逗号分隔值

维基百科，自由的百科全书

逗号分隔值（Comma-Separated Values，**CSV**，有时也称为**字符分隔值**，因为分隔字符也可以不是逗号），其文件以纯文本形式存储表格数据（数字和文本）。纯文本意味着该文件是一个字符序列，不含必须像二进制数字那样被解读的数据。CSV文件由任意数目的记录组成，记录间以某种换行符分隔；每条记录由字段组成，字段间的分隔符是其它字符或字符串，最常见的是逗号或制表符。通常，所有记录都有完全相同的字段序列。

CSV文件格式的通用标准并不存在，但是在RFC 4180中有基础性的描述。使用的字符编码同样没有被指定，但是7-bit ASCII是最基本的通用编码。

逗号分隔值
逗号分隔列

扩展名	`.csv`或`.txt`
互联网媒体类型	`text/csv`
格式类型	跨平台，串行数据流
專門屬	数据库按字段分隔的列表的信息组织
标准	RFC 4180

目录

用法

CSV是一种通用的、相对简单的文件格式，被用户、商业和科学广泛应用。最广泛的应用是在程序之间轉移表格数据，而这些程序本身是在不兼容的格式上进行操作的（往往是私有的和/或无规范的格式）。因为大量程序都支持某种CSV变体，至少是作为一种可选择的输入/输出格式。

例如，一个用户可能需要交换信息，从一个以私有格式存储数据的数据库程序，到一个数据格式完全不同的电子表格。最可能的情况是，该数据库程序可以导出数据为“CSV”，然后被导出的CSV文件可以被电子表格程序导入。

“CSV”并不是一种单一的、定义明确的格式（尽管RFC 4180有一个被通常使用的定义）。因此在实践中，术语“CSV”泛指具有以下特征的任何文件：

1. 纯文本，使用某个字符集，比如ASCII、Unicode、EBCDIC或GB2312（简体中文环境）等；
2. 由记录组成（典型的是每行一条记录）；
3. 每条记录被分隔符分隔为字段（典型分隔符有逗号、分号或制表符；有时分隔符可以包括可选的空格）；
4. 每条记录都有同样的字段序列。

在这些常规的约束条件下，存在着许多CSV变体，故CSV文件并不完全互通。然而，这些变异非常小，并且有许多应用程序允许用户预览文件（这是可行的，因为它是纯文本），然后指定分隔符、转义规则等。如果一个特定CSV文件的变异过大，超出了特定接收程序的支持范围，那么可行的做法往往是人工检查并编辑文件，或通过简单的程序来修复问题。因此在实践中，CSV文件还是非常方便的。

历史

逗号分隔值是一种数据格式，列表方式（“自由形式”）输入/输出被定义在FORTRAN 77（77代表1977年）中。列表方式的输入使用了逗号和/或空格作为分隔符，所以用来结束引文的字符串不能包含逗号或空格。[1]

相对于固定列宽数据格式，逗号分隔值的列表不仅输入（例如输入到打孔卡）更加方便，而且当一个值被错打一列时也不容易产生错误结果。

逗号分隔列（CSL）是一种数据格式，起初在最古老的简单电脑中被称为**逗号分隔值**（CSV）。在个人电脑（当时更普遍地被称为“家用电脑”）产业，一个常见的应用是，小企业使用模板和邮件列表生成推销邮件。

CSL/CSV被用来作为简单的数据库。一些早期的软件应用，比如文字处理器，允许一系列“变量数据”在两个文件之间被合并：一个是模板文件，一个是包含姓名、地址和其它数据字段的CSL数据库。许多应用程序仍然有这种能力。[2]

逗号分隔列过去和现在都被用于在两个不同架构的机器之间交换数据库信息。纯文本的CSV文件大幅避免了不兼容性，比如字节顺序和字长。这些文件大部分是可读的，所以在没有完美的文档或通讯的情况下仍然很容易处理。

常规功能

CSV格式最好被用来表现记录集合或序列，其中的每条记录都有完全相同的字段序列。这相当于关系数据库中一个单一的关系，或者典型的电子表格中的数据（虽然不能计算）。

CSV格式没有被限定于某个特定字符集。不管用Unicode还是用ASCII，都没有问题（尽管特定程序支持的CSV可能会有它们自己的局限性）。甚至从一个字符集翻译到另一个字符集，CSV文件都不会有问题（不象几乎所有的私有数据格式）。然而，CSV不提供任何途径来表明使用的是什么字符集，所以必须另外通讯，或在接收结束时指出（如果可能）。

如前所述，包含多个关系的数据库不能导出为单一的CSV文件。最多，只能添加更多的标识约定，例如标识和分隔不同的关系。这种标识并不难于设计和实现，但是因为没有这方面的约定，所以基本不具备可移植性。

同样地，CSV本身不能表达分层级的或面向对象的数据库或其它数据。这是因为每一条CSV记录都应当有同样的结构。CSV因此基本不适用于文档，比如由HTML、XML或者其它的标记语言或文字处理技术创建的。

字段会变化的统计数据库往往有着类似关系的结构，但是会有一些字段组是可重复的。例如，健康数据库如人口与健康调查，对于一名给定的父母会为每个孩子重复一些问题（也许直到某个确定的最大的孩子数目）。统计分析系统往往包含可以“旋转”这类数据的工具：例如，一条包含5个孩子信息的“父母”记录，可以被分解为5条单独的记录，每条记录包含（a）一个孩子的信息，和（b）一份所有未指定孩子的信息的拷贝。CSV可以用“横的”或“竖的”形式来表达这类数据。

在关系数据库中，类似的问题很容易解决——为每个类似的组另外创建一个关系，并使用外键（如父母的身份证号或名字）将“孩子”记录与相关的“父母”记录连接起来。在标记语言如XML中，这类组会被包含在一个容器中（例如，`<child>`），然后按照需要重复该容器。对于CSV，则尚无被广泛接受的单文件解决方案。

缺乏规范

“CSV”这个名字表示是用逗号来分隔数据字段的。然而，“CSV”这个术语还被广泛用于形式各异的一系列格式。例如，许多所谓的“CSV”文件实际上使用制表符代替逗号（这种文件可以更精确地被称为“TSV”，即制表符分隔值，Tab-Separated Values）；一些实现允许或要求使用单引号或双引号来包裹某些或全部字段；还有一些实现保留最前面的一条记录作为表头来包含字段名序列。

其它的实现差别包括如何处理更多常见的字段分隔符（比如空格或分号[3]）和在文本字段中的换行符。[4]更加微妙的是对一个空行的翻译：写一条零字段的记录，或者一条有一个零长度字段的记录同样可以得到这样的结果；因此对它的解码是有歧义的——只不过这种歧义通常无伤大雅。

RFC 4180所记载的标准可以使CSV交换简单化。然而，这一标准仅指定了对基于文本的字段的处理。对各个字段的文本的翻译仍然是由引用程序指定的。特别地，并没有标准来指定如何表示小数，尽管它们普遍被嵌入在CSV数据中，并且一些国家使用点作为小数分隔符，一些国家使用逗号。例如，法文CSV文件可能将圆周率的近似值写为3,14159。

一个通用的（如果技术上不令人满意）互操作性解决方案是依赖人工介入：因为CSV文件是纯文本，用户可以通过使用文本编辑器轻易预览和诊断绝大部分（如果不是全部）问题。这样做的例子包括 Microsoft Access 等（关系）数据库产品，其在导入 CSV/TSV 这种数据时便允许用户一边预览一边调整诸如表头行、分隔符之类的选项。

进行中的标准化

“CSV”格式中大量变体的存在说明并没有一个“CSV标准”。[5][6]在常见用法中，几乎任何定界符分隔的文本数据都可以被统称为“CSV”文件。不同的CSV格式可能不会兼容。

不过，RFC 4180是一个将CSV正式化的努力。它定义了互联网媒体类型“text/csv”，并且采用它的规则的CSV文件将会具有广泛的可移植性。它有如下要求：

- 以（CR/LF）字符结束的DOS风格的行（最后一行可选）。
- 一条可选的表头记录（没有可靠的方式来检测它是否存在，所以导入时必须谨慎）。
- 每条记录“应当”包含同样数量的逗号分隔字段。
- 任何字段都*可以*被包裹（用双引号）。
- 包含换行符、双引号和/或逗号的字段*应当*被包裹。（否则，文件很可能不能被正确处理）。
- 字段中的一个（双）引号字符必须被表示为两个（双）引号字符。

这个格式很简单，可以被大部分声称可以读取CSV文件的程序处理。例外是（a）程序可以不支持在被包裹的字段中换行，（b）程序可以将可选的表头当作数据，或者将第一个数据行当作可选的表头。

技术背景

该格式可以追溯至早期的商用电脑，它被广泛用于在电脑间传递数据，这些电脑可能会有不同的内在字长，需要不同的数据格式，等等。为此，CSV文件在所有电脑平台上通用。

CSV是一种分隔的文本文件，它使用逗号来分割值（许多CSV导入/导出工具的实现也允许使用其它的分隔符）。简单的CSV实现可以禁止字段值中包含逗号或其它特殊字符如换行符。更复杂的CSV实现允许这些特殊字符，它们往往要求用"（双引号）包裹这些包含保留字符（如逗号、双引号或不太通用的换行符）的数值。被嵌入的双引号字符可以用连续两个双引号来表示（Creativyst 2010），或者使用转义字符如反斜杠（例如在Sybase Central中）。

在计算机科学术语中，CSV文件可以被认为是一个“平面文件数据库”。

基本规则及举例

存在许多描述“CSV”格式的非正式文件。IETF RFC 4180（如上所述）定义了格式“text/csv”互联网媒体类型，并已注册于IANA。（Shafranovich 2005）另一个相关规范由含字段文本提供。Creativyst（2010）提供了一个概述，说明了在最广泛使用的应用程序中所使用的变体，并解释了CSV怎样才能最好地被使用和支持。

这些和其它“CSV”规范及实现的典型规则如下：

- **CSV**是一种被分隔的数据格式，它有被逗号字符分隔的字段/列和以换行结束的记录/行。
- CSV文件不要求特定的字符编码、字节序或行结束符格式（某些软件不支持所有行结束变体）。
- 一条记录结束于行结束符。然而，行结束符可能被作为数据嵌入到字段中，所以软件必须识别被包裹的行结束符（见下述），以便从可能的多行中正确组装一条完整的记录。
- 所有记录应当有相同数目、相同顺序的字段。
- 字段中的数据被翻译为一系列字符，而不是一系列比特或字节（见RFC 2046，section 4.1）。例如，数值量65535可以被表现为5个ASCII字符“65535”（或其它形式如“0xFFFF”、“000065535.000E+00”等等）；但不会被作为单个二进制整数的2字节序列（而非两个字符）来处理。如果不遵循这个“纯文本”的惯例，那么该CSV文件就不能包含足够的信息来正确地翻译它，该CSV文件将不大可能在不同的电脑架构间正确传递，并且将不能与*text/csv* MIME类型保持一致。
- 相邻字段必须被单个逗号分隔开。然而，“CSV”格式在分隔字符的选择上变化很大。特别是在某些区域设置中逗号被用作小数点，则会使用分号、制表符或其它字符来代替。

```
1997,Ford,E350
```

- 任何字段都*可以*被*包裹*（使用双引号字符）。某些字段*必须*被包裹，详见后续规则。

```
"1997","Ford","E350"
```

- 如果字段包含被嵌入的逗号，必须被包裹。

```
1997,Ford,E350,"Super, luxurious truck"
```

- 每个被嵌入的双引号字符必须被表示为两个双引号字符。

```
1997,Ford,E350,"Super, ""luxurious"" truck"
```

- 如果字段包含被嵌入的换行，必须被包裹（然而，许多简单的CSV实现不支持字段内换行）。

```
1997,Ford,E350,"Go get one now
they are going fast"
```

- 在某些CSV实现中，起头和结尾的空格和制表符被截掉。*这一实践是有争议的，也不符合RFC 4180。RFC 4180声明“空格被看作字段的一部分，不应当被忽略。”*。

```
1997, Ford, E350
not same as
1997,Ford,E350
```

- 然而，该RFC并没有说当空白字符出现在被包裹的值之外该如何处理。

```
1997, "Ford" ,E350
```

- 在截掉起头和结尾空格的CSV实现中，将这种空格视为有意义数据的字段必须被包裹。

```
1997,Ford,E350," Super luxurious truck "
```

- 第一条记录可以是“表头”，它在每个字段中包含列名（并没有可靠途径来告知一个文件是否这样包含表头；然而，一般在列名中仅使用字母、数字和下划线，而不使用其它字符）。

```
Year,Make,Model
1997,Ford,E350
2000,Mercury,Cougar
```

举例

年份	品牌	型号	描述	价格
1997	Ford	E350	ac, abs, moon	3000.00
1999	Chevy	Venture "Extended Edition"		4900.00
1999	Chevy	Venture "Extended Edition, Very Large"		5000.00
1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799.00

以上数据表可以以CSV格式表示如下：

```
Year,Make,Model,Description,Price
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,"Venture ""Extended Edition""","",4900.00
1999,Chevy,"Venture ""Extended Edition, Very Large""",,5000.00
1996,Jeep,Grand Cherokee,"MUST SELL!
air, moon roof, loaded",4799.00
```

美国/英国的CSV文件（小数点是点，值分隔符是逗号）举例：

```
Year,Make,Model,Length
1997,Ford,E350,2.34
2000,Mercury,Cougar,2.38
```

德国和荷兰同样的CSV/DSV文件（小数点是逗号，值分隔符是分号）举例：

```
Year;Make;Model;Length
1997;Ford;E350;2,34
2000;Mercury;Cougar;2,38
```

后者的格式是不遵循RFC 4180的。若要使它遵循，可以使用逗号代替分号作为分隔符，或者使用小数点表示法的国际符号，或者包裹所有包含小数点的数字。

应用程序支持

CSV文件格式非常简单，被几乎所有的电子表格和数据库管理系统支持。许多编程语言都有可利用的库来支持CSV文件。许多实现都支持变换字段分隔字符和一些包裹约定以最大化接收者处理数据的机会，尽管使用最简单的约定是最安全的。

Microsoft Excel会打开.csv文件，但依赖于系统的地区设置，它可以使用分号作为分隔符来代替逗号，这是由于在某些语言中逗号被用作小数点。并且，Excel的许多地区版本不能在CSV中处理Unicode。当遇到这种困难时，一个简单的解决方案是将文件扩展名从`.csv`变为`.txt`；然后从一个已经运行的Excel中用“打开”命令打开文件。

当粘贴文本数据到Excel中时，通常制表符被用作分隔符：如果你复制“hello<tab>goodbye”到剪贴板中并把它粘贴到Excel中，它会进入到两个单元格中。“hello,goodbye”粘贴进Excel会进入一个单元格中，包括逗号。如果您在Excel中使用“文本到列”功能并且改变设置，它同样为粘贴进文本数据改变设置。

OpenOffice.org Calc和LibreOffice Calc处理CSV文件，并且通过一个*文本导入*对话框粘贴文本，对话框要求用户自己指定分隔符、编码、列格式等。

在类Unix系统上，有许多工具程序至少可以处理某些CSV文件。许多此类工具有办法改变分隔字符，但是很少能支持任何其它变体（或Unicode）。这些可用的程序有：

- column
- cut
- paste
- join
- sort
- uniq（-f来跳过比较前N个字段）

参见

- 数据序列化格式比较
- CSV应用程序支持
- 定界符分隔值
- 含字段文本
- 制表符分隔值

註釋

1. 列表方式的I/O, Fortran 77语言参考资料（英文）, Oracle
2. 例如Microsoft Word。
3. 例如LibreOffice 3.4的.csv设置导入。
4. 例如，这一错误 (https://www.libreoffice.org/bugzilla/show_bug.cgi?id=40680)（英文）实际上记载了OpenOffice和LibreOffice在处理文本字段内换行符的不同。
5. CSV文件的读写（英文）. [2011年7月24日]. “没有“CSV标准””
6. Y. Shafranovich. 逗号分隔值（CSV）文件的通用格式和MIME类型（英文）. [2011年9月12日].

参考资料

- Creativyst, 如何：逗号分隔值（CSV）文件格式（英文）, creativyst.com, 2010 [2010年5月24日]
- Shafranovich, Y., 逗号分隔值（CSV）文件的常用格式与MIME类型（英文）, 因特网社会, 2005年10月, RFC 4180

外部链接

- B2B（商业对商业）应用的CSV-1203文件格式规范 (http://arquivo.pt/wayback/20160516100434/http://www.mastpoint.com/csv-1203)（英文）
- 如何：逗号分隔值（CSV）文件格式 (http://www.creativyst.com/Doc/Articles/CSV/CSV01.htm)（英文）
- RFC 4180：CSV文件格式的RFC规范（英文）
