山东科技大学——测绘与空间信息学院

Python程序设计

地理信息科学系　刘洪强

J6-557　电话：86081170

2021年4月22日星期四

课程安排：

36个学时，其中授课24个学时，实验12个学时

成绩：

出勤5% + 实验报告25% + 考试70%

章节内容

第1章 认识Python

第2章 Python编程基础

第3章 函数、类、包和模块

第4章 文件操作

第5章 地图文档管理

第6章 数据链接查找与修复

第7章 地图制图与输出

第8章 地理处理工具的执行

第9章 地理处理工具的创建

第10章 数据查询与选择

第11章 数据访问模块

第12章 获取GIS数据的列表和描述

第9章　地理处理工具的创建

创建自定义地理处理工具

CreateFeatureclass_management()函数

9.1 创建自定义地理处理工具

除了能够在脚本中使用任何可用的工具外，还可以创建自己的自定义工具，它们也能从脚本中调用。自定义工具通常用来处理特定的地理处理任务，并且也容易共享。

9.1 创建自定义地理处理工具

自定义的脚本工具必须添加到用户创建的自定义工具箱中，因为ArcToolbox提供的系统工具箱是只读工具箱，无法向其中添加新工具。

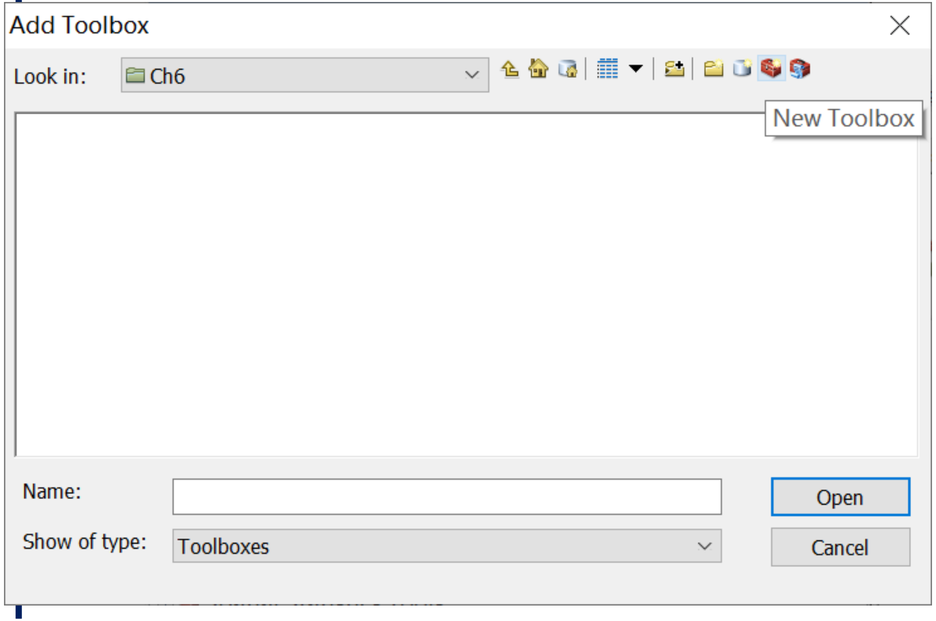
9.1 创建自定义地理处理工具

1. 在Arcmap中，打开一个空的地图文档，然后点击“ArcToolbox”窗口

2. 在ArcToolbox中的任意空白区域右击鼠标，选择“Add Toolbox”

3. 在“Add Toolbox”对话框中，单击“new toolbox”按钮，创建一个新的工具箱，默认名称为“Toolbox.tbx”，将其重命名为“WildfireTools.tbx”

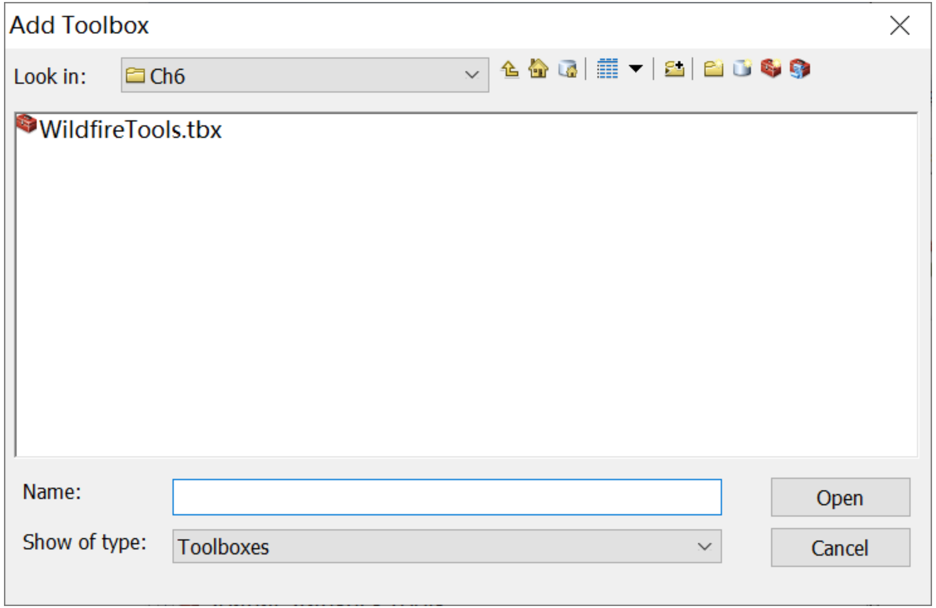
9.1 创建自定义地理处理工具



Look in: Ch6

Name:

Show of type: Toolboxes



Look in: Ch6

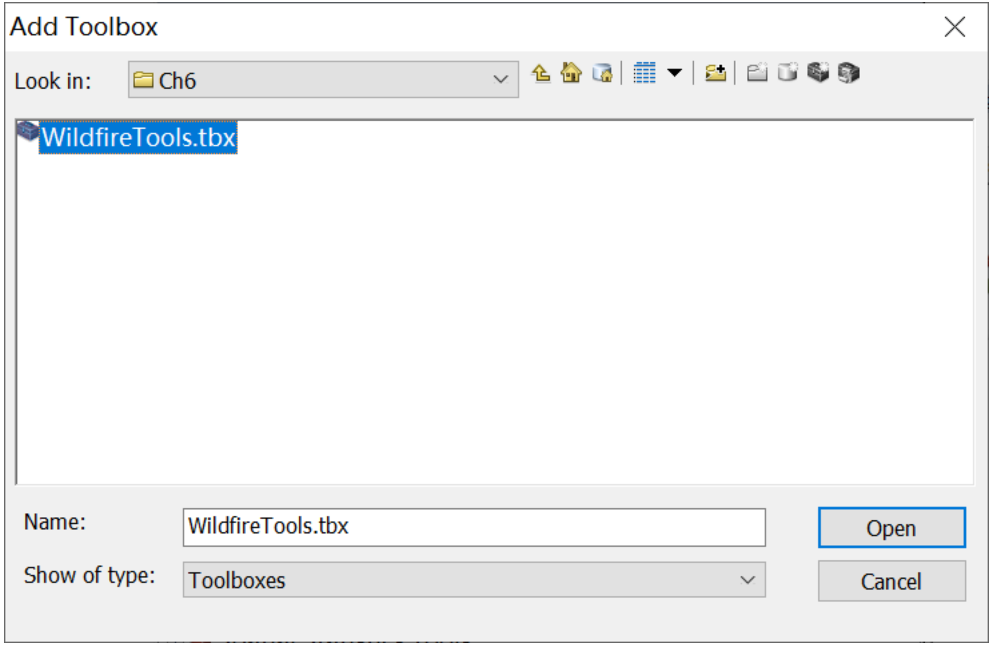
WildfireTools.tbx

Name:

Show of type: Toolboxes

9.1 创建自定义地理处理工具

4. 选择“WildfireTools.tbx”并单击“Open”按钮打开工具箱，此时工具箱添加到ArcToolbox中。



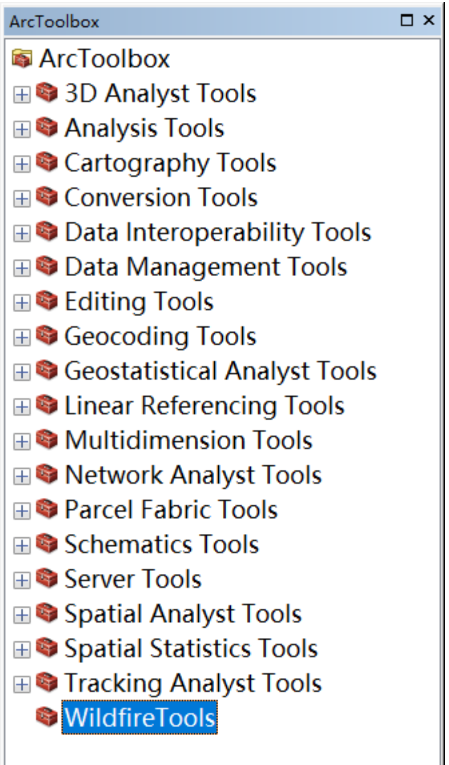
Add Toolbox

Look in: Ch6

WildfireTools.tbx

Name: WildfireTools.tbx

Show of type: Toolboxes



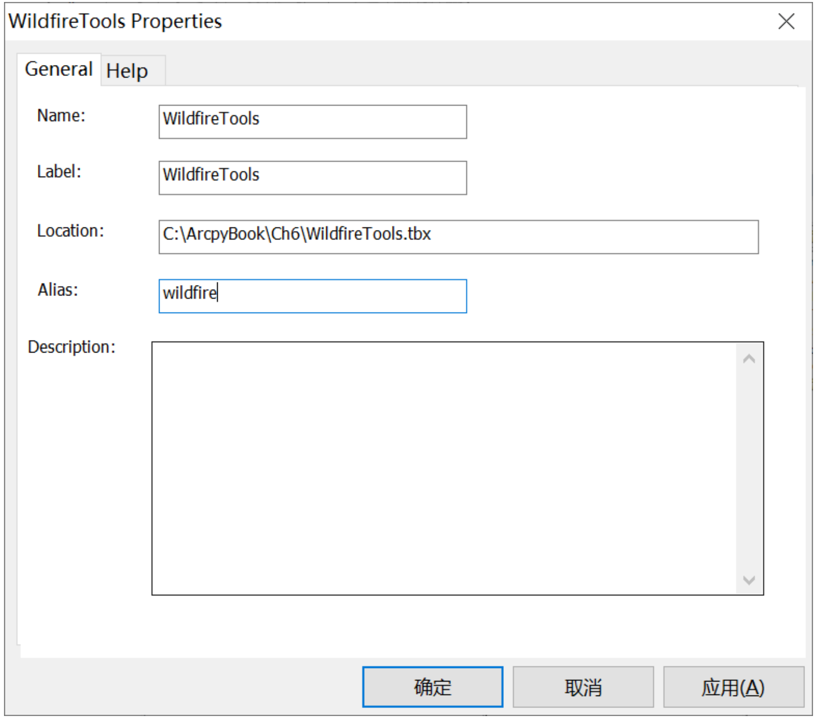
ArcToolbox

- ArcToolbox
  - 3D Analyst Tools
  - Analysis Tools
  - Cartography Tools
  - Conversion Tools
  - Data Interoperability Tools
  - Data Management Tools
  - Editing Tools
  - Geocoding Tools
  - Geostatistical Analyst Tools
  - Linear Referencing Tools
  - Multidimension Tools
  - Network Analyst Tools
  - Parcel Fabric Tools
  - Schematics Tools
  - Server Tools
  - Spatial Analyst Tools
  - Spatial Statistics Tools
  - Tracking Analyst Tools
  - WildfireTools

9.1 创建自定义地理处理工具

5. 为工具箱指定别名。添加WildfireTools的别名为“wildfire”

工具箱的别名用来定义工具的唯一性，应尽量简短且不能包含任何特殊字符。


WildfireTools Properties

Name: WildfireTools

Label: WildfireTools

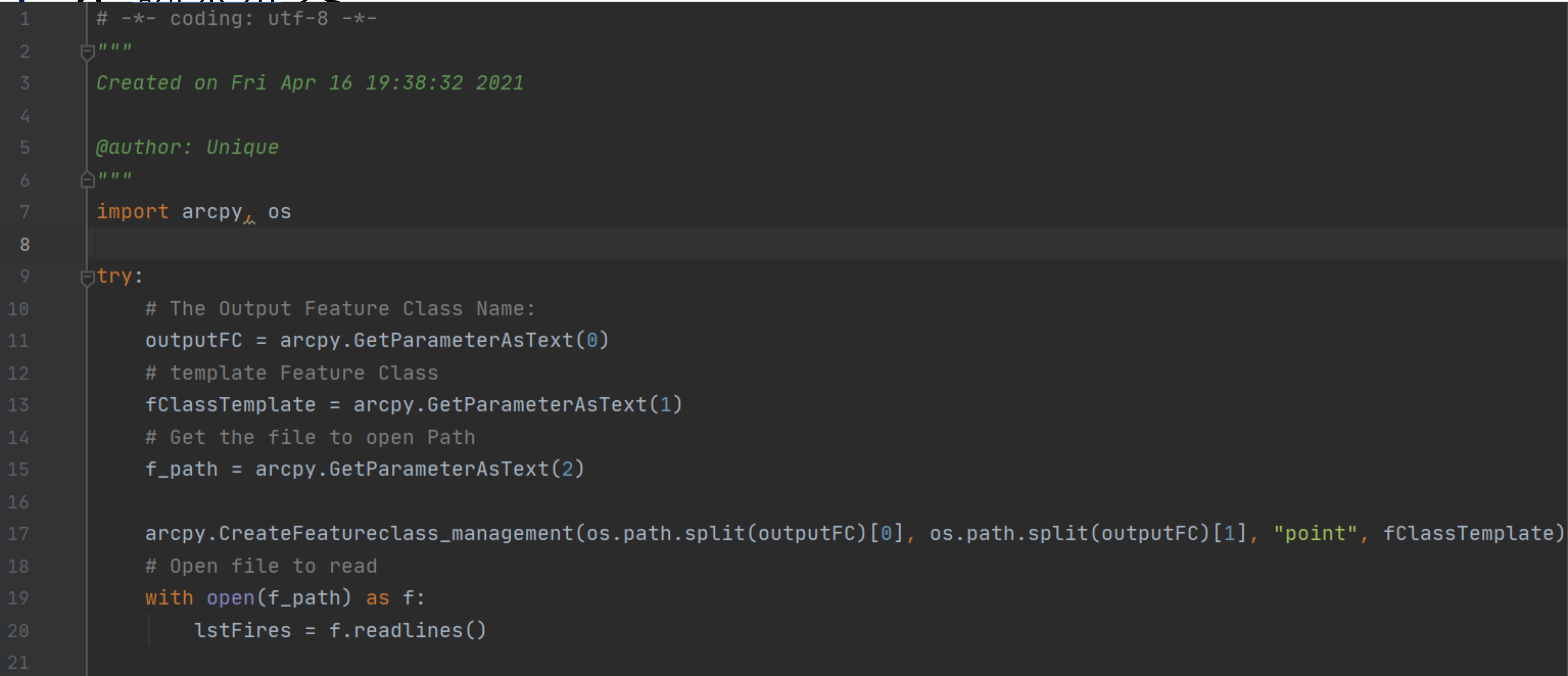
Location: C:\ArcpyBook\Ch6\WildfireTools.tbx

Alias: wildfire

Description:

9.1 创建自定义地理处理工具

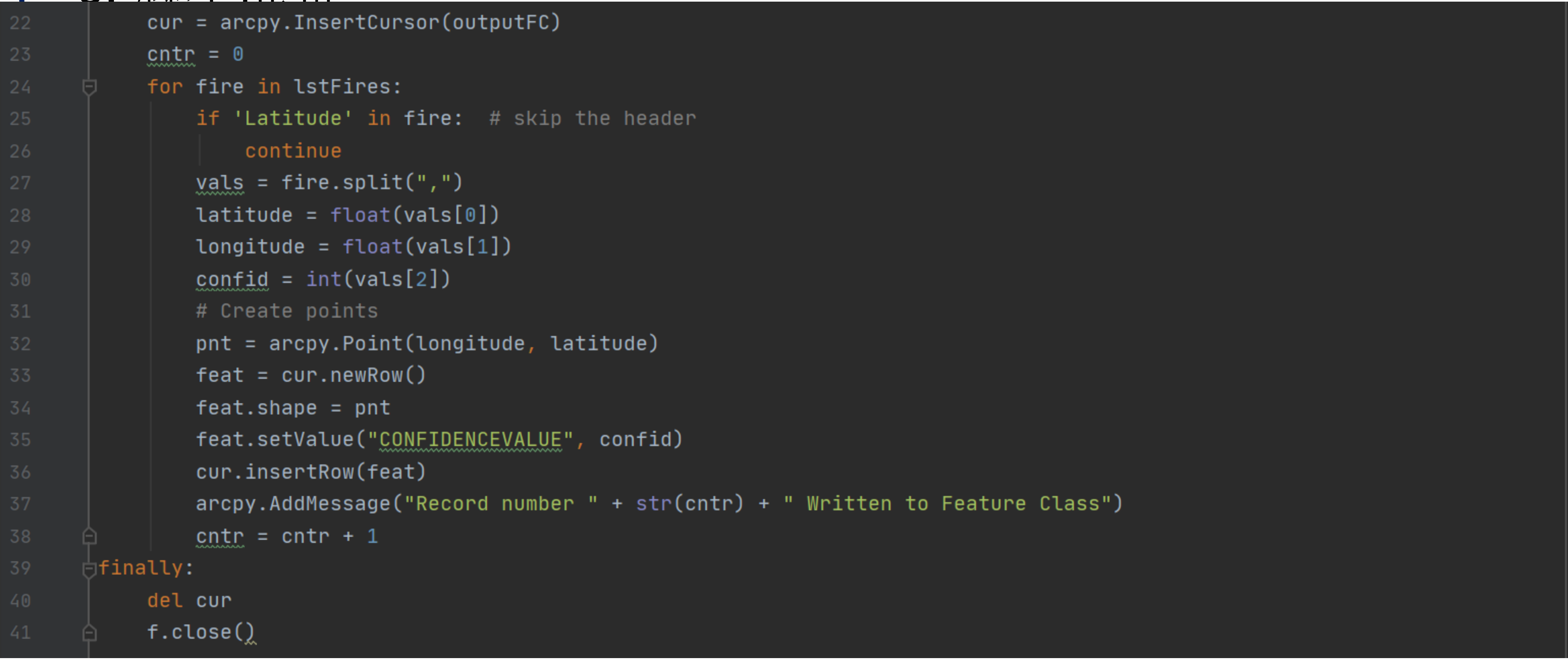
6. 脚本准备



```python
# -*- coding: utf-8 -*-
"""
Created on Fri Apr 16 19:38:32 2021

@author: Unique
"""
import arcpy, os

try:
    # The Output Feature Class Name:
    outputFC = arcpy.GetParameterAsText(0)
    # template Feature Class
    fClassTemplate = arcpy.GetParameterAsText(1)
    # Get the file to open Path
    f_path = arcpy.GetParameterAsText(2)

    arcpy.CreateFeatureclass_management(os.path.split(outputFC)[0], os.path.split(outputFC)[1], "point", fClassTemplate)
    # Open file to read
    with open(f_path) as f:
        lstFires = f.readlines()

```

9.1 创建自定义地理处理工具

6. 脚本准备

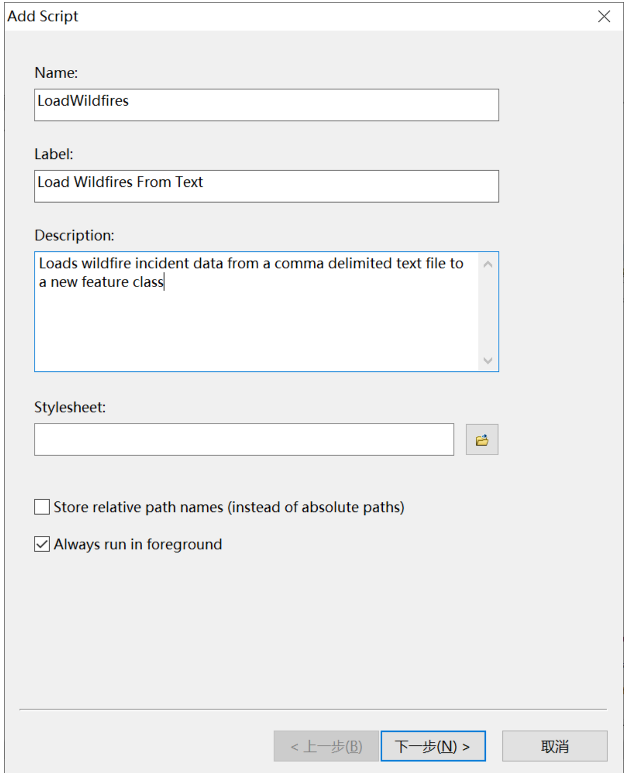


```
    cur = arcpy.InsertCursor(outputFC)
    cntr = 0
    for fire in lstFires:
        if 'Latitude' in fire:  # skip the header
            continue
        vals = fire.split(",")
        latitude = float(vals[0])
        longitude = float(vals[1])
        confid = int(vals[2])
        # Create points
        pnt = arcpy.Point(longitude, latitude)
        feat = cur.newRow()
        feat.shape = pnt
        feat.setValue("CONFIDENCEVALUE", confid)
        cur.insertRow(feat)
        arcpy.AddMessage("Record number " + str(cntr) + " Written to Feature Class")
        cntr = cntr + 1
finally:
    del cur
    f.close()
```

9.1 创建自定义地理处理工具

7. 为已经创建的Wildfire Tools工具箱添加脚本。右键---Add---Script

Name字段不能含有任何空格或特殊字符；
Label字段是显示在脚本旁边的名称；
描述性信息可用于说明脚本执行的操作。


Add Script

Name: LoadWildfires

Label: Load Wildfires From Text

Description: Loads wildfire incident data from a comma delimited text file to a new feature class

Stylesheet:

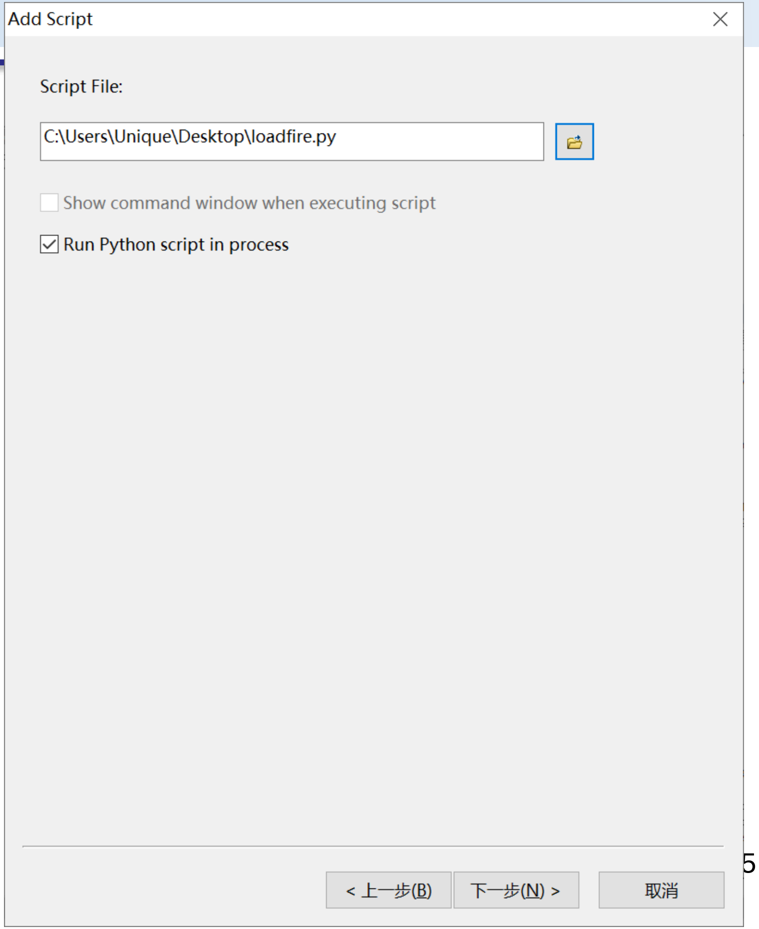
□ Store relative path names (instead of absolute paths)

☑ Always run in foreground

9.1 创建自定义地理处理工具

8. 指定要连接的工具脚本。

确保“Run Python script in process”复选框勾选。在进程中运行Python脚本可提高脚本的性能



Add Script

Script File:

C:\Users\Unique\Desktop\loadfire.py

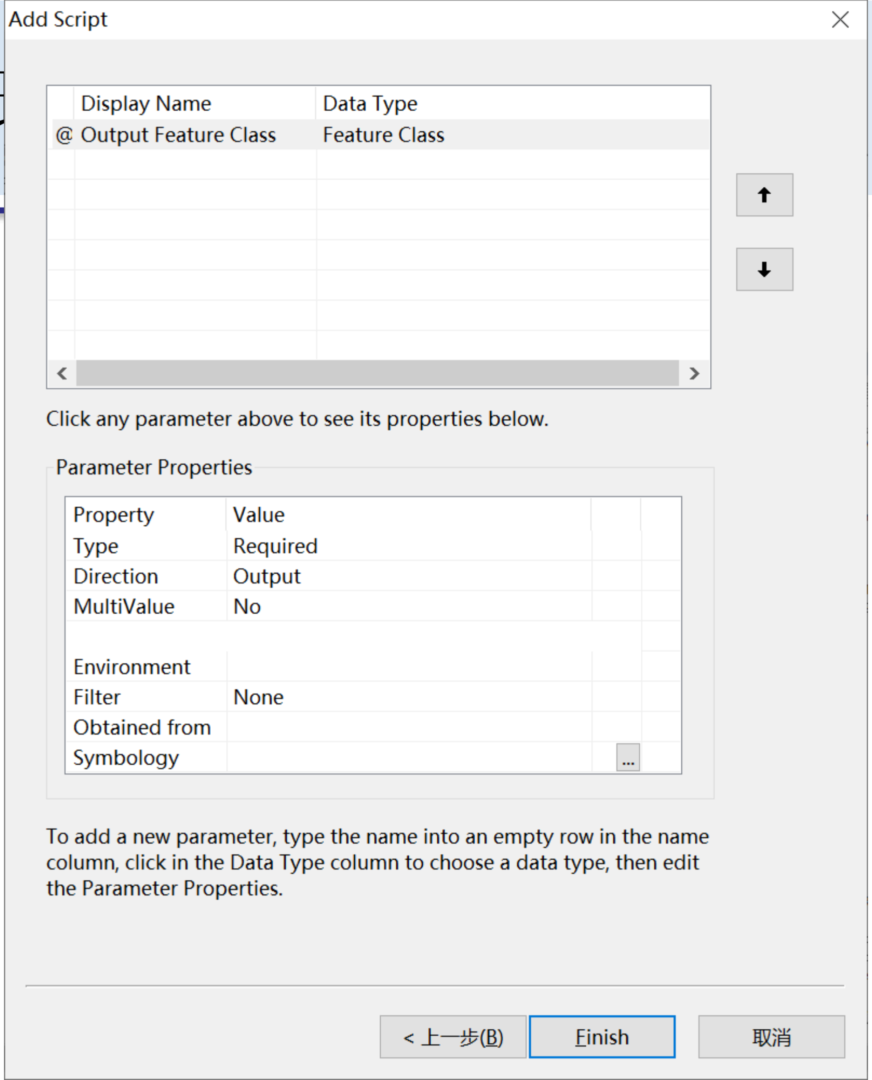
□ Show command window when executing script

☑ Run Python script in process

9.1 创建自定义地理处

9. 参数输入及设置。

窗口中输入的每个参数都相当于单独调用一次GetParameterAsText()方法



Add Script

	Display Name	Data Type
@	Output Feature Class	Feature Class

Click any parameter above to see its properties below.

Parameter Properties

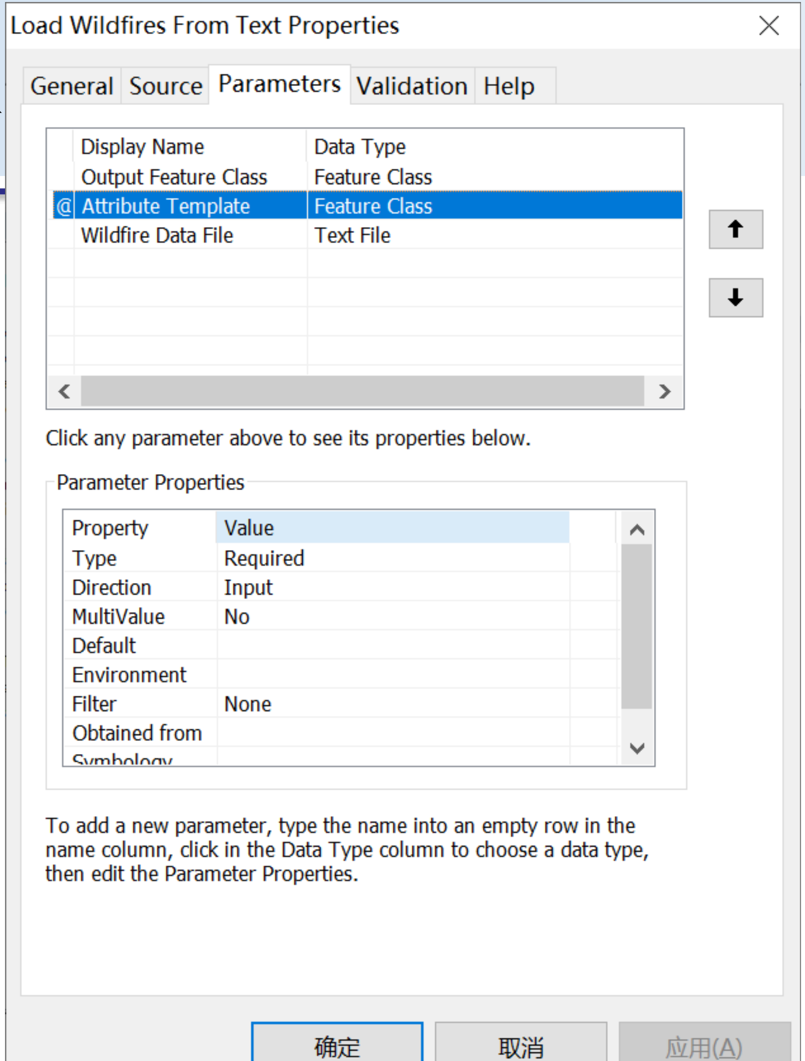
Property	Value
Type	Required
Direction	Output
MultiValue	No
Environment	
Filter	None
Obtained from	
Symbology	

To add a new parameter, type the name into an empty row in the name column, click in the Data Type column to choose a data type, then edit the Parameter Properties.

9.1 创建自定义地理处理工

9. 参数输入及设置。

窗口中输入的每个参数都相当于单独调用一次GetParameterAsText()方法



Load Wildfires From Text Properties

	Display Name	Data Type
	Output Feature Class	Feature Class
@	Attribute Template	Feature Class
	Wildfire Data File	Text File

Click any parameter above to see its properties below.

Parameter Properties

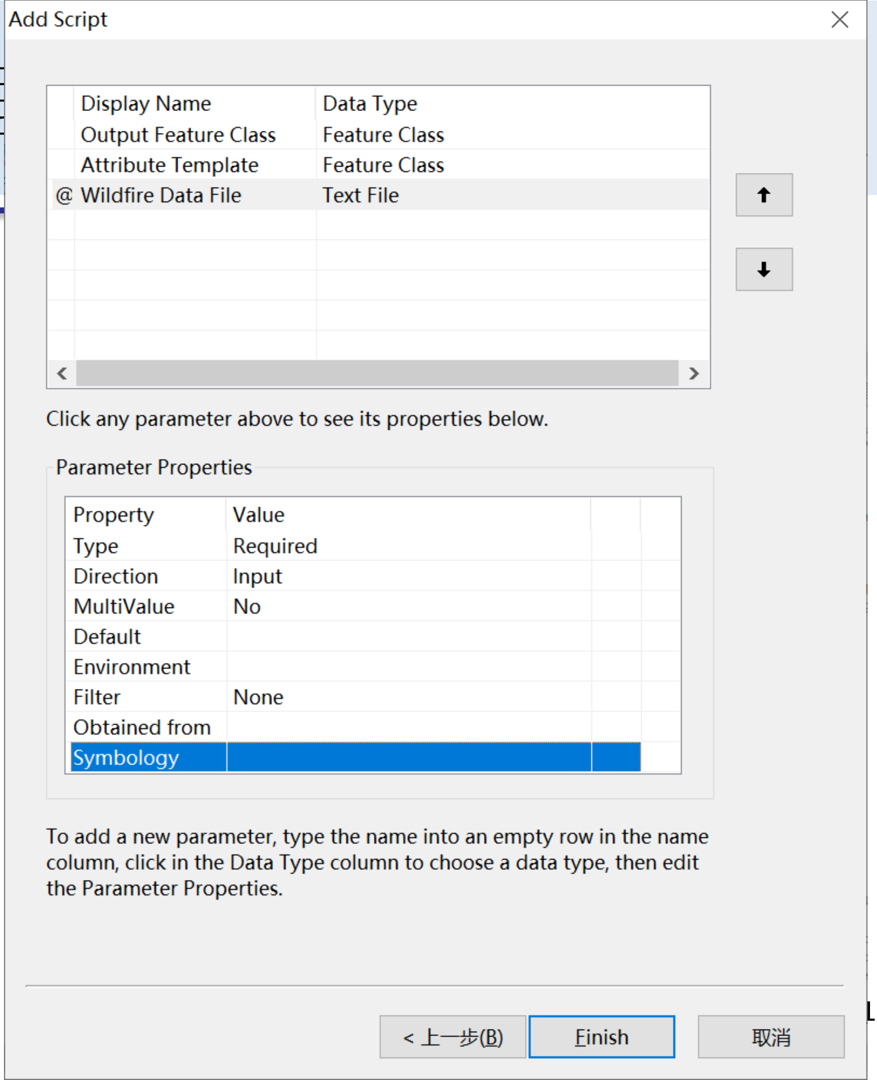
Property	Value
Type	Required
Direction	Input
MultiValue	No
Default	
Environment	
Filter	None
Obtained from	
Symbology	

To add a new parameter, type the name into an empty row in the name column, click in the Data Type column to choose a data type, then edit the Parameter Properties.

9.1 创建自定义地理处

9. 参数输入及设置。

窗口中输入的每个参数都相当于单独调用一次GetParameterAsText()方法



Add Script

	Display Name	Data Type
	Output Feature Class	Feature Class
	Attribute Template	Feature Class
@	Wildfire Data File	Text File

Click any parameter above to see its properties below.

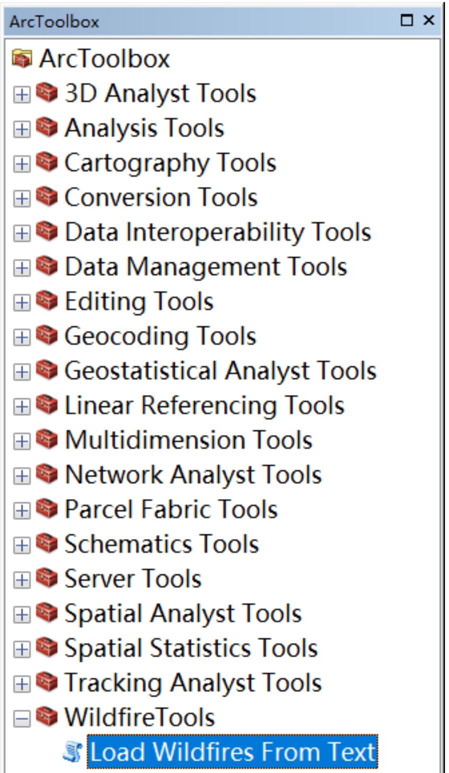
Parameter Properties

Property	Value
Type	Required
Direction	Input
MultiValue	No
Default	
Environment	
Filter	None
Obtained from	
Symbology	

To add a new parameter, type the name into an empty row in the name column, click in the Data Type column to choose a data type, then edit the Parameter Properties.

9.1 创建自定义地理处理工具

10. 新的脚本工具被添加到Wildfire Tools工具箱中。

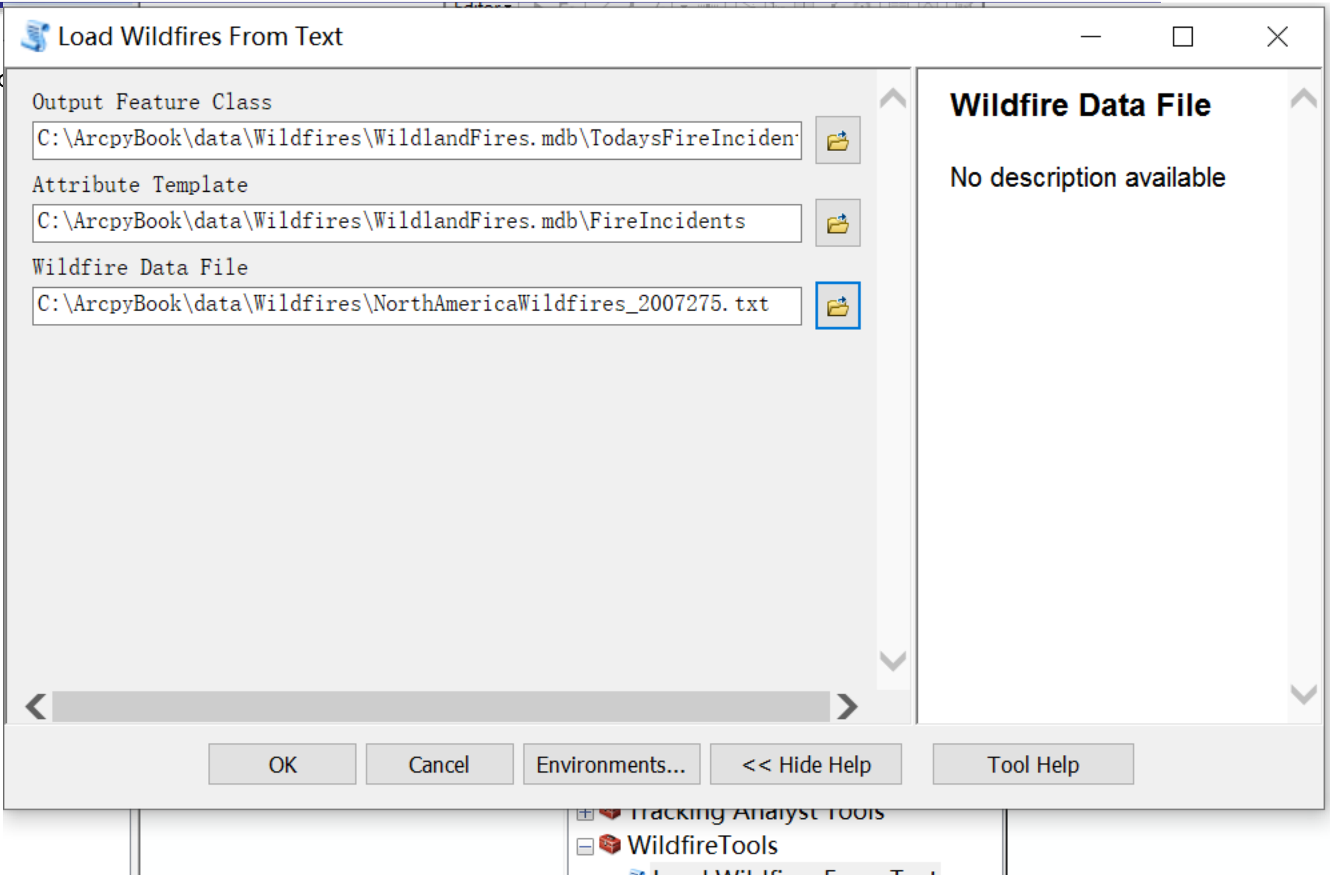


ArcToolbox

- ArcToolbox
  - 3D Analyst Tools
  - Analysis Tools
  - Cartography Tools
  - Conversion Tools
  - Data Interoperability Tools
  - Data Management Tools
  - Editing Tools
  - Geocoding Tools
  - Geostatistical Analyst Tools
  - Linear Referencing Tools
  - Multidimension Tools
  - Network Analyst Tools
  - Parcel Fabric Tools
  - Schematics Tools
  - Server Tools
  - Spatial Analyst Tools
  - Spatial Statistics Tools
  - Tracking Analyst Tools
  - WildfireTools
    - Load Wildfires From Text

9.1 创建自定义地理处理工具

11. 脚本工具测试。



Output Feature Class

C:\ArcpyBook\data\Wildfires\WildlandFires.mdb\TodaysFireInciden

Attribute Template

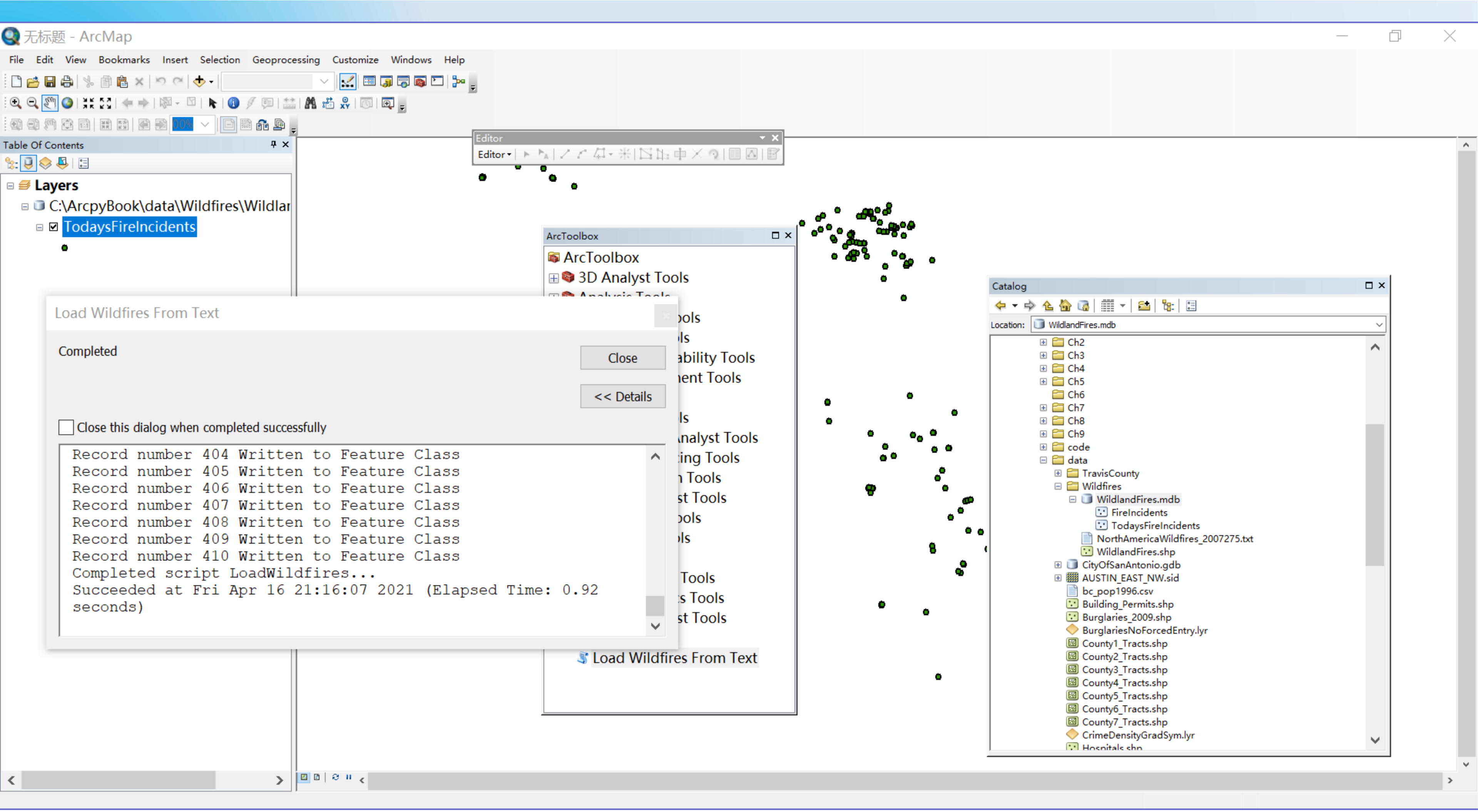
C:\ArcpyBook\data\Wildfires\WildlandFires.mdb\FireIncidents

Wildfire Data File

C:\ArcpyBook\data\Wildfires\NorthAmericaWildfires_2007275.txt

Wildfire Data File

No description available



Load Wildfires From Text

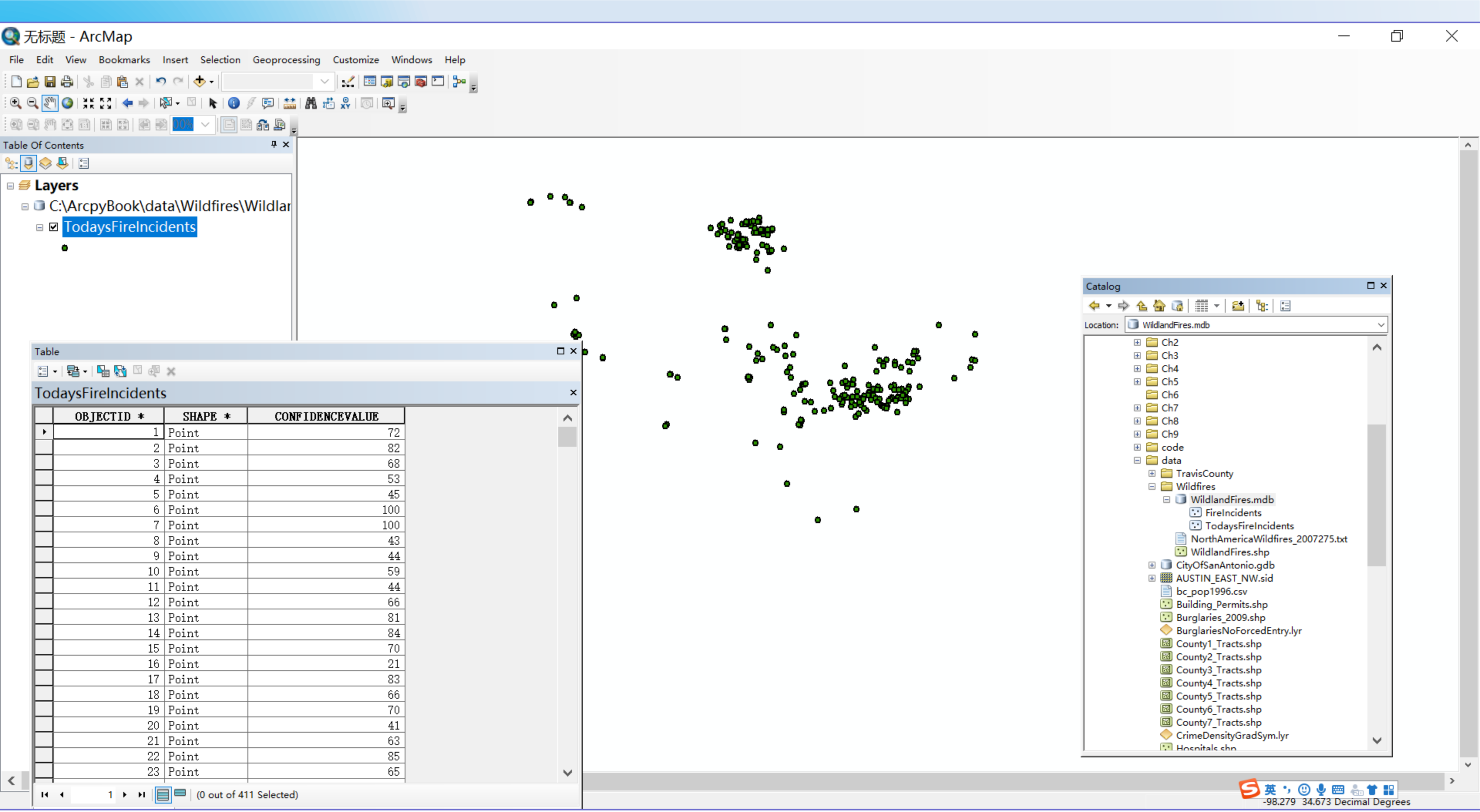
Completed

Close

<< Details

□ Close this dialog when completed successfully

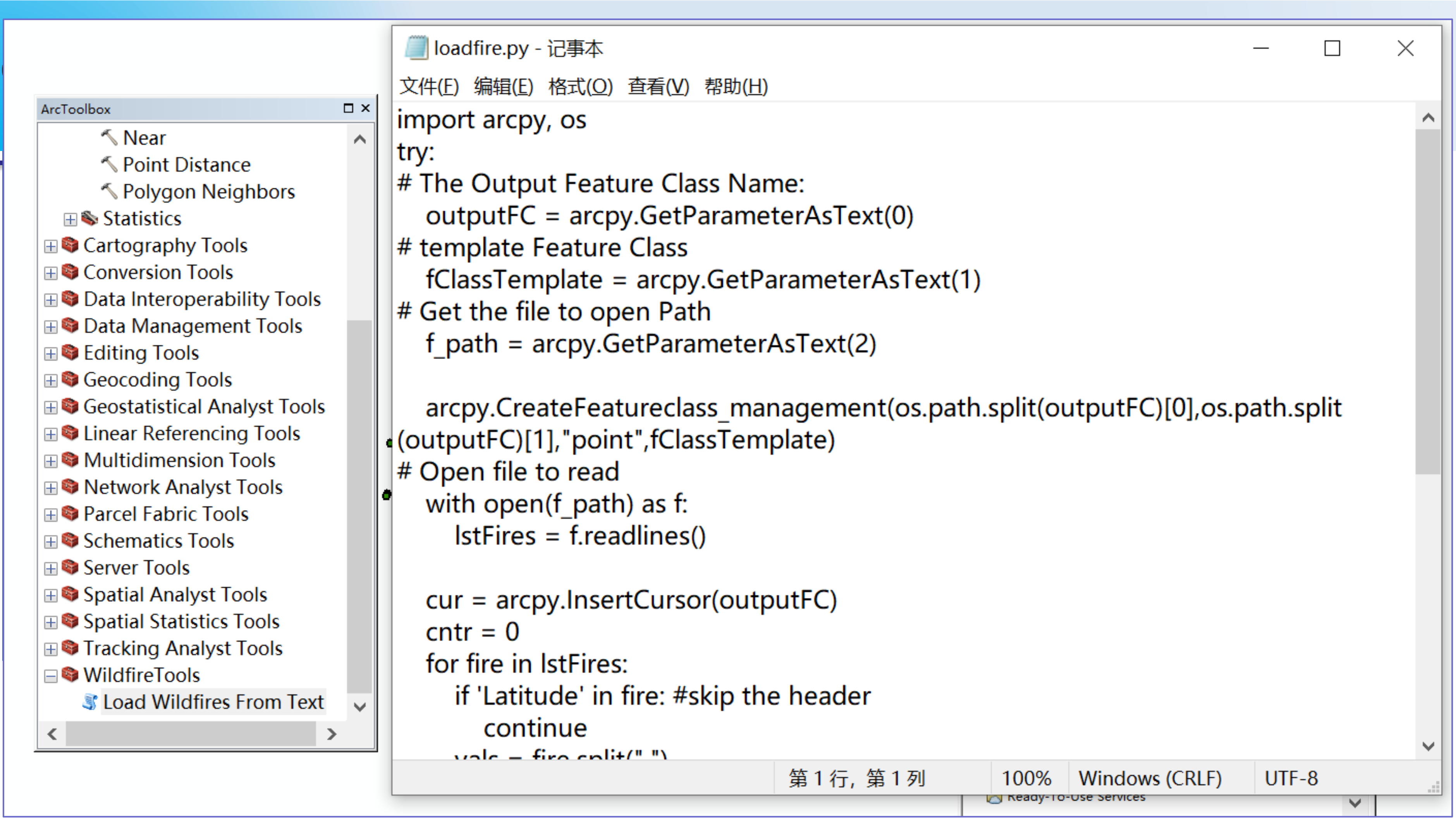
```
Record number 404 Written to Feature Class
Record number 405 Written to Feature Class
Record number 406 Written to Feature Class
Record number 407 Written to Feature Class
Record number 408 Written to Feature Class
Record number 409 Written to Feature Class
Record number 410 Written to Feature Class
Completed script LoadWildfires...
Succeeded at Fri Apr 16 21:16:07 2021 (Elapsed Time: 0.92
seconds)
```



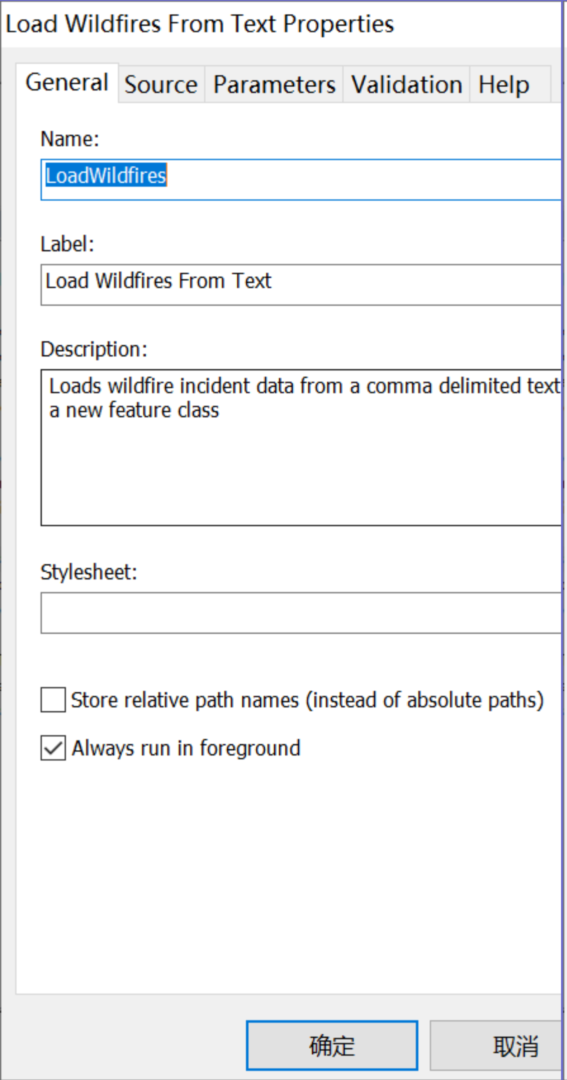
TodaysFireIncidents

OBJECTID *	SHAPE *	CONFIDENCEVALUE
1	Point	72
2	Point	82
3	Point	68
4	Point	53
5	Point	45
6	Point	100
7	Point	100
8	Point	43
9	Point	44
10	Point	59
11	Point	44
12	Point	66
13	Point	81
14	Point	84
15	Point	70
16	Point	21
17	Point	83
18	Point	66
19	Point	70
20	Point	41
21	Point	63
22	Point	85
23	Point	65

1 (0 out of 411 Selected)



```
import arcpy, os
try:
# The Output Feature Class Name:
    outputFC = arcpy.GetParameterAsText(0)
# template Feature Class
    fClassTemplate = arcpy.GetParameterAsText(1)
# Get the file to open Path
    f_path = arcpy.GetParameterAsText(2)

    arcpy.CreateFeatureclass_management(os.path.split(outputFC)[0],os.path.split
(outputFC)[1],"point",fClassTemplate)
# Open file to read
    with open(f_path) as f:
        lstFires = f.readlines()

    cur = arcpy.InsertCursor(outputFC)
    cntr = 0
    for fire in lstFires:
        if 'Latitude' in fire: #skip the header
            continue
        vals = fire.split(",")
```



Name:
LoadWildfires

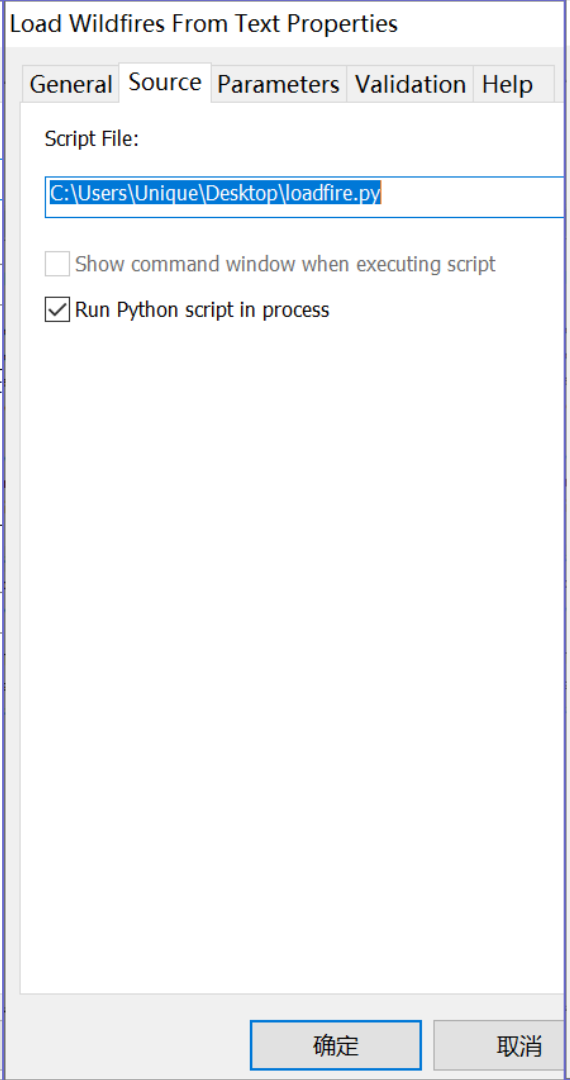
Label:
Load Wildfires From Text

Description:
Loads wildfire incident data from a comma delimited text
a new feature class

Stylesheet:

□ Store relative path names (instead of absolute paths)

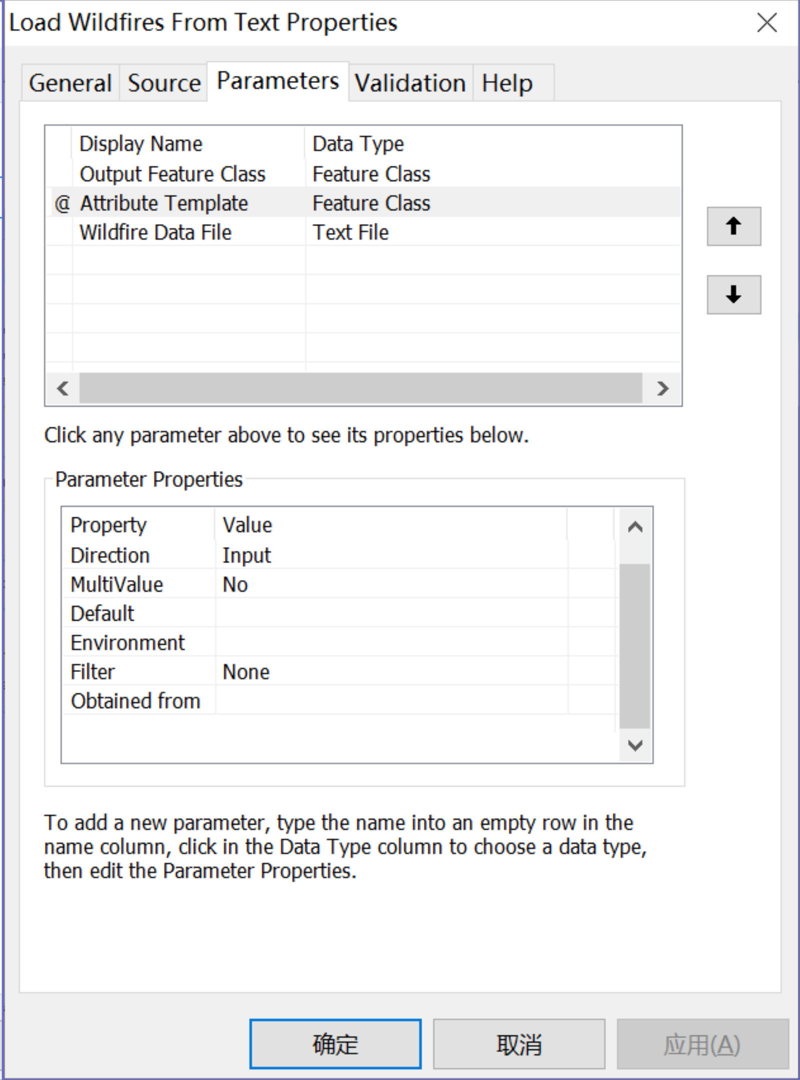
☑ Always run in foreground



Script File:
C:\Users\Unique\Desktop\loadfire.py

□ Show command window when executing script

☑ Run Python script in process



	Display Name	Data Type
	Output Feature Class	Feature Class
@	Attribute Template	Feature Class
	Wildfire Data File	Text File

Click any parameter above to see its properties below.

Parameter Properties

Property	Value
Direction	Input
MultiValue	No
Default	
Environment	
Filter	None
Obtained from	

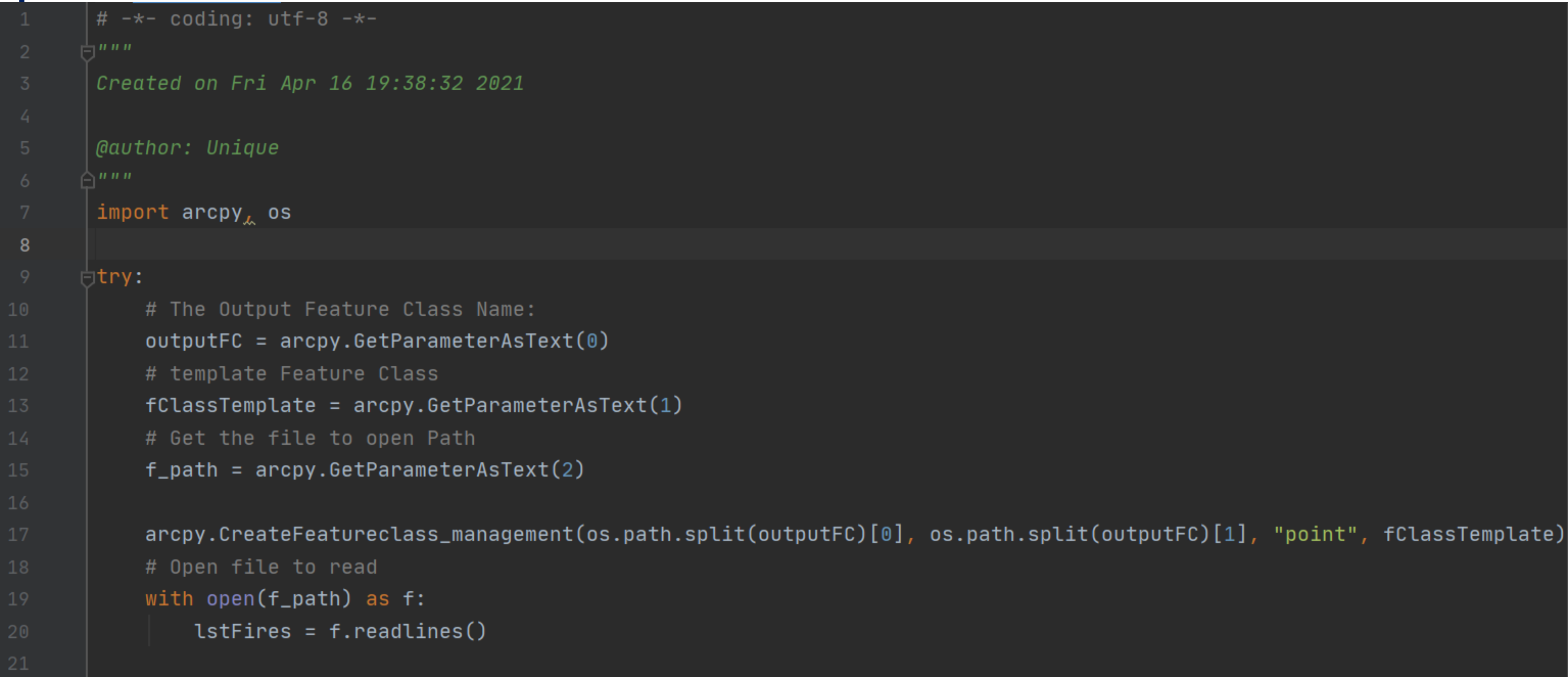
To add a new parameter, type the name into an empty row in the name column, click in the Data Type column to choose a data type, then edit the Parameter Properties.

9.1 创建自定义地理处理工具

几乎所有的脚本工具都有参数，并且这些参数的值要在工具对话框中输入。当执行工具时，参数值会传递给脚本，脚本读取这些值，然后进行工作。Python脚本可以接受参数的输入，参数的输入使脚本成为动态的。

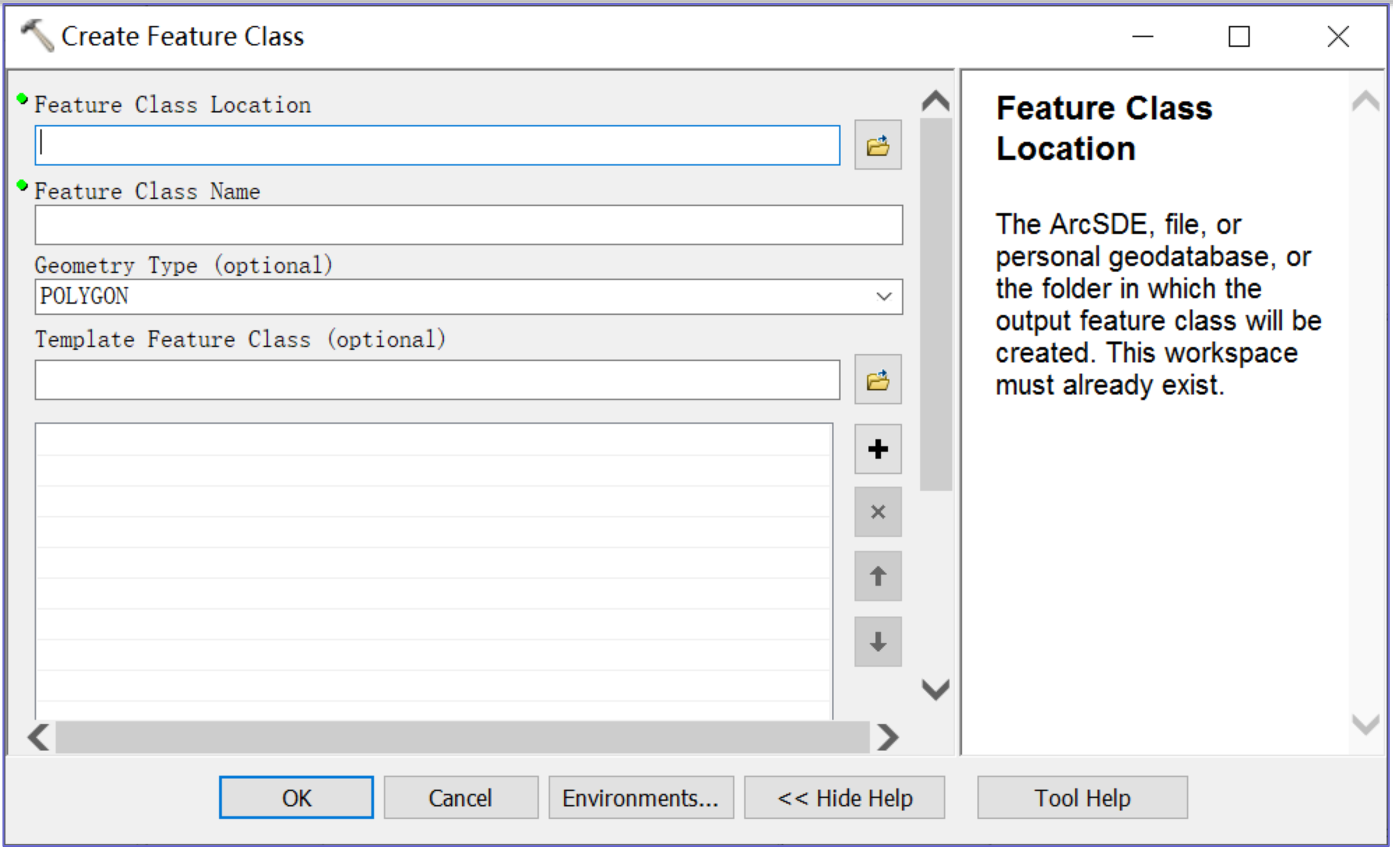
GetParameterAsText()方法用来捕获输入的参数，它的索引值从0开始，即第1个参数的索引值为0，并且每个连续参数的索引值按1递增。

9.2 CreateFeatureclass_management()函数



```
# -*- coding: utf-8 -*-
"""
Created on Fri Apr 16 19:38:32 2021

@author: Unique
"""
import arcpy, os

try:
    # The Output Feature Class Name:
    outputFC = arcpy.GetParameterAsText(0)
    # template Feature Class
    fClassTemplate = arcpy.GetParameterAsText(1)
    # Get the file to open Path
    f_path = arcpy.GetParameterAsText(2)

    arcpy.CreateFeatureclass_management(os.path.split(outputFC)[0], os.path.split(outputFC)[1], "point", fClassTemplate)
    # Open file to read
    with open(f_path) as f:
        lstFires = f.readlines()

```

9.2 CreateFeatureclass_management()函数



Create Feature Class

Feature Class Location

Feature Class Name

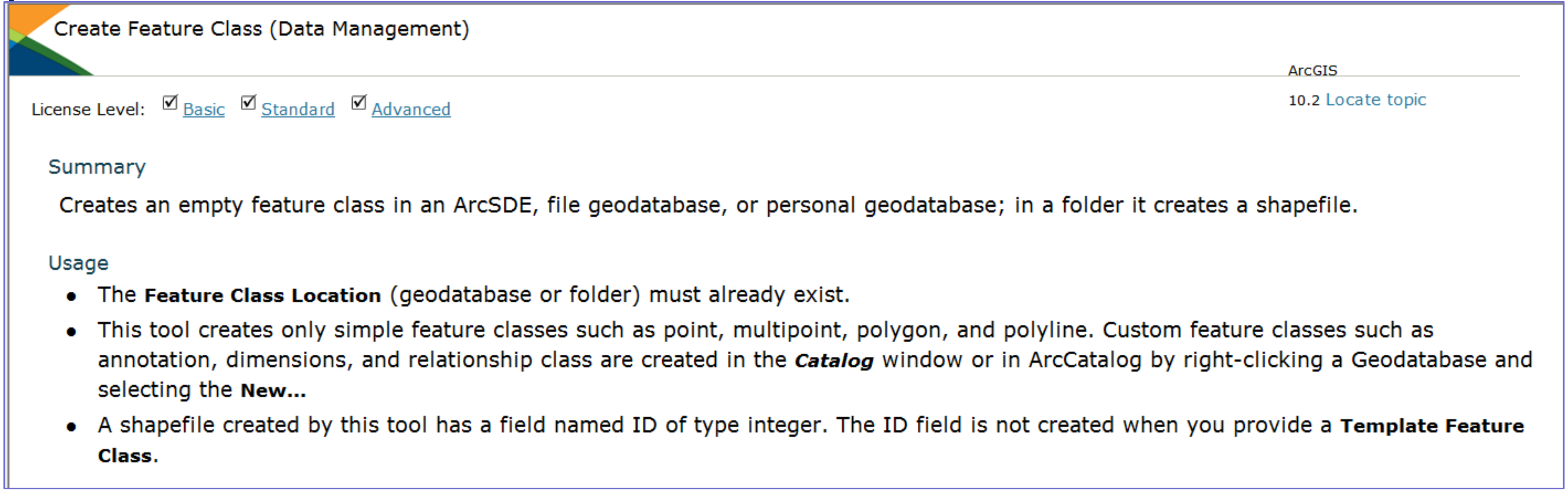
Geometry Type (optional)
POLYGON

Template Feature Class (optional)

Feature Class Location

The ArcSDE, file, or personal geodatabase, or the folder in which the output feature class will be created. This workspace must already exist.

9.2 CreateFeatureclass_management()函数



Create Feature Class (Data Management)

ArcGIS 10.2 Locate topic

License Level: ☑ Basic ☑ Standard ☑ Advanced

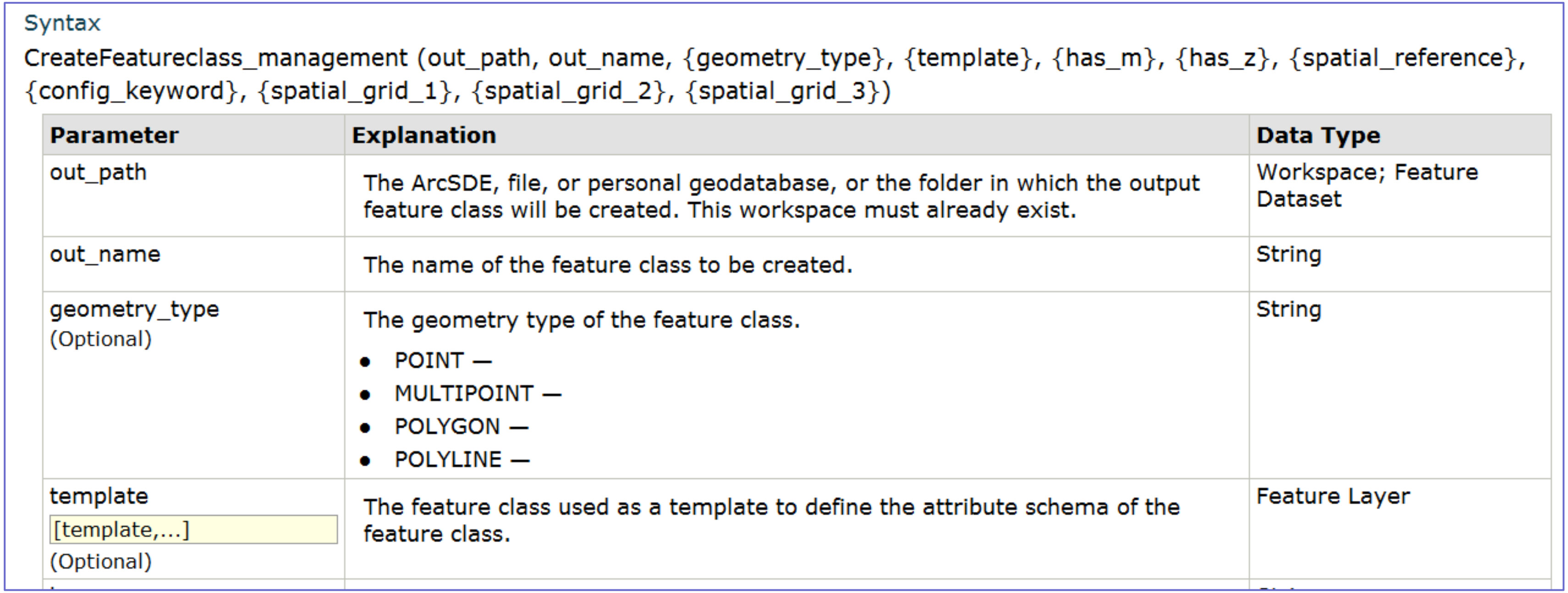
Summary

Creates an empty feature class in an ArcSDE, file geodatabase, or personal geodatabase; in a folder it creates a shapefile.

Usage

- The **Feature Class Location** (geodatabase or folder) must already exist.
- This tool creates only simple feature classes such as point, multipoint, polygon, and polyline. Custom feature classes such as annotation, dimensions, and relationship class are created in the ***Catalog*** window or in ArcCatalog by right-clicking a Geodatabase and selecting the **New...**
- A shapefile created by this tool has a field named ID of type integer. The ID field is not created when you provide a **Template Feature Class**.

9.2 CreateFeatureclass_management()函数



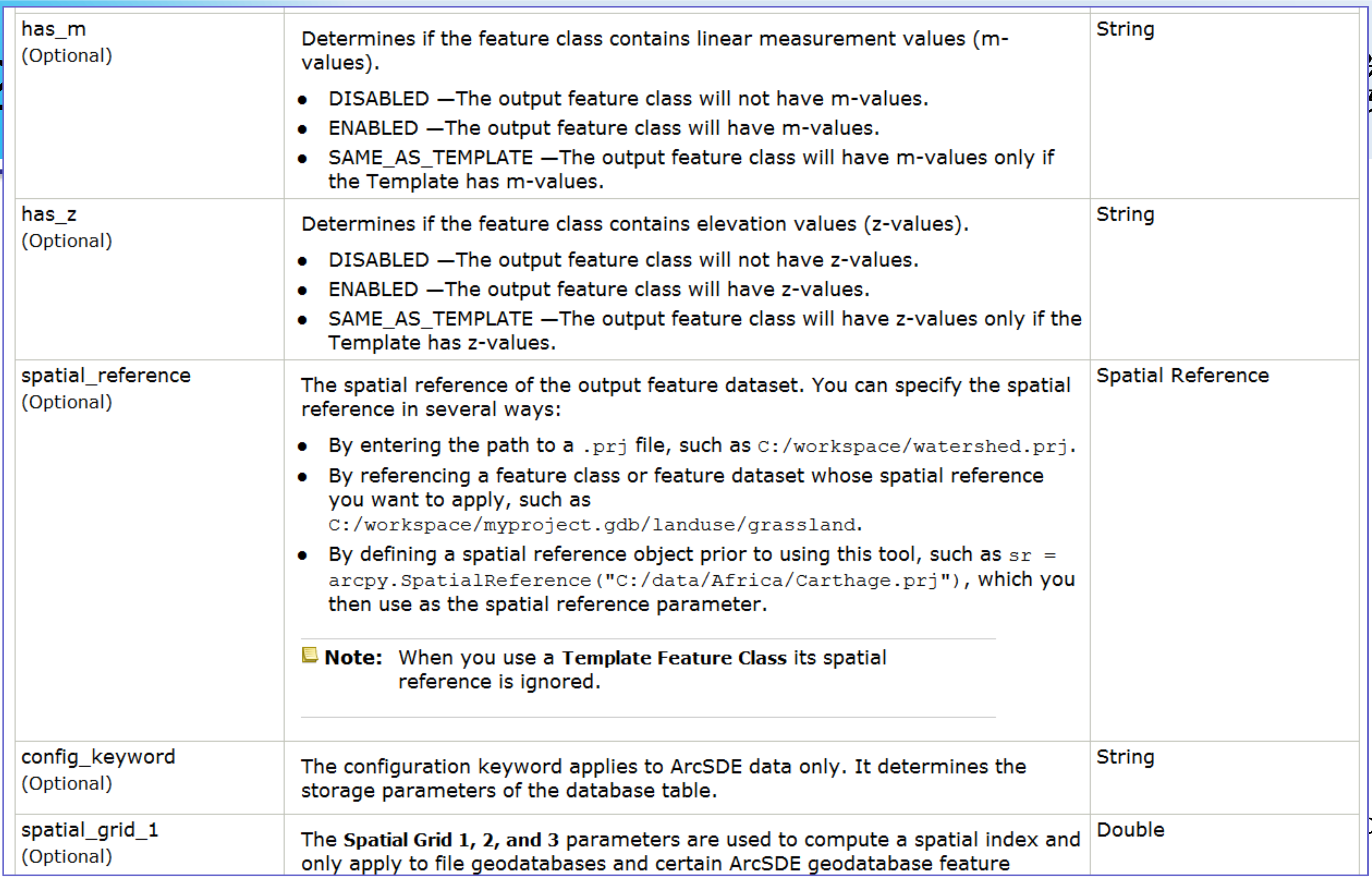
Syntax

CreateFeatureclass_management (out_path, out_name, {geometry_type}, {template}, {has_m}, {has_z}, {spatial_reference}, {config_keyword}, {spatial_grid_1}, {spatial_grid_2}, {spatial_grid_3})

Parameter	Explanation	Data Type
out_path	The ArcSDE, file, or personal geodatabase, or the folder in which the output feature class will be created. This workspace must already exist.	Workspace; Feature Dataset
out_name	The name of the feature class to be created.	String
geometry_type (Optional)	The geometry type of the feature class. • POINT — • MULTIPOINT — • POLYGON — • POLYLINE —	String
template [template,...] (Optional)	The feature class used as a template to define the attribute schema of the feature class.	Feature Layer

9.2

数



has_m (Optional)	Determines if the feature class contains linear measurement values (m-values). • DISABLED —The output feature class will not have m-values. • ENABLED —The output feature class will have m-values. • SAME_AS_TEMPLATE —The output feature class will have m-values only if the Template has m-values.	String
has_z (Optional)	Determines if the feature class contains elevation values (z-values). • DISABLED —The output feature class will not have z-values. • ENABLED —The output feature class will have z-values. • SAME_AS_TEMPLATE —The output feature class will have z-values only if the Template has z-values.	String
spatial_reference (Optional)	The spatial reference of the output feature dataset. You can specify the spatial reference in several ways: • By entering the path to a `.prj` file, such as `C:/workspace/watershed.prj`. • By referencing a feature class or feature dataset whose spatial reference you want to apply, such as `C:/workspace/myproject.gdb/landuse/grassland`. • By defining a spatial reference object prior to using this tool, such as `sr = arcpy.SpatialReference("C:/data/Africa/Carthage.prj")`, which you then use as the spatial reference parameter. **Note:** When you use a **Template Feature Class** its spatial reference is ignored.	Spatial Reference
config_keyword (Optional)	The configuration keyword applies to ArcSDE data only. It determines the storage parameters of the database table.	String
spatial_grid_1 (Optional)	The **Spatial Grid 1, 2, and 3** parameters are used to compute a spatial index and only apply to file geodatabases and certain ArcSDE geodatabase feature	Double

The End