**启动脚本**

首次启动3ds Max时，MAXScript会搜索任何启动脚本文件，然后自动加载并运行。 如果你有总是使用并且需要预加载的函数库，或者你想要建立自定义用户界面设置，定义脚本化插件或加载脚本化实用程序卷展栏，则此功能非常有用。

3ds Max启动时，将始终遵循下面描述的脚本加载顺序。 软件的某些部分仍然不能在启动过程的不同阶段启动和运行。了解它的顺序和原因将帮助你组织你的代码以及让所有功能和用户界面自动加载并且在一起正确工作。

**在没有文件的情况下启动3ds Max**

**第1步-标准脚本**

3ds Max将评估在stdplugs \ stdscripts下找到的运输脚本。 这些定义了基础3ds Max软件的几个脚本元素所需的功能。

如果你需要定义一些由你自己的宏脚本使用的函数或全局变量，那么这些函数应该放到stdplugs \ stdscripts中的一个.ms文件中。 这将确保它们将被首先定义。

**第2步-用户定义的插件路径**

接下来，3ds Max将在Customize ...> Configure Paths ...> Plug-Ins及其子目录定义的目录中加载脚本文件。请注意，这与常规插件的加载方式不同。插件将只从指定的路径加载，而不是搜索子目录。在这里，你可以定义其他脚本使用的函数，方法和结构，特别是宏脚本。

3ds Max场景尚未创建，此时尚未创建视口。你不能在这些脚本中创建场景对象或执行任何与用户界面相关的操作。当然，你可以定义执行这些操作的函数，只是你不能调用这些函数。

在这个过程中，任何名为startup.ms的脚本文件都将被忽略。这使你可以将脚本化的插件脚本放置在插件文件夹中进行自动加载。它们可以像DLL插件一样被管理，并用于将启动脚本组织到自己的目录中，以便于管理。你可以通过将其名称放在括号中来防止对嵌套目录进行扫描，例如“（旧版本）”，允许你方便的在基于目录的分组中启用和禁用脚本。

**第3步-宏脚本**

接下来，将评估UI \ macroscripts中的宏脚本 .mcr文件 - 它们定义可以在用户界面中显示的操作项目。 在这一点上，用户界面还没有建立的时候，宏脚本就需要填写他们的相关按钮。

**第4步-启动脚本**

最后，将会评估startup.ms和启动文件夹中的任何脚本。 此时，场景，视口和用户界面都已启动。 你可以在这里做任何你想要的。

在3ds Max 9及更高版本中，首先运行3ds Max系统启动脚本目录中的启动脚本，然后运行用户启动脚本目录中的启动脚本，前提是用户启动脚本目录不同于最大系统启动脚本目录。

MAXScript首先在以下目录中搜索名为startup.ms的文件，顺序如下：

当MAXScript找到第一个出现的startup.ms时停止搜索。

可以通过关闭首选项对话框的MAXScript页面中的自动启动MAXScript选项来关闭启动脚本文件的自动加载，如主产品联机帮助中MAXScript首选项设置主题中所述。

**用文件启动3ds Max**

考虑以下两个用例：

1. 从命令行启动3ds Max，最后使用命令行脚本：

3dsmax.exe somefile.max

3dsmax.exe somefile.max -U MAXScript somescript.ms

1. 将a.MAX文件拖到桌面上的3ds Max快捷方式中

在3ds Max 2010中，加载顺序已更改为：

|  |  |
| --- | --- |
| 文件/回调处理 | 注解 |
| 标准插件\标准脚本 | 尚未创建3ds Max场景和用户界面。 |
| 插件路径 | 尚未创建3ds Max场景和用户界面。 |
| 宏脚本 | 尚未创建用户界面。 |
| \启动 | 用户界面已创建。在回调之前加载的启动脚本！ |
| #文件打开预处理 | 场景创建。 |
| #文件打开前 | 文件对象尚不可访问。 |
| .Max File | 加载命令行/丢弃的文件 |
| #文件打开后 | 场景对象可以从这里访问。 |
| #打开文件后处理 | 文件已经完成加载。 |
| 命令行-U MAXScript脚本 | 执行命令行脚本（如果有的话） |