省调电网运行数据交换工作中得到进一步充实，补充了对象指针、属性继承、嵌套描述等内容，并于2006年5月以《电力系统数据模型描述语言E》发表于《电力系统自动化》杂志。

为在电力系统更好地推广E语言使用，2006年10月，国家电力调度通信中心成立E语言规范研究工作小组，结合近两年来E语言在调度系统中的实际应用情况，对E语言内容进行了必要的修改和扩充，并召集部分网省公司专家进行了专题讨论和修改。2007年3月16日形成《电力系统数据标记语言――E语言规范》（征求意见稿），并以国调“调自[2007]54号文”方式下发至公司系统各调度机构征求意见。其后，工作小组根据反馈意见对规范进行多次修改，于2007年10月形成送审稿。2008年1月11日，国调中心在京组织召开了由有关专家参加的评审会议，对本标准送审稿进行了评审并根据评审意见对本规范进行了修改，形成报批稿。

**3.** 编制的主要依据

IEC 61970-301电力系统公用数据模型。

**4.** 主要章节及有关说明

前言部分简要说明了E语言规范的设计背景、命名原因、基本特点和优点，强调了E语言在设计时所遵循的基本原则。

（1）目的

说明设计E语言规范的目的是简化标记，减少冗余，提高效率。

（2）适用范围

E语言规范主要适用于描述电力系统模型和电力系统应用系统间数据交换两大方面。

（3）基本语法部分

提出E语言数据的三种基本结构：即横表式结构、单列式结构和多列式结构，对注释、系统声明、数据块起始和结束、数据块头定义、数据块、空格的处理方式、连接符等内容进行了规范。

（4）扩展语法部分

对E语言横表式结构的属性类型定义、属性量纲定义、属性限值定义进行了规范，对E语言对象指针的用法、属性继承的定义、无结构数据描述、嵌套描述方式、模板的使用、属性别名的定义、计算公式的定义进行了规范。

（5）附录A

分别提供了横表式、单列式、多列式三种基本结构的E语言数据样例。

（6）附录B

对属性类型标准进行了说明，主要包括整数、浮点数、字符串、时间、对象指针等类型；

对量纲标准进行了说明，主要包括频率、负荷及电力、电量、电压、容量、阻抗、长度等。