



**GlobalMapper**

---

**脚本语言参考指南 v19**



**Blue Marble GEOGRAPHICS**

*Mind the gap between world and map*  
[www.bluemarblegeo.com](http://www.bluemarblegeo.com)

# 一般概述

Global Mapper 脚本文件允许用户创建自定义批处理，这些批处理将使用内置到 Global Mapper 中的功能。从脚本中，可以导入软件支持的多种格式的数据，根据需要重新投影该数据或对其执行分析，并将其导出到新文件。可以使用 **文件>运行脚本...** 选项从应用程序界面中运行脚本。它们还可以通过双击 \*.gms 文件或从安装了 Global Mapper 的计算机上的使用命令行调用，在后台自动运行。

## 语法

Global Mapper 脚本文件由一系列命令行组成。每个命令行都以命令 **COMMAND** 开始。命令后跟随着一系列参数/值对。这些参数/数值对应写为 **parameter=value**。在等号之前或之后不应存在空格。

单个参数/数值对应应该用空格分隔。如果一组值内部需要空格，则可以围绕整个值使用引号。例如，对于带有空格的文件名，参数组可能像这样 **FILENAME="c:\\my documents\\test.tif"**。

命令、参数和值是不区分大小写的。在本指南中，它们全部是以大写形式编写的。

每个命令行通常由一行组成。若要将命令扩展到另一行，请在行尾使用反斜杠字符（\）。有一些例外，包括 [DEFINE\\_PROJ](#) 和 [DEFINE\\_SHAPE](#) 命令，以及 [DIR\\_LOOP\\_START](#) 和 [DIR\\_LOOP\\_END](#) 命令提供的循环功能。

## 布尔值

脚本语言中的许多参数都是布尔值，这意味着它们只有两个状态：打开或关闭。

本指南列出的值列表，**YES** 是启用该选项，**NO** 将其禁用，但有许多可接受的值可以指定状态：

- **YES** 可以用 **YES**、**Y**、**TRUE**、**T**、**1** 或无值表示（见下文）。
- **NO** 可以用 **NO**、**N**、**FALSE**、**F** 或 **0** 表示。

仅使用具有 **YES** 或 **NO** 参数期望值的参数名称，即可启用或禁用功能的参数（从 v13.1 开始）。因此，不必用 **FLAG\_PARAM\_NAME=YES**，只用 **FLAG\_PARAM\_NAME** 就可以得到指定的相同的行为。

## 通配符和变量

某些参数接受包括通配符在内的值。

- **\***（星号）表示任意长度的通配符。
- **?**（问号）表示单个字符通配符。

参数还可以将变量作为值或值的一部分。变量由 **%VARIABLE%** 分隔。有些变量是内置的，但也可以在脚本中定义。在命令中使用自定义变量之前，必须先定义自定义变量（脚本按顺序从上到下运行）。通常，变量是在脚本的顶部定义的，便于使用。如果定义的变量与以前定义的变量同名，则它将脚本其余部分替换为新的值。

有关详细信息，请参阅 [DEFINE\\_VAR](#)。

## 注释

以正斜杠字符（/）开头的任何行都被视为注释，并被脚本处理引擎忽略。这意味着您可以在行的开头使用 C 或 C++ 样式的注释（如 // 或 /\*）。



**请注意：** 有关用户创建的在文本编辑器中常见的 [语法突出显示](#) 和来自用户的 [共享示例脚本](#)，请参阅 [Global Mapper 用户论坛](#)。

## 脚本命令快速参考

- [ADD\\_MEASURE\\_ATTRS](#) - 添加/更新测量属性到图层中的图元
- [APPLY\\_FORMULA](#) - 将公式应用于加载的栅格图层以创建新的栅格图层
- ASSIGN\_TYPE - 已弃用，使用 [EDIT\\_VECTOR](#) 代替
- [CALC\\_ATTR](#) - 基于现有属性和第二个值计算新的属性值
- [CALC\\_ATTR\\_FORMULA](#) - 基于结合现有属性的公式计算新的属性值
- [CALC\\_VOLUMES](#) - 使用当前高程数据计算区的体积

- [CALC\\_VOLUME\\_BETWEEN\\_SURFACES](#) - 计算两个高程网格之间的体积
- [COMBINE\\_LINES](#) - 根据属性值将连接的线图元结合到新的线或区中
- [COMBINE\\_TERRAIN](#) - 结合两个已加载的地形层生成新的地形层
- [COPY\\_ATTRS](#) - 将属性从一种类型的图元复制到另一个空间
- [COPY\\_LAYER\\_FILES](#) - 将已加载图层的基本文件复制到新的磁盘位置
- [CROP\\_AREAS\\_TO\\_LINES](#) - 基于线图元拆分或裁剪区图元
- [DEFINE\\_LAYER\\_STYLE](#) - 定义供以后使用的图层样式
- [DEFINE\\_PROJ](#) - 定义供以后使用的投影
- [DEFINE\\_SDB\\_CONNECTION](#) - 定义空间数据库连接
- [DEFINE\\_SHADER](#) - 定义供以后使用的高程/坡度着色器
- [DEFINE\\_SHAPE](#) - 定义供以后使用的形（即多边形）
- [DEFINE\\_TEXT\\_FILE](#) - 定义带图元的嵌入式文本文件以使用 [IMPORT\\_ASCII](#) 加载
- [DEFINE\\_VAR](#) - 定义供以后使用的变量
- [DEFINE\\_VAR\\_TABLE](#) - 定义用于查找的变量值表
- [DIR\\_LOOP\\_END](#) - 结束目录中的文件上的命令循环
- [DIR\\_LOOP\\_START](#) - 在目录中的文件上启动一个命令循环
- [EDIT\\_MAP\\_CATALOG](#) - 创建或编辑地图目录
- [EDIT\\_VECTOR](#) - 编辑与类型/名称/属性查询匹配的已加载矢量图元
- [EMBED\\_SCRIPT](#) - 在此脚本中运行另一个脚本文件
- [EXPORT\\_ANY](#) - 基于目标类型自动使用合适的导出命令
- [EXPORT\\_CLOUD](#) - 导出数据到云（Amazon s3）
- [EXPORT\\_ELEVATION](#) - 导出高程数据到文件
- [EXPORT\\_METADATA](#) - 导出图层的元数据到文件
- [EXPORT\\_PACKAGE](#) - 导出数据到 Global Mapper 打包文件
- [EXPORT\\_PDF](#) - 导出数据到 PDF 文件
- [EXPORT\\_PDF3D](#) - 导出数据到 3D PDF 文件
- [EXPORT\\_RASTER](#) - 导出栅格和高程数据到文件
- [EXPORT\\_VECTOR](#) - 导出矢量数据到文件
- [EXPORT\\_VECTOR\\_SPATIAL\\_DB](#) - 导出矢量数据到空间数据库文件
- [EXPORT\\_WEB](#) - 导出矢量数据到 web 文件
- [FORCE\\_EXIT](#) - 强制 Global Mapper 立即退出并返回代码
- [GENERATE\\_CONTOURS](#) - 从高程数据生成地形
- [GENERATE\\_ELEV\\_GRID](#) - 从加载的 3D 矢量数据生成高程网格
- [GENERATE\\_EQUAL\\_VAL\\_AREAS](#) - 从高程/地形层中的相等值生成区图元
- [GENERATE\\_LAYER\\_BOUNDS](#) - 为每个加载层生成具有边界区图元的图层
- [GENERATE\\_PATH\\_PROFILE](#) - 生成 3D 路径剖面并将其保存到 XYZ 文件
- [GENERATE\\_POINTS\\_FROM\\_ELEV\\_GRID](#) - 在高程单元中心生成点
- [GENERATE\\_REPORT](#) - 生成有关加载的矢量图元的报告
- [GENERATE\\_RIDGE\\_LINES](#) - 从高程数据生成山脊线
- [GENERATE\\_VIEWSHED](#) - 从高程数据生成视图
- [GENERATE\\_WATERSHED](#) - 从高程数据生成流域
- [GLOBAL\\_MAPPER\\_SCRIPT](#) - 脚本标头行
- [IF/ELSE IF/ELSE/END IF](#) - 基于变量值的条件执行（If/Then/Else）
- [IMPORT](#) - 从文件导入数据
- [IMPORT\\_ARCHIVE](#) - 从存档文件导入数据（.zip、.tar.gz 等）
- [IMPORT\\_ASCII](#) - 从文件导入通用 ASCII 数据
- [IMPORT\\_CLOUD](#) - 导入云数据集
- [IMPORT\\_DIR\\_TREE](#) - 导入目录树中的所有数据文件
- [IMPORT\\_OSM\\_TILE](#) - 导入平铺（OSM/TMS/Google 地图/Bing 地图）在线源
- [IMPORT\\_SPATIAL\\_DB](#) - 导入空间数据库
- [IMPORT\\_TERRASERVER](#) - 已弃用，Terraserver-USA/MSRMAPS.COM 服务截至2012年5月1日

- [IMPORT\\_WMS](#) - 导入 WMS 层
- [JOIN\\_TABLE](#) - 将属性从文件链接到加载的矢量层
- [LAYER\\_LOOP\\_END](#) - 结束在加载层上的命令循环
- [LAYER\\_LOOP\\_START](#) - 在加载层上启动一个命令循环
- [LIDAR\\_CLASSIFY](#) - 自动分类 Lidar 点
- [LIDAR\\_COMPARE](#) - 将点云与控制点进行比较
- [LIDAR\\_EXTRACT](#) - 从 Lidar 自动提取建筑物轮廓和树点
- [LOAD\\_PROJECTION](#) - 从 PRJ 文件加载新的全局投影
- [LOAD\\_STYLE\\_FILE](#) - 加载样式/类型文件 (.gm\_style)
- [LOAD\\_TYPE\\_FILTER](#) - 已弃用, 使用 SET\_OPT 代替 (从 GMF (Global Mapper 筛选器) 文件加载 Lidar 筛选器或类型筛选器)
- [LOG\\_MESSAGE](#) - 记录状态消息
- [MAP\\_LAYOUT](#) - 定义地图布局 (边距、比例等)
- [PAN\\_SHARPEN](#) - 平移锐化带有平移图像的颜色图层 (创建新图层)
- [PLAY\\_SOUND](#) - 播放蜂鸣声或指定的声音文件
- [QUERY\\_LAYER\\_METADATA](#) - 将图层元数据值放在一个变量中
- [RESTORE\\_LAST\\_SAVED\\_VIEW](#) - 恢复上次保存的视图
- [RUN\\_COMMAND](#) - 运行命令行
- [SAVE\\_CURRENT\\_VIEW](#) - 保存当前视图
- [SAVE\\_PROJECTION](#) - 保存当前全局投影到 PRJ 文件
- [SAVE\\_WORKSPACE](#) - 保存工作区 (GMW) 文件与加载的层
- [SET\\_BG\\_COLOR](#) - 设置背景颜色
- [SET\\_LAYER\\_OPTIONS](#) - 更新加载层的显示选项
- [SET\\_LOG\\_FILE](#) - 设置日志文件的名称
- [SET\\_OPT](#) - 设置常规选项 (如位置显示格式、显示选项等)
- [SET\\_VERT\\_DISP\\_OPTS](#) - 设置垂直显示选项
- [SET\\_VIEW](#) - 设置显示视图
- [SHIFT\\_LAYER](#) - 按指定的距离移动图层的位置
- [SHOW\\_3D\\_VIEW](#) - 显示 3D 视图窗口
- [SORT\\_LAYERS](#) - 根据某些条件对加载的层进行排序
- [SPLIT\\_LAYER](#) - 基于属性拆分图层
- [UNLOAD\\_ALL](#) - 卸载所有当前加载的数据
- [UNLOAD\\_LAYER](#) - 卸载单个图层
- [VAR\\_LOOP\\_END](#) - 在值的范围内结束一个命令循环
- [VAR\\_LOOP\\_START](#) - 在数值值范围内启动一个命令循环
- [VIEW\\_LAYOUT](#) - 定义多视图布局

## 特殊参数类型

- [属性名称](#)
- [到多边形/区的裁剪操作](#)
- [网格化/平铺为小区块的操作](#)
- [投影规格](#)
- [指定用于操作的边界框](#)
- [Lidar 高级筛选选项](#)

# 常见的脚本任务

## GLOBAL\_MAPPER\_SCRIPT

在 V18 之前，**GLOBAL\_MAPPER\_SCRIPT** 必须是脚本文件中的第一个命令。通常情况下，整个命令行将如下所示：

```
GLOBAL_MAPPER_SCRIPT VERSION=1.00
```

您可以对此命令使用以下参数：

- **VERSION** - 指定所使用的脚本语言的版本。此参数是必需的。您应该始终使用 **VERSION=1.00**。
- **ENABLE\_PROGRESS** - 指定在此脚本处理时是否应显示任何进度对话框。默认情况下，在主地图视图的上下文中或在加载工作区时运行的脚本启用此功能。脚本在其自己的沙箱中运行，或者从它们的命令行中默认禁用进度。使用 **ENABLE\_PROGRESS=NO** 处理此脚本的过程中禁用任何进度对话框的显示。
- **GM\_VERSION** - 指定保存工作区 (.gmw) 文件的 Global Mapper 版本。在 v16.2.2 和更高版本中获得支持。格式将像 **GM\_VERSION="16.2.2"**。
- **TIMESTAMP** - 指定保存工作区的时间。这将按 ISO-8601 时间格式保存，如 **TIMESTAMP="2015-06-03T13:08:39Z"**。
- **SHOW\_WARNINGS** - 指定在将工作空间 / 脚本加载到主地图视图中后是否应显示警告消息。使用 **SHOW\_WARNINGS=NO** 禁用警告的显示。任何真正的错误消息将始终显示。
- **LOG\_TO\_COMMAND\_PROMPT** - 指定是否应将记录的消息写入调用命令提示符（如果脚本在命令行上传递）。如果添加 **LOG\_TO\_COMMAND\_PROMPT=YES** 并在命令行上传递 .gms 文件，则任何记录的消息都将写入正常的日志上下文以及命令提示符。请确保使用 **start/wait** 语法或 .bat 文件调用 global\_mapper.exe，以确保命令行输出按需要的位置进行。
- **REQUIRE\_WORKSPACE** - 运行此脚本所需加载的工作区文件的名称。如果为此参数提供了名称，并且该工作区当前未加载到 Global Mapper 中，则脚本将立即中止。如果您有不同的脚本，只有在其他工作区处于活动状态并希望防止意外选择错误脚本时，才需要使用这些脚本，则可以使用此方法。

## SAVE\_WORKSPACE

**SAVE\_WORKSPACE** 命令将当前加载的图层保存到工作区 (GMW) 文件中。该命令支持以下参数。

- **FILENAME** - 保存 GMW 文件的完整路径

## EMBED\_SCRIPT

**EMBED\_SCRIPT** 命令允许您从脚本中调用另一个脚本或加载工作区文件。这在许多情况下都很有用。例如，如果要为每个脚本操作加载一组通用数据文件，则只需创建加载这些文件的脚本，然后在其他脚本中嵌入该脚本即可。

该命令支持以下参数：

- **FILENAME** - 要运行的脚本或工作区文件的完整路径
- **SKIP\_UNLOAD\_ALL** - 指定是否应跳过正在运行的脚本文件中的第一个 **UNLOAD\_ALL** 或 **UNLOAD\_LAYER** 命令。这对于嵌入工作区文件很有用，这些文件通常会卸载开始使用的所有内容。使用 **SKIP\_UNLOAD\_ALL=YES** 跳过第一个 **UNLOAD\_ALL** 或 **UNLOAD\_LAYER** 命令。



为了安全起见，可以在应用程序界面中的**配置**对话框中的**常规**选项卡的**高级**部分禁用此命令。如果已禁用，则任何具有 **RUN\_COMMAND** 的脚本都不会执行该命令，并将产生错误。无法通过脚本重新启用它。

## RUN\_COMMAND

**RUN\_COMMAND** 命令允许您使用一组参数在 Windows 上执行任何程序。该命令支持以下参数：

- **COMMAND\_LINE** - 要运行的完整的命令行。如果需要在命令行中使用引号，请改用单引号（即 ' 而不是 "），它们将在运行命令之前转换为双引号。注意要运行 DOS shell 命令，如 **mkdir**，您将需要通过 **cmd.exe** 运行它，如 **COMMAND\_LINE="cmd /C mkdir 'C:\temp\export test\new folder'"**（注意使用单引号）。
- **PRESERVE\_APOSTROPHES** - 指示是否将撇号转换为命令字符串中的双引号。使用 **PRESERVE\_APOSTROPHES=YES** 保留省略号。默认值为 **NO**。因此，如果不指定此参数，命令字符串中的单引号将转换为双引号。

- **WAIT\_FOR\_COMPLETE** - 指定脚本是否应等待命令行运行完成后再继续。默认值是等待命令完成（即 **WAIT\_FOR\_COMPLETE=YES**）。如果您只希望命令行运行，然后立即让脚本继续处理，请使用 **WAIT\_FOR\_COMPLETE=NO**。
- **CAPTURE\_RESULT** - 指定将存储程序结果的变量的名称。在这里使用该变量之前，不必通过 **DEFINE\_VAR** 命令创建该变量。但如果是，则可以创建该变量。只有当 **RUN\_COMMAND** 等待程序完成时，才会存储结果。请参阅 [WAIT FOR COMPLETE](#)。
- **HIDE\_WINDOW** - 指定命令启动的任何窗口（如命令窗口）最初都将被隐藏。添加 **HIDE\_WINDOW=YES** 到隐藏窗口。

## 示例

下面是运行 Global Mapper 的另一个实例并加载文件的示例：

```
RUN_COMMAND COMMAND_LINE="'c:\program files (x86)\GlobalMapper16\global_mapper.exe'
'c:\temp\export test\blue_springs.opt'" WAIT_FOR_COMPLETE=NO
```

下面是一个调用另一个 .exe 并将 .exe 的返回代码存储到变量 **RESULT** 的示例：

```
RUN_COMMAND COMMAND_LINE="'c:\temp\test1.exe'" CAPTURE_RESULT="RESULT"
```

## PLAY\_SOUND

**PLAY\_SOUND** 命令播放系统的信息声音或指定的声音文件。如果您希望在脚本完成时进行可听到的确认，这将非常有用。该命令支持以下参数：

- **FILENAME** - 播放声音文件（如 .wav）的完整路径。如果未指定，则将播放信息蜂鸣声。

## FORCE\_EXIT

**FORCE\_EXIT** 命令中止脚本，并可选择立即关闭 Global Mapper，而无需执行正常的关机过程。如果您通过 **CreateProcess** 调用运行 Global Mapper 脚本，并且 Global Mapper 程序在脚本完成时没有返回，或者需要提供特定的返回代码，这将非常有用。此命令支持以下参数：

- **CLOSE\_APP** - 指定除了停止脚本之外，是否应关闭 Global Mapper 应用程序。默认情况下，启用此功能。因此添加 **CLOSE\_APP=NO** 仅中止脚本而不是整个应用程序。
- **RETURN\_CODE** - 指定要使用的数字返回代码。如果未提供，则脚本没有遇到任何错误，返回 **0**；或者遇到错误，返回 **1**。

## LOG\_MESSAGE

**LOG\_MESSAGE** 命令将字符串写入状态窗口和任何活动日志文件。您可以使用 **SET\_LOG\_FILE** 命令来设置日志文件以保存消息。该命令的 **USER\_FILENAME** 参数允许您将 **LOG\_MESSAGE** 文本写入与默认脚本消息不同的文件。如果要记录变量的值，可以在命令字符串中包含变量。**LOG\_MESSAGE** 后的所有内容都将被写入。例如，如果您有一个名为 **WATER\_LEVEL\_FT** 的变量，则可以在前面记录其值和时间戳。如下所示：

```
LOG_MESSAGE %TIMESTAMP%: The current value of WATER_LEVEL_FT is %WATER_LEVEL_FT%
```

如果要将消息记录到命令行（如果运行在命令行上传递的脚本），请确保在脚本开头的 **GLOBAL\_MAPPER\_SCRIPT** 标头中包含 **LOG\_TO\_COMMAND\_PROMPT=YES**。

其他内置变量（请参阅 [DEFINE\\_VAR](#)）允许您记录脚本经历的时间（以秒为单位）。例如，您可以使用以下方法记录导入和导出的时间以及脚本运行的总时间：

```
GLOBAL_MAPPER_SCRIPT VERSION=1.00

LOG_MESSAGE Script <%SCRIPT_FILENAME%> started at %DATE% %TIME%

IMPORT FILENAME="test.tif"

LOG_MESSAGE Import took %TIME_SINCE_LAST_LOG%

EXPORT_RASTER TYPE=GEOTIFF FILENAME="output.tif"

LOG_MESSAGE Export took %TIME_SINCE_LAST_LOG%

LOG_MESSAGE Total Script Run Time: %TIME_SINCE_START%
```

## SET\_LOG\_FILE

**SET\_LOG\_FILE** 命令设置状态、警告和错误消息日志记录文件的名称。如果指定的日志文件已经存在，则消息将追加到其末尾。该命令支持以下参数：

- **FILENAME** - 要将消息写入的日志文件的完整路径。
- **USER\_FILENAME** - 日志文件的完整路径，以写入 **LOG\_MESSAGE** 命令的消息。如果您希望自己的消息转到与任何自动生成的脚本消息不同的文件，请使用此选项。用一个空值传入以重置 **LOG\_MESSAGE** 消息以返回到共享文件。

```
IF COMPARE_STR="%VAR1%=val1"

// 条件为 true 时要运行的附加脚本命令。

END_IF
```

## Global Mapper 脚本中的条件执行

Global Mapper 脚本可以通过使用 **IF**、**ELSE\_IF**、**ELSE** 及 **END\_IF** 命令合并条件逻辑。至少，在将条件逻辑合并到脚本中时，用户必须使用 **IF** 和 **END\_IF** 命令，如下面的示例所示：

运行包含示例的脚本时，如果变量 **%VAR1%** 的值为 **val1**，则将执行 **IF** 之后和 **END\_IF** 之前的语句。如果变量 **%VAR1%** 的值不是 **val1** 的值，则将跳过 **IF** 后面的语句，并且脚本处理将继续处理 **END\_IF** 之后的第一个语句。如果用户希望在 **IF** 上指定的条件不正确的情况下运行一组特定的命令，则可以使用 **ELSE** 命令：

```
IF COMPARE_STR="%VAR1%=val1"

// 当 IF 条件为true 时运行的脚本命令。

ELSE

// 当 IF 条件为 false 时运行的脚本命令。

END_IF
```

现在，如果变量 **%VAR1%** 的值为 **val1**，则将执行 **IF** 之后和 **ELSE** 之前的语句，并跳过 **ELSE** 之后和 **END\_IF** 之前的命令。如果变量 **%VAR1%** 的值不是 **val1**，则将跳过 **IF** 之后和 **ELSE** 之前的语句，并且将运行 **ELSE** 之后和 **END\_IF** 之前的命令。

如果用户有几个条件需要测试，其中只有一个可以是 **true**，则可以使用 **ELSE\_IF** 命令：

```
IF COMPARE_STR="%VAR1%=val1"

// 当 IF 条件为真时运行的脚本命令。

ELSE_IF COMPARE_STR="%VAR1%=val2"

// 当 ELSE_IF 条件为 true 时运行的脚本命令。

ELSE

// 当所有其他条件为 false 时运行的脚本命令。

END_IF
```

**IF** 后面的命令将按上述方式处理，但现在有第二个条件正在测试中。当变量 **%VAR1%** 的值为 **val2** 时，则在 **ELSE\_IF** 之后和 **ELSE** 之前的命令将运行。在存在多个 **IF/ELSE\_IF** 条件的情况下，当所有其他条件都为 **false** 时，将运行 **ELSE** 之后的命令。

### 逻辑条件命令

**IF** 和 **ELSE\_IF** 命令需要一个 **COMPARE\_STR** 参数来指定要测试的条件。

用户可以指定多个 **COMPARE\_STR** 参数，所有这些参数都必须为 **true**，才能导致执行后续命令（逻辑 **AND**）。使用 **COMPARE\_OP="ANY"** 参数指定，如果其中任何一个条件为 **true**（逻辑 **OR**），则应运行后续命令。

### IF

如果比较条件为 **true**，请执行后续命令。**IF** 语句是一个代码块，必须使用 **END\_IF** 关闭。

- **COMPARE\_STR - COMPARE\_STR** 必须由 **<value><运算符><value>** 组成，其中值可以是常量或变量名（包含在 **"%"** 中）。必须指定这两个值。运算符是必需的，可以是：
  - **"="** - 等于
  - **"!="** - 不等于
  - **"<"** - 小于
  - **"<="** - 小于或等于
  - **">"** - 大于



- **">="** - 大于或等于
- **COMPARE\_NUM** - 默认情况下，比较将是不区分大小写的字符串比较。如果要执行数字比较，请指定 **COMPARE\_NUM="YES"** 参数。
- **COMPARE\_OP** - 如果定义了多个比较字符串（**COMPARE\_STR**），请指定 **IF** 语句被视为 **true** 的处理选项。
  - **ANY** - 通过多次比较，如果一个或多个单独的条件为 **true** (逻辑 **OR**)，则结果为 **true**。
  - **ALL** - 这是默认设置。对于要执行 **IF** 语句之后的命令，所有比较都必须为 **true**。

## ELSE\_IF

可以测试第二个比较条件，如果为 **true**，则将执行后续命令。可用的参数与 **IF** 相同，见上文。

## ELSE

对于 **IF** 或 **ELSE\_IF** 条件，所有其他未返回 **true** 的情况都将执行其他语句之后的命令。

## END\_IF

**IF** 命令可以嵌套，因此 **IF**、**ELSE\_IF** 或 **ELSE** 命令后面的命令块可以包含另一个 **IF** 命令：

```
IF COMPARE_STR="%VAR1%=val1" IF COMPARE_STR="%VAR2%>10"
// 当 IF 条件为 true 时运行的脚本命令。
END_IF
ELSE
// 当 IF 条件为 false 时运行的脚本命令。
END_IF
```



# 批处理模式操作

通过将 Global Mapper 脚本在命令行上传递到 Global Mapper.exe 文件，可以自动运行该文件。脚本文件将在不显示用户界面的情况下运行，当脚本文件完成处理时，Global Mapper 将立即退出。这使您可以轻松地从一个应用程序或 DOS 批处理文件运行 Global Mapper 脚本。



请注意，脚本文件需要具有 .gms 的扩展名才能正常工作。

## 批处理变量

在批处理模式下运行时，可以在命令行上定义变量，以便在运行脚本时可以使用这些变量。在文件名之后的命令行上提供一对令牌。每一对必须如下所示：

-<var name> <var value>

或

/<var name> <var value>

### 命令行示例：

```
global_mapper.exe "c:\temp\myscript.gms" -var1 01 -var2 33
```

这定义了可在脚本中使用的两个变量：**var1=01** 和 **var2=33**。有关如何使用变量的详细信息，请参阅 [DEFINE VAR](#) 命令。

## 批处理选项

**/showprogress** - 如果存在，指示 Global Mapper 在处理脚本时显示进度条。此参数仅在从命令行运行脚本时才有效。

### 命令行示例：

```
"C:\Program Files\GlobalMapper17_64bit\global_mapper.exe" "C:\Scripts\export.gms" /showprogress
```

# 脚本示例

裁剪、合并和重投影 4 个 USGS DRG 到新的 GeoTIFF 和 JPEG 文件。

```
GLOBAL_MAPPER_SCRIPT VERSION=1.00 UNLOAD_ALL

// 导入我们想要合并的四个24K DRG。我们使用 CLIP_COLLAR 选项
// 以表明我们希望圈环在导入时自动从 DRG 中移除。
IMPORT FILENAME="C:\DATA\DRG\KANSAS CITY\039094B2.TIF" \
TYPE=AUTO ANTI_ALIAS=NO AUTO_CONTRAST=NO CLIP_COLLAR=AUTO TEXTURE_MAP=NO
IMPORT FILENAME="C:\DATA\DRG\KANSAS CITY\039094A1.TIF" \
TYPE=AUTO ANTI_ALIAS=NO AUTO_CONTRAST=NO CLIP_COLLAR=AUTO TEXTURE_MAP=NO
IMPORT FILENAME="C:\DATA\DRG\KANSAS CITY\039094A2.TIF" \
TYPE=AUTO ANTI_ALIAS=NO AUTO_CONTRAST=NO CLIP_COLLAR=AUTO TEXTURE_MAP=NO
IMPORT FILENAME="C:\DATA\DRG\KANSAS CITY\039094B1.TIF" \
TYPE=AUTO ANTI_ALIAS=NO AUTO_CONTRAST=NO CLIP_COLLAR=AUTO TEXTURE_MAP=NO

// 加载投影文件，将全局投影设置为地理（纬度/经度）弧度，其基准为 NAD 83。
LOAD_PROJECTION FILENAME="C:\DATA\PRJ Files\geo_degrees_nad83.prj"
// Use the EXPORT_RASTER command to generate a new 8-bit per pixel GeoTIFF file
EXPORT_RASTER FILENAME="C:\DATA\EXPORTED DATA\merged_drg_8bpp.tif" \
TYPE=GEOTIFF PALETTE=OPTIMIZED

// 现在，使用 EXPORT_RASTER 命令生成一个灰度 GeoTIFF 文件。
// 我们也为这个文件创建一个世界文件。
EXPORT_RASTER FILENAME="C:\DATA\EXPORTED DATA\merged_drg_gray.tif" \
TYPE=GEOTIFF PALETTE=GRAYSCALE GEN_WORLD_FILE=YES

// 使用export_raster命令创建JPEG文件。还创建了一个世界性文件
// 和一个投影文件，使它更容易在其他地方。
EXPORT_RASTER FILENAME="C:\DATA\EXPORTED DATA\merged_drg.jpg"\
TYPE=JPEG GEN_WORLD_FILE=YES GEN_PRJ_FILE=YES
```

从文件夹中的所有 USGS DEM 生成地形，并将其导出到 DXF 和 形文件

```

GLOBAL_MAPPER_SCRIPT VERSION=1.00 UNLOAD_ALL

// Loop over all DEM files in a folder and convert them
DIR_LOOP_START DIRECTORY="C:\DATA\SDTS_DEM\24K\" FILENAME_MASKS="*.DEM.STDS.TAR.GZ" RECURSE_
DIR=NO

// Import an archived SDTS DEM file. Global Mapper will automatically
// determine that this is an archived SDTS DEM file and load it
// correctly.
IMPORT FILENAME="%FNAME_W_DIR%" ANTI_ALIAS=YES
// Generate 50 ft contours from the loaded DEM data.
GENERATE_CONTOURS INTERVAL=50 ELEV_UNITS=FEET
// Export the contours to a new DXF file. The created file will have
// 3D polyline features for the contours.
EXPORT_VECTOR FILENAME="%DIR%FNAME_WO_EXT%_CONTOURS.DXF" TYPE=DXF GEN_PRJ_FILE=YES
// Export the contours to a 3D shape file.
EXPORT_VECTOR FILENAME="%DIR%FNAME_WO_EXT%_CONTOURS.SHP" TYPE=SHAPEFILE \
SHAPE_TYPE=LINES GEN_3D_LINES=YES GEN_PRJ_FILE=YES
// Unload the loaded data
UNLOAD_ALL
// End the loop
DIR_LOOP_END

```

基于属性和显示标签编辑矢量图元

```

GLOBAL_MAPPER_SCRIPT VERSION=1.00

// Import the file to modify
IMPORT FILENAME="C:\Temp\export test\tiger_wyandotte_sample.gmp"

// Assign the type "railroad" to all features with a CFCC attribute with a value of A41
// and a display label with '74' in it somewhere.
EDIT_VECTOR LINE_TYPE="RAILROAD" COMPARE_STR="CFCC=A41" COMPARE_STR("<Feature Name>=*74*"
// Assign the name "Burlington Northern Railroad" to all features with a CFCC attribute with a
value of A41
EDIT_VECTOR ATTR_VAL="<Feature Name>=Burlington Northern Railroad" COMPARE_STR="CFCC=A41"

```

# 公式参考

## 预定义公式

### 归一化差异植被指数 (NDVI)

$$NDVI = (NIR-RED) / (NIR+RED)$$

归一化差异植被指数 (**NDVI**) 是一种常用的植被指数，来源于遥感图像的近红外 (**NIR**) 和红色通道。**NDVI** 是植被光合活性或绿色的指标。植被指数是基于不同表面不同类型的光线反射不同类型的观察。

**NDVI** 是通过确定每个像素内遥感图像的红色波段和近红外波段的比率来计算的，该比率用作图像中红色波段和近红外波段之间的归一化差异。

**NDVI** 有几个预定义的选项，具体取决于在哪个波段中找到 **NIR** 和 红色。

- NDVI (RGBI): NIR = B4, RED=B1
- NDVI (IBGR): NIR=B1, RED= B4
- NDVI (Landsat 8 OLI) : NIR=B5, Red=B4
- NDVI (Landsat 5 TM) : NIR=B4, Red=B3
- NDVI (Sentinel 2 MSI): NIR=B8, Red=B4

### 归一化差水指数 (NDWI)

$$NDWI = (G-NIR) / (G+NIR)$$

归一化差水指数 (**NDWI**) 来源于近红外 (**NIR**) 和 绿色 (**G**) 通道。这个公式突出了水体中的水量。

另一种计算方法是使用近红外 (**NIR**) 和短波红外 (**SWIR**) 通道  $[(NIR-SWIR) / (NIR+SWIR)]$ 。植被中的水量主要影响 **SWIR** 通道的光谱反射率。有关 **SWIR** 通道中包含的植被的信息是唯一的。**NDWI** 应被视为一个独立的植被指数。

- NDWI : G=B2, NIR=B4
- NDWI (IBGR): G= B3, NIR=B1
- NDWI (RGBI): G=B2, NIR=B4
- NDWI (Landsat 8 OLI): G=B3, NIR=B5
- NDWI (Landsat 5 TM): G=B2, NIR=B4
- NDWI (Sentinel 2 MSI): G=B3, NIR=B8

### 归一化燃烧比率 (NBR)

$$NBR = (NIR-SWIR) / (NIR+SWIR)$$

归一化燃烧比率 (**NBR**) 计算用于突出显示燃烧区域和估计火灾的严重程度。燃烧区域在短波红外 (**SWIR**) 中反映的较高和近红外 (**NIR**) 的低，因此具有较低的 **NBR** 值。与 **NDVI** 类似，健康的植被将具有较高的 **NBR** 值。波段数 (B4, B6) 基于 Landsat 4-5 波段名称，但该公式可用于任何图像，方法是在随后的对话框中将近红外波段指定为 B4，将 **SWIR** 波段指定为 B6。

- NBR [Normalized Burn Ratio]: NIR=B4, SWIR/Thermal=B6
- NBRI (Landsat 8 OLI): NIR=B5, SWIR/Thermal=B7
- NBRI (Landsat 5 TM): NIR=B4, SWIR/Thermal=B7
- NBRI (Sentinel 2 MSI): NIR=B8, SWIR/Thermal=B12

### 归一化色素叶绿素比指数 (NPCRI)

$$NPCRI = (Red-Blue) / (Red+Blue)$$

归一化色素叶绿素比指数 (**NPCRI**) 是一个数值指标，可用于测定作物和植被叶绿素含量。这种公式通常存在于进动农业中的应用中。例如，氮含量低的作物可能具有较高的类胡萝卜素与叶绿素的比率。**NPCRI** 使用红色和蓝色光谱波段捕获量化叶绿素和氮所需的信息。该公式也可用于来自各种卫星的红色和蓝色光谱波段。

- NPCRI (Landsat 8 OLI) : Red=B4, Blue=B2
- NPCRI (Landsat 5 TM) : Red=B3, Blue=B1
- NPCRI (Sentinel 2 MSI) : Red=B4, Blue=B2

### 归一化差异降雪指数 (NDSI)

$$\text{NDSI} = (\text{Green} - \text{SWIR}) / (\text{Green} + \text{SWIR})$$

归一化差异降雪指数(**NDSI**) 是一个数字指标, 显示陆地覆盖。在此公式中使用绿色和短波红外 (**SWIR**) 光谱波段绘制雪盖图。由于雪吸收了 **SWIR** 中的大部分入射辐射, 而云却不吸收, 这使得 **NDSI** 能够区分雪和云。该公式常用于雪盖测绘应用以及冰川监测。

- **NDSI** (Landsat 8 OLI) : Green=B3, SWIR=B6
- **NDSI** (Landsat 5 TM) : Green=B2, SWIR=B5
- **NDSI** (Sentinel 2 MSI) : Green=B3, SWIR=B11

## 归一化差异冰川指数 (NDGI)

$$\text{NDGI} = (\text{NIR} - \text{Green}) / (\text{NIR} + \text{Green})$$

归一化差异冰川指数 (**NDGI**) 用于使用绿色和红色光谱波段帮助探测和监测冰川。该公式常用于冰川探测和冰川监测应用。

- **NDGI** (Landsat 8 OLI) : NIR=B3, Green=B4
- **NDGI** (Landsat 5 TM) : NIR=B2, Green=B3
- **NDGI** (Sentinel 2 MSI) : NIR=B3, Green=B4

## 归一化差异水分指数 (NDMI)

$$\text{NDMI} = (\text{NIR} - \text{SWIR}) / (\text{NIR} + \text{SWIR})$$

与植被水分相关的归一化差异水分指数 (**NDMI**), 与其他蔬菜指标 (**NDVI** 和/或 **ADVI**) 结合使用。该公式使用近红外 (**NIR**) 和短波红外 (**SWIR**) 来捕捉植被地区水分的变化。**NDMI** 适用于研究干旱监测和植被水分状况细微变化的应用。**NDMI** 还可用于确定野火危险评估中的燃料水分。

- **NDMI** (Landsat 8 OLI) : NIR=B5, SWIR=B6
- **NDMI** (Landsat 5 TM) : NIR=B4, SWIR=B5
- **NDMI** (Sentinel 2 MSI) : NIR=B8, SWIR=B11

## 高级植被指数 (AVI)

$$\text{AVI} = [\text{NIR} * (1 - \text{Red}) * (\text{NIR} - \text{Red})]^{1/3}$$

高级植被指数 (**AVI**) 类似于 **NDVI**, **NDVI** 使用红色和近红外光谱波段来监测作物和森林随时间的变化。随着 **AVI** 和 **NDVI** 的时间组合允许不同类型的植被和提取物候特征的判别。

- **AVI** (Landsat 8 OLI) : NIR= B5, Red=B4
- **AVI** (Landsat 5 TM) : NIR=B4, Red=B3
- **AVI** (Sentinel 2 MSI) : NIR=B8, Red=B4

## 裸土指数 (BSI)

$$\text{BSI} = ((\text{Red} + \text{SWIR}) - (\text{NIR} + \text{Blue})) / ((\text{Red} + \text{SWIR}) + (\text{NIR} + \text{Blue}))$$

裸土指数(**BSI**) 结合了蓝色、红色、近红外 (**NIR**) 和短波红外 (**SWIR**) 波段, 以捕获土壤变化。这些波段以规范化的方式使用, 用 **SWIR** 和红色波段来量化土壤矿物组成, 而蓝色和近红外波段则用于增强植被的存在。该数值指标可用于土壤测绘和作物鉴定。

- **BSI** (Landsat 8 OLI) : Red= B6, SWIR=B4, NIR=B5, Blue=B2
- **BSI** (Landsat 5 TM) : Red=B5, SWIR=B3, NIR=B4, Blue=B1
- **BSI** (Sentinel 2 MSI) : Red=B11, SWIR=B4, NIR=B8, Blue=B2

## 阴影指数 (SI)

$$\text{SI} = [(1 - \text{Red}) * (1 - \text{Green}) * (1 - \text{Blue})]^{1/3}$$

树冠阴影提供有关树木和植物排列的基本信息。树冠阴影的特征与从树冠反射的总光谱辐射有关。阴影指数(**SI**) 使用红色、蓝色和绿色波段计算, 以模拟未反射回传感器的总能量。这方面的主要应用是林业和作物监测, 可以与 **AVI** 和 **BSI** 结合起来, 了解植被状况。

- **SI** (Landsat 8 OLI) : Red=B2, Green=B3, Blue=B4
- **SI** (Landsat 5 TM) : Red=B1, Green=B2, Blue=B3
- **SI** (Sentinel 2 MSI) : Red=B2, Green=B3, Blue=B4

## 自定义公式

自定义公式允许您根据 **Global Mapper** 中尚未预定义的公式为栅格图像生成新波段。公式是可以组合数字、波段值或高程以计算新波段的数学表达式。他们可以使用标准的数学运算符, 如 **+**、**-**、**\***、**/**、**^** (乘方)、比较和关系运算符, 以及 **LOG**、**LOG10**、**MIN** 和 **MAX** 等函数来生成新值。

## 操作数

- **数字** - 下面是一些数字的例子：
  - **3.14159**
  - **42**
  - **1234e-3**（科学记数法，结果为 **1.234**）
  - **123,456**（欧洲记数法，结果为 **123,456**）
- **波段值** - 波段值指图像中的特定栅格波段。允许的波段引用（同样，下面的名称不区分大小写）：
  - **red** 【或 **R**】：红色波段
  - **green** 【或 **G**】：绿色波段
  - **blue** 【或 **B**】：蓝色波段
  - **B#** 【其中 **#** 是一个大于零的整数】：一个波段的编号，例如，**B3**
  - **elev** 【或 **Z**】：用于地形图层的高程伪波段（在这种情况下，与 **B1** 是等效的）

## 运算符

公式使用各种数学和逻辑运算符来形成结果，类似于电子表格公式。它们是（优先顺序，按从低到高）：

- **OR** 【逻辑 **OR**】：两个操作数都被视为布尔值，并返回布尔值（如果其中一个操作数不是零，则为 **1**，否则为 **0**）。
- **AND** 【逻辑 **AND**】：两个操作数都被视为布尔值，并返回布尔值（如果两个操作数都是非零，则为 **0**）。
- **=, <>, ~=** 【比较运算符】：等于、不等于和不区分大小写的字符串比较。
- **<, <=, >, >=** 【关系运算符】：分别是小于、小于或等于、大于和大于或等于。
- **+, -** 【加法运算符】：分别为加和减。
- **\*, /** 【乘法运算符】：分别为乘和除。操作数被假定为数字。
- **^**：指数运算
- **+, -, NOT** 【一元运算符】：分别是正、负和逻辑 **NOT**。一元加和减的操作数被假定为数字；一元 **NOT** 的操作数被假定为布尔值。

您还可以使用括号来指定操作顺序。在没有括号的情况下，在较低优先级操作之前执行较高的优先级操作。也就是说，在公式 "**B1 + r \* 3**" 中，结果是 "**B1**" 加上 "**r**" 和 "**3**" 的乘积的值（等价于 "**B1 + (r \* 3)**"）。

## 函数

计算器提供了几个内置函数来帮助计算值。请注意，函数名称不区分大小写，也就是说，**log** 与 **LOG** 相同。

- **LOG (表达式)**：自然对数。表达式参数假定为数值。
- **LOG10 (表达式)**：基于 10 的对数。表达式参数假定为数值。
- **MAX (表达式 1, 表达式 2)**：表达式 1 和表达式 2 的最大值。
- **MIN (表达式 1, 表达式 2)**：表达式 1 和表达式 2 的最小值。
- **IF (表达式 1, 表达式 2, 表达式 3)**：计算表达式 1，如果为 **true**，返回表达式 2，否则返回表达式 3
- **SIN (表达式)**：返回表达式的正弦，假定为以弧度表示。
- **COS (表达式)**：返回表达式的余弦，假定为以弧度表示。
- **TAN (表达式)**：返回表达式的正切，假定为以弧度表示。
- **SIN\_DEG (表达式)**：返回表达式的正弦，假定以度表示。
- **COS\_DEG (表达式)**：返回表达式的余弦，假定以度表示。
- **TAN\_DEG (表达式)**：返回表达式的正切，假定以度表示。
- **ASIN (表达式)**：返回表达式的反正弦（逆正弦），假定为以弧度表示。
- **ACOS (表达式)**：返回表达式的反余弦（逆余弦），假定为以弧度表示。
- **ATAN (表达式)**：返回表达式的反正切（逆正切），假定用弧度表示。
- **SQRT (表达式)**：平方根。表达式参数假定为大于或等于 **0** 的数值。
- **AND (表达式列表)**：如果所有以逗号分隔的表达式列表都为非零，则返回 **true (1)**，否则返回 **false (0)**。
- **OR (表达式列表)**：如果任何以逗号分隔的表达式列表为非零，则返回 **true (1)**，否则返回 **false (0)**。

# 后记

本书为 Global Mapper 帮助文件中的 GlobalMapper\_ScriptingReference.pdf 文档的中文翻译版。

本书为个人业余时间翻译，并未取得版权方的许可，只可用于学习目的，不可用于商业谋利。

欢迎各位朋友学习交流，本人的[EMAIL: DEFTYOUNG@163.COM](mailto:DEFTYOUNG@163.COM)。

书中的各种特殊文本的样式如下：

命令（函数）文本

**COMMAND**

命令参数名

**PARAMETER**

运算符

**=**

数值（文本）

**VALUE**

变量

**%VARIABLE%**

超级链接

[超级链接](#)

菜单、选项卡

**文件>运行脚本...**

其他代码

`GLOBAL_MAPPER_SCRIP`

脚本代码

`GLOBAL_MAPPER_SCRIPT`

示例代码

`GLOBAL_MAPPER_SCRIPT VERSION=1.00`