**GFZRNX 配置及格式相关功能的使用**

[2016-10-30](http://www.gnss.help/2016/10/30/gfzrnx-config-translate/)

[GFZRNX](http://www.gnss.help/categories/GFZRNX/)

[GFZRNX](http://www.gnss.help/tags/GFZRNX/), [RINEX格式转换](http://www.gnss.help/tags/RINEX%E6%A0%BC%E5%BC%8F%E8%BD%AC%E6%8D%A2/)

提到 GNSS 领域的数据预处理工具，你首先想到的肯定是 [TEQC](https://www.unavco.org/software/data-processing/teqc/teqc.html) 程序。该程序因其丰富的数据转换、编辑和质量检查等功能而被人们所熟知。但该程序也并非尽善尽美。比方说，截至目前，该程序在 RINEX 各版本之间（尤其是 RINEX 2 与 3 之间）的格式转换功能尚弱。

本文将介绍另一个 GNSS 数据预处理程序：GFZRNX。该程序也具有丰富的功能，可以帮助你轻松跨越 RINEX 2 与 3 版本格式之间的鸿沟。如果配合 TEQC 程序双剑合璧，更能让你在数据预处理工作方面游刃有余。

**程序简介**

[GFZRNX](http://semisys.gfz-potsdam.de/semisys/scripts/download/) 程序是由德国波兹坦地学研究中心（GFZ）的开发的一款用于 GNSS 数据预处理（主要适用于 RINEX 2 与 3 版本格式）的程序，支持 Windows、Linux、macOS 等常见的操作系统。此程序并不开源，但对于非商业用途的用户，提供免费的使用授权。在程序的[介绍页面](http://semisys.gfz-potsdam.de/semisys/scripts/download/" \t "_blank)，点击 “GFZ software” 标签即可看到程序的下载链接。

GFZRNX 程序支持对 RINEX 格式的观测数据文件（Obs）、广播星历文件（Nav）和气象数据文件（Met）的操作。目前提供的功能有：

* RINEX 格式检查与修复；
* RINEX 版本格式转换；
* RINEX 文件分割与拼接；
* RINEX 文件头信息编辑与导出；
* RINEX 数据编辑，如采样抽取、观测卫星筛选等；
* RINEX 文件差异比较；
* RINEX 数据质量分析。

如果你在数据处理时使用了此程序，请在论文后面为它添加如下的一个引用：

Nischan, Thomas (2016): GFZRNX - RINEX GNSS Data Conversion and Manipulation Toolbox. GFZ Data Services. <http://dx.doi.org/10.5880/GFZ.1.1.2016.002>

**环境配置**

**Windows 操作系统**

对于 Windows 操作系统，下载对应的可执行文件后，打开 “命令提示符” 窗口，使用 cd 命令进入程序所在目录，然后键入程序名（gfzrnx\_win32.exe 或 gfzrnx\_win64.exe）即可运行程序。

我可不想每次使用该程序前切换工作目录，也不喜欢命令后面诸如 “win32” 或 “win64” 这样的后缀。因此我将其重命名为 “gfzrnx.exe”，然后移动至 “C:\Windows\System32” 文件夹内。这样每次使用该程序时，只需要在 “命令提示符” 窗口键入 gfzrnx 就行了。当然，你也可以将该程序所在目录添加到系统的 Path 变量。

**Linux 或 macOS 操作系统**

对于 Linux 或 macOS 操作系统，下载对应的可执行文件后，首先将其重命名为 “gfzrnx” 以去除冗长的后缀，然后为程序分配可执行权限：

|  |  |
| --- | --- |
| 1 | $ chmod +x gfzrnx |

这样就可以在程序所在目录中通过命令 gfzrnx 来使用程序了。如果你希望在任何目录下都可以使用此程序，可以将其移动到 “/usr/bin” 目录下。

**操作模式**

使用 GFZRNX 程序前，首先要了解其操作模式。该程序的输入，即可以来自文件，也可以来自标准输入或管道。对于来自于文件的数据，需要使用 -finp 参数指定。对于程序的输出，默认为标准输出，也可以使用重定向或 -fout 参数将输出转到文件。对于出错信息，默认为标准错误输出，也可以使用 -errlog 参数（或重定向）将出错信息转到文件。因此程序常见的操作模式为：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp <input\_files> -errlog <error\_log> [options] > <output\_file> |

这里的 [options] 代表附加的参数选项，用于指定数据处理中使用的功能。使用 GFZRNX 程序进行数据预处理时，任何符合 RINEX 2 或 3 格式标准的数据都可以作为输入，但只输出最新的 RINEX 2（目前为 RINEX 2.11） 或 3（目前为 RINEX 3.03）版本的数据。如果未明确指定输出数据的格式版本，则默认为 RINEX 3.03。

**格式相关操作**

**格式检查与修复**

GFZRNX 程序提供对 RINEX 格式的检查功能，可以检测你输入的数据文件是否合乎规范。对于格式不正确的内容，甚至会自动尝试修复它。经过格式检查，GFZRNX 将会更新文件头中的信息并且移除所有出错的观测数据。在使用诸如 PANDA、GAMIT/GLOBK 等高精度 GNSS 数据处理程序时，该功能可以解决很多因为数据格式造成的错误。要在数据预处理中使用该功能，只需在操作命令中添加 -chk 参数。

以 BJFS 站 2016 年 9 月 12 日的观测数据 bjfs2560.16o 为例，其原始数据格式为 RINEX 2.10。下面的命令将调用 GFZRNX 程序对该文件进行格式检查，并将尝试修复后的数据保存至新的文件 bjfx2560.16o：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp bjfs2560.16o -chk > bjfx2560.16o |

该命令执行完成后，如果你查看生成的 bjfx2560.16o 文件，将发现该文件已经符合最新的 RINEX 3 格式标准。如果你不希望将输出的文件转换为 RINEX 3 格式，可以添加一个 -kv 参数，例如：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp bjfs2560.16o -chk -kv > bjfx2560.16o |

此时查看生成的新文件 bjfx2560.16o，将发现其数据格式没有变成 RINEX 3，但也不再是原始的 RINEX 2.10，而是 RINEX 2.11。因为这是最新的 RINEX 2 格式标准，-kv 参数只指定数据格式大版本不发生变化。

**格式转换**

使用 GFZRNX 程序的数据格式转换功能时，只需在操作命令中使用 -vo 参数。其输入值被限制为 2 或 3，代表输出数据的 RINEX 格式版本。

以 CHAN 站点于 2016 年 9 月 12 日的观测数据为例，其原始文件为 RINEX 2.11 格式。下面的命令将其格式转化为 RINEX 3：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp chan2560.16o -vo 3 > CHAN00CHN\_R\_20162560000\_01D\_30S\_MO.rnx |

转化广播星历文件：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp chan2560.16n -vo 3 > CHAN00CHN\_R\_20162560000\_01D\_GN.rnx |

转化气象数据文件：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp dav12560.16m -vo 3 > DAV100ATA\_R\_20162560000\_01D\_30S\_MM.rnx |

类似的，下面的命令将之前的命令所生成的文件转化回 RINEX 2：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp CHAN00CHN\_R\_20162560000\_01D\_30S\_MO.rnx -vo 2 > chan2560.12o |

如果你在 UNIX/Linux 操作系统上进行操作，还可以通过管道使 GFZRNX 程序与 RNXCMP、TEQC 等程序相配合。例如：

|  |  |
| --- | --- |
| 1 | $ crx2rnx CHAN00CHN\_R\_20162560000\_01D\_30S\_MO.crx - | gfzrnx -f -vo 2 > chan2560.16o |

运行该命令后，将从最初的 CHAN00CHN\_R\_20162560000\_01D\_30S\_MO.crx 文件直接得到符合 RINEX 2 格式标准的文件 chan2560.16o。这里的 -f 参数指示强制覆盖可能的已有文件。

至此，GFZRNX 程序的格式检查、修复与转换功能已经介绍完毕。对于此程序其他功能的使用介绍，请查看本站 [#GFZRNX](http://www.gnss.help/tags/GFZRNX/)标签中的文章。

**GFZRNX 常用的文件编辑命令**

[2016-11-15](http://www.gnss.help/2016/11/15/gfzrnx-edit/)

[GFZRNX](http://www.gnss.help/categories/GFZRNX/)

[GFZRNX](http://www.gnss.help/tags/GFZRNX/), [RINEX编辑](http://www.gnss.help/tags/RINEX%E7%BC%96%E8%BE%91/)

GFZRNX 是由德国波兹坦地学研究中心（GFZ）开发的一款用于 GNSS 数据预处理（适用于 RINEX 2 与 3 版本格式）的程序，支持对 RINEX 格式的观测数据文件（Obs）、广播星历文件（Nav）和气象数据文件（Met）的操作。

[前文](http://www.gnss.help/2016/10/30/gfzrnx-config-translate/)已经介绍过该程序的配置和文件格式转换功能的使用，本文将以实例的方式介绍其文件编辑功能的使用方法,包括文件分割与拼接、数据提取、采样率抽取、观测卫星筛选和观测量编辑等。

**文件编辑命令**

**文件分割**

要使用程序的文件分割功能，只需在运行时添加 -split 参数，然后输入分割文件的时段长度，其中时长以秒为单位。

示例，将 SHAO 站于 2016 年 2 月 11 日全天的观测数据分割为 24 个时长为 1 小时的观测文件：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp shao0420.16o -fout ::RX2:: -split 3600 |

上述命令中的 -fout ::RX2:: 参数指定输出文件以 RINEX 2 的命名方式自动命名。运行该命令，将得到时长为 1 小时的 24 个观测文件：shao042a.16o、shao042b.16o 至 shao042x.16o。

查看输出的这些文件，你将发现它们已被自动转换为 RINEX 3 格式，别忘了这是程序默认的输出格式。但是这可能不是你想要的，如果希望文件分割前后观测数据的大版本号不变，可以在命令中添加 -kv 参数。即：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp shao0420.16o -fout ::RX2:: -split 3600 -kv |

你也可能希望得到 RINEX 3 格式的输出文件，并且以 RINEX 3 格式的命名方式作为文件名。以下为一个示例：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp shao0420.16o -fout ::RX3:: -split 3600 |

需要补充的是，目前该程序似乎存在一个 Bug。当指定分割后的文件时段长于 1 小时，程序将为输出文件命名为类似 “site0010.16o” 的形式。这样的后果是：当 -split 参数指定的时段长度大于 3600 秒，程序将只输出第一个时段的文件。因为后续输出的文件与第一个文件重名，造成程序终止。当然，如果你在命令中还添加了 -f 参数用于强制覆盖重名文件，那么将只得到最后一个时段的观测文件。因为之前输出的文件被覆盖了。鉴于此，要获得时长超过 1 小时的数据时，建议使用下文介绍的数据提取的操作方式。

**数据提取**

数据提取即从观测文件中提取任意一段时间的数据。在使用该功能时，使用 -epo\_beg 参数来指定首历元开始时刻，使用 -d 参数指定以秒为单位的时长。其中输入的开始时刻可使用简化儒略日、GPS 周、年月日、年积日等多种形式。

依然以上文使用的 SHAO 站的观测数据为例。该天为 2016 年第 42 日，第 1883 GPS 周的星期四，对应的简化儒略日为 57429。示例，从 shao0420.16o 中提取 2 点开始，时长为 2 小时的观测数据：

|  |  |
| --- | --- |
| 1 | gfzrnx -finp shao0420.16o -epo\_beg 2016-02-11\_02:00:00 -d 7200 -kv > shao042c.16o |

运行这个命令后，将得到包含所需数据的文件 shao042c.16o。按照日期指定方式的不同，这个命令还可以如此改写：

|  |  |
| --- | --- |
| 1  2  3 | $ gfzrnx -finp shao0420.16o -epo\_beg 18834\_02:00:00 -d 7200 -kv > shao042c.16o # 日期以 GPS 周指定  $ gfzrnx -finp shao0420.16o -epo\_beg 2016042\_02:00:00 -d 7200 -kv -fout shao042c.16o # 日期以年与年积日指定  $ gfzrnx -finp shao0420.16o -epo\_beg 57429\_02:00:00 -d 7200 -kv -fout shao042c.16o # 日期以简化儒略日指定 |

**文件拼接**

说过文件分割与数据提取，现在介绍其逆操作——文件拼接。使用该功能时不需其它参数，只需以 -finp 参数指定要拼接的文件列表。其顺序可以是任意的，GFZRNX 程序能自动确定拼接的顺序。

下面的命令将前面文件分割时得到的 24 个文件中的前 3 个拼接到一起，并保持拼接前后文件的 RINEX 格式大版本号不变：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp shao042a.16o shao042c.16o shao042b.16o -kv > shao0420.16o |

**采样率抽取**

高采样率的文件体积通常很大，重新进行采样率抽取可以对其瘦身。要应用该功能可以使用 -smp 参数指定输出文件的采样间隔：

下面的命令将采样间隔为 30 秒的源文件重采样为 60 秒：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp shao0420.16o -smp 60 > shao0420\_60s.16o |

运行该命令，得到采样间隔为 60 秒的观测文件 shao0420\_60s.16o。

**观测量编辑**

GFZRNX 程序还支持直接对观测量进行编辑。要使用该功能，可以使用 -obs\_types 参数来指定要保留的观测量列表。其中多个项目之间以逗号分隔。

下面的命令将在输出文件中删去除了 L1、L2、P1、P2、C1、C2 之外的观测量：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp daej0420.16o -obs\_types L1,L2,P1,P2,C1,C2 -kv > temp0420.16o |

如果不关心观测频段而只关心观测类型，还可以使用如下的命令：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp daej0420.16o -obs\_types L,P,C -kv > temp0420.16o |

或者只关心观测频段不关心观测类型，可以使用如下的命令：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp daej0420.16o -obs\_types 1,2 -kv > temp0420.16o |

**观测卫星筛选**

有些 GNSS 数据观测量较差，或者受数据处理程序所限，可能需要从观测数据中删除某些观测数据。GFZRNX 程序支持对卫星或卫星系统进行筛选。

-prn 参数和 -no\_prn 参数用于对卫星进行筛选。其中 -prn 参数用于设置保留的卫星，而 -no\_prn 参数用于设置要去除的卫星。对于多个卫星的操作，可以用逗号进行分隔，亦可使用“-”指定起止卫星号。

下面的命令将 GLONASS 卫星 R1 与 R5 的观测数据删除：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp daej0420.16o -no\_prn R01,R05 -kv > temp0420.16o #卫星号最好使用两位数字，否则易出错 |

运行命令后，检查输出的文件 temp0420.16o，发现其中 R1 与 R5 的观测数据被删去了。

下面的命令则用于删除从 R1 到 R5 之间所有卫星的观测：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp daej0420.16o -no\_prn R01-05 -kv > temp0420.16o |

运行命令后，检查输出的文件 temp0420.16o，发现其中 R1、R2、R3、R4、R5 的观测数据都被删去了。

-prn 参数的使用方式与 -no\_prn 类似。以下的命令将在输出文件中只保留对从 G1 到 G30、从 R1 到 R10 卫星的观测：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp daej0420.16o -prn G01-30,R01-10 -kv > temp0420.16o |

**卫星系统筛选**

除了使用 -prn 或 -no\_prn 对某些卫星的数据进行操作，还可以使用 -satsys 参数直接对卫星系统的筛选，以下的命令将在观测文件中删去除 GPS 和 GLONASS 系统之外的所有卫星：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp daej0420.16o -satsys GR -kv > temp2000.16o |

值得注意的是，受到 RINEX 2 标准的限制（RINEX 2.11 标准未定义北斗观测量），筛选包含北斗卫星的观测数据时可能会出现问题。具体表现为，当输出格式为 RINEX 2 时，虽然设置了保留北斗系统卫星，但输出文件中的北斗卫星观测量被空白取代。不过对星历文件操作是没有问题的。

下面的命令将从混合的星历数据中分离出北斗卫星的信息：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp BRDC00IGS\_R\_20170420000\_01D\_MN.rnx -satsys C > brdc0420.17c |

运行命令后，得到只包含北斗卫星轨道信息的星历文件 brdc0420.17c。

**星历重排序**

上文对卫星系统筛选后输出的星历文件中，卫星轨道信息按卫星的 PRN 编号升序排列，这可能不是你想要的。通过 -ns参数可以对其中的卫星轨道信息重新排序。该参数接受两个选项：prn 或 time，其中 prn 指定输出文件中卫星轨道信息按照卫星 PRN 编号排序，而 time 则将按照发布时间排序。

示例，对输入的广播星历文件按照发布时间重新排序：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp brdc0420.17n -ns time -kv > brdn0420.17n |

对输入的广播星历文件按照卫星的 PRN 编号重新排序：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp brdc0420.17n -ns prn -kv > brdn0420.17n |

# GFZRNX 文件头信息编辑功能详解

[2016-12-10](http://www.gnss.help/2016/12/10/gfzrnx-header-edit/)

[GFZRNX](http://www.gnss.help/categories/GFZRNX/)

[GFZRNX](http://www.gnss.help/tags/GFZRNX/), [RINEX编辑](http://www.gnss.help/tags/RINEX%E7%BC%96%E8%BE%91/)

[前文](http://www.gnss.help/2016/11/15/gfzrnx-edit/)在介绍 GFZRNX 程序的文件编辑命令时，并没有提到其对 RINEX 格式文件的文件头信息的导出和编辑功能。相比 TEQC 程序，该程序的文件头信息编辑的功能更强大，但也更复杂。为协调各文章的篇幅，将其独立成此文。

首先需要说明的是，GFZRNX 支持对 RINEX 格式的观测文件（O-文件）、导航文件（N-文件）和气象文件（M-文件）的操作，但考虑到我们编辑得最多的还是观测文件，因此本文的示例以对 O-文件的操作为主。

## 信息导出

介绍信息编辑功能前，让我们先顺便了解一下 GFZRNX 程序文件头信息导出的功能。该功能可以将文件头中的信息提取出来，便于查看、归档。

信息导出功能使用 -meta 参数，支持 TXT、JSON、XML 等格式，使用起来非常简单。下面的命令以 TXT 格式导出观测信息：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp shao0420.16o -meta basic:txt > shao0420.txt |

下面的命令以 JSON 格式导出观测信息：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp shao0420.16o -meta basic:json > shao0420.json |

下面的命令以 XML 格式导出观测信息：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -finp shao0420.16o -meta basic:xml > shao0420.xml |

限于篇幅，为保证行文流畅性，将导出的信息附在文末。

## 信息编辑

GFZRNX 程序的文件头信息编辑包括三种模式：更新/插入模式、替换模式和重命名模式。其中更新/插入模式主要用于修改文件头中的信息，替换模式用于编辑文件头标志，而重命名模式常用于重命名观测的卫星号和观测量。

使用该程序的信息编辑功能时，必须通过 -crux 参数引入一个配置文件。一份配置文件模板可以使用下面的命令获得：

|  |  |
| --- | --- |
| 1 | $ gfzrnx -show\_crux > example.txt |

运行该命令后，得到示例文件 example.txt。但该文件几乎空无一物，并没有太大的演示价值，因此本文将以我的配置文件为例。

### 更新/插入模式

一份更新/插入模式的配置文件就像这样：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | update\_insert:  #-------------  O - SHAO.DAEJ:  "REC # / TYPE / VERS": { 1: "TRIMBLE NETG3" }  "ANT # / TYPE": { 1: "TRM59800.00", 2: "NONE" }  O - 2015209:00000 2016365:86399 - SHAO:  "APPROX POSITION XYZ": { 0: -3857167.6484, 1: 3108694.9138, 2: 4004041.6876 }  "ANTENNA: DELTA H/E/N" : { 0: 0.1209, 1: 0.0008, 2: 0.0007 }  O - SHAO:  "OBSERVER / AGENCY": { 0 : "SHAO", 1 : "SHAO CAS" }  O - DAEJ:  "OBSERVER / AGENCY" + 00000000:000000 20130126:235959: { 0: "KASI", 1: "KASI" }  "OBSERVER / AGENCY" + 20130127:000000 00000000:000000: { 0: "KASI", 1: "KASI KOREA" } |

该文件首行的“update\_insert”即声明了该配置为更新/插入模式。第 2 行为注释，以“#”号开头。

之后的三行作为一组。其中第 3 行的首个字符定义了要编辑的数据类型（O、M、N 分别代表 RINEX 格式的观测文件、气象文件和导航文件）。若省略该字符，则表示将设置应用到所有数据类型。连字符之后是要应用这些设置的目的站点列表，多个点名之间使用“.”号分隔。因此这里的配置指示：将这一组设置应用到 SHAO 和 DAEJ 两个站点。如果你希望将配置应用到所有输入的文件，可以在这里使用 “ALL”。

文件的第 4 行是一个配置项。如你所知，RINEX 文件头信息中包含“REC # / TYPE / VERS”的这一行有 3 个数据项，分别表示接收机编号、接收机类型和接收机版本。如果我们将这一行的数据项以 0 开始编号，那么三项对应的是：

* 0 ：接收机编号；
* 1 ：接收机类型；
* 2 ：接收机版本。

因此，这个配置项意为：更新“REC # / TYPE / VERS”这一行的信息，将其中编号为 1 的项目（即接收机类型）修改为“TRIMBLE NETG3”。

文件第 5 行与上一行类似，更新天线类型为“TRM59800.00”，更新天线罩为“NONE”。

第 7 行与第 3 行类似，但在文件类型和站点名之间插入了一段字符“2015209:00000 2016365:86399”。这段字符指明，以下的配置仅应用于观测时间在该时间段之内的数据。其中的起止时刻以年、年积日和秒数的方式指定。之后的两行配置分别更新了观测信息中的先验坐标和天线偏移。

第 10 行和第 11 行配置将 SHAO 站的观测者和观测机构分别修改为“SHAO”和“SHAO CAS”。这里省略了应用配置的起止时刻，因此默认将此配置应用到所有时间段的观测数据。

最后的 3 行对 DAEJ 站的观测者和观测机构进行配置，但在配置项中插入了以“+”号开始的一个起止时刻。这表示，在 2013 年 1 月 26 日 23 点 59 分 59 秒之前的观测数据，其观测者和观测机构修改为“KASI”，在 2013 年 1 月 27 日 0 点 0 分 0 秒之后的观测数据，其观测者和观测机构分别修改为“KASI”和“KASI KOREA”。这里的时刻是以年月日、时分秒的形式指定的。事实上，该配置文件中的日期和时刻的格式分别各有两种，即年积日、年月日和时分秒、日积秒。

要应用该配置文件，可以将配置文件保存为 updist\_crux.txt。然后执行类似如下的命令：

|  |  |
| --- | --- |
| 1  2 | $ gfzrnx -finp daej0420.16o -crux updist\_crux.txt -kv > temp/dnew0420.16o  $ gfzrnx -finp shao0420.16o -crux updist\_crux.txt -kv > temp/snew0420.16o |

以上两个命令将分别得到修改后的文件 dnew0420.16o 和 snew0420.16o。

### 替换模式

替换模式的配置于之前的更新/插入模式的配置方式有许多相同之处。一份替换模式的配置文件如下所示：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | replace:  #-------  ALL:  string\_from: "PGM/RUN BY/DATE"  string\_to: "PGM / RUN BY / DATE"  DAEJ.SHAO:  regexp\_from: "^(.{60})PGM\s\*/\s\*RUN\s\*BY\s\*/\s\*DATE\s\*$"  regexp\_to: "$1PGM / RUN BY / DATE" |

首行声明配置为替换模式，之后以“#”号开头的一行为注释。第 3 行和第 6 行各指出了配置项的应用到哪些站点。

对于设置将被替换的项目和替换后的内容，既可以使用字符串指定，如第 4 到 5 行；也可以使用[正则表达式](https://zh.wikipedia.org/wiki/%E6%AD%A3%E5%88%99%E8%A1%A8%E8%BE%BE%E5%BC%8F" \t "_blank)指定，如第 7 到 8行。只需分别以“string”和“regexp”作为配置项的前缀。

对于配置项的后缀，“from”指定要被替换的内容，而“to”指定替换后的内容。

与之前类似，要应用该配置文件，可以将配置文件保存为 repl\_crux.txt。然后执行类似如下的命令：

|  |  |
| --- | --- |
| 1  2 | $ gfzrnx -finp daej0420.16o -crux repl\_crux.txt -kv > temp/dnew0420.16o  $ gfzrnx -finp shao0420.16o -crux repl\_crux.txt -kv > temp/snew0420.16o |

### 重命名模式

重命名模式常用来修改观测卫星号和观测类型标志。实际上，该模式也只有这两个功能。并且在声明时需要明确指明要重命名的是卫星号还是观测类型。

一份修改卫星号的配置文件如下：

|  |  |
| --- | --- |
| 1  2  3  4  5 | rename: prn  #----------  ON - 20140105:000000 20150101:000000 - E51 - E01: ALL  ON - 20140105:000000 00000000:000000 - E52 - E02: DAEJ.SHAO  E53 - E03: ALL |

首行声明配置为重命名卫星编号，之后的一行为注释。

第 3 行一开始的“ON”指定将配置应用于观测文件与导航文件，然后的两个时刻指定应用配置的时间。之后指定将卫星“E51”重命名为“E01”，应用的站点为所有站点。第 4 行与上一行类似，但限制将配置应用到 DAEJ 和 SHAO 两个站点。最后一行没有指定设置要应用的文件类型和时段，因此该设置将应用到所有时段的所有文件。

一份重命名观测类型的配置文件如下所示：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | rename: obs  #----------  20140105:000000 20150101:000000 - L2X - L2L - G : DAEJ.SHAO  20140105:000000 20150101:000000 - L2L - L2X - G : DAEJ  20140105:000000 20150101:000000 - \*\*X - \*\*L - C : ALL  20140105:000000 20150101:000000 - \*2 - \*1 - G04.G08 : ALL  \*2 - \*1 - C : ALL |

首行声明配置为重命名观测类型，之后的一行为注释。

第 3 行与第 4 行，指定了应用配置的时间段、需重命名的观测类型、重命名后的类型、卫星系统和应用的站点。第 4 行以通配符的方式指定了要重命名的观测类型与重命名后的类型。第 5 行与上一行类似，但限制了应用范围为 G4 和 G8 卫星。最后一行省略了时段，因此该配置将应用于所有观测时段的数据。

应用重命名模式的命令方式与之前的两个模式一致，这里不在赘述。

## 导出信息样例

### TXT 格式导出的观测信息

以下为上文导出的 shao0420.txt 文件的内容。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | antenna:  height:  x = 0  y = 0  z = 0  name = AOAD/M\_T  number = 429  radome = JPLA  data:  epoch:  first = 2016 02 11 00 00 00.0000000  interval = 30.000  last = 2016 02 11 23 59 30.0000000  file:  md5 = c8ad2534683bf037c7a9e77eab2ef0a3  system = G  type = O  version = 2.11  receiver:  firmware = CQ00  name = ASHTECH UZ-12  number = UC2200524020  site:  agency = SHANGHAI OBSERVATORY  name = SHAO  number = 21605M002  observer = GGN  position:  x = -2831733.5830  y = 4675665.9580  z = 3275369.4100 |

### JSON 格式导出的观测信息

以下为上文导出的 shao0420.json 文件的内容（为增加可读性，手动添加了缩进）。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | {  "receiver":{  "number":"UC2200524020",  "name":"ASHTECH UZ-12",  "firmware":"CQ00"  },  "site":{  "number":"21605M002",  "position":{  "y":"4675665.9580",  "x":"-2831733.5830",  "z":"3275369.4100"  },  "name":"SHAO",  "agency":"SHANGHAI OBSERVATORY",  "observer":"GGN"  },  "file":{  "system":"G",  "version":"2.11",  "type":"O",  "md5":"c8ad2534683bf037c7a9e77eab2ef0a3"  },  "data":{  "epoch":{  "first":"2016 02 11 00 00 00.0000000",  "last":"2016 02 11 23 59 30.0000000",  "interval":"30.000"  }  },  "antenna":{  "number":"429",  "name":"AOAD/M\_T",  "height":{  "y":0,  "x":0,  "z":0  },  "radome":"JPLA"  }  } |

### XML 格式导出的观测信息

以下为上文导出的 shao0420.xml 文件的内容。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  <meta\_info>  <antenna>  <height>  <x><![CDATA[0]]></x>  <y><![CDATA[0]]></y>  <z><![CDATA[0]]></z>  </height>  <name><![CDATA[AOAD/M\_T]]></name>  <number><![CDATA[429]]></number>  <radome><![CDATA[JPLA]]></radome>  </antenna>  <data>  <epoch>  <first><![CDATA[2016 02 11 00 00 00.0000000]]></first>  <interval><![CDATA[30.000]]></interval>  <last><![CDATA[2016 02 11 23 59 30.0000000]]></last>  </epoch>  </data>  <file>  <md5><![CDATA[c8ad2534683bf037c7a9e77eab2ef0a3]]></md5>  <system><![CDATA[G]]></system>  <type><![CDATA[O]]></type>  <version><![CDATA[2.11]]></version>  </file>  <receiver>  <firmware><![CDATA[CQ00]]></firmware>  <name><![CDATA[ASHTECH UZ-12]]></name>  <number><![CDATA[UC2200524020]]></number>  </receiver>  <site>  <agency><![CDATA[SHANGHAI OBSERVATORY]]></agency>  <name><![CDATA[SHAO]]></name>  <number><![CDATA[21605M002]]></number>  <observer><![CDATA[GGN]]></observer>  <position>  <x><![CDATA[-2831733.5830]]></x>  <y><![CDATA[4675665.9580]]></y>  <z><![CDATA[3275369.4100]]></z>  </position>  </site>  </meta\_info> |