AddTimeStringField

Title AddTimeStringField

Description

This tool is a simple strf time script intended to add a new text field with a formatted string based on the format string input into it. This tool uses pandas to convert and add the new unique field using the arcpy.da extend table function.

Usage

This tool is intended to provide an easy way to created formatted string fields using the strftime function.

Syntax

AddTimeStringField (Input_Feature_Class, Date_Time_Field, New_Field_Name, Format_String)

Parameter	Explanation	Data Type
Input_Feature_Class	Dialog Reference Is the input feature class or table for which a new time field will be added. Python Reference Depends on https://docs.python.org/2/library/time.html#time.strftime.	Feature Layer
Date_Time_Field	Dialog Reference This is the ArcGIS date field that will be used to construct the datetime objects used in the created Pandas data frame. Python Reference Generally the fields are selected from the feature class to be converted into a numpy array, then into a pandas data frame, then back to structured numpy array to be joined based on the object ID. This tool assumes there is an object ID to use to join to.	Field
New_Field_Name	Dialog Reference This is the name of the new text field that will be added to the feature class and then populated by a new time string based on the format string. If the name already exists, then a unique one will be added. There is no python reference for this parameter.	String
Format_String	Dialog Reference The format string determines the output time format. Do not add quotes.	String

Check the documentation link here for details.

Directive Meaning Notes

%a Locale’s abbreviated weekday name.

%A Locale’s full weekday name.

%b Locale’s abbreviated month name.

%B Locale’s full month name.

%c Locale’s appropriate date and time representation.

%d Day of the month as a decimal number [01,31].

%H Hour (24-hour clock) as a decimal number [00,23].

%I Hour (12-hour clock) as a decimal number [01,12].

%j Day of the year as a decimal number [001,366].

%m Month as a decimal number [01,12].

%M Minute as a decimal number [00,59].

%p Locale’s equivalent of either AM or PM. (1)

%S Second as a decimal number [00,61]. (2)

%U Week number of the year (Sunday as the first day of the week) as a decimal number [00,53]. All days in a new year preceding the first Sunday are considered to be in week 0. (3)

%w Weekday as a decimal number [0(Sunday),6].

%W Week number of the year (Monday as the first day of the week) as a decimal number [00,53]. All days in a new year preceding the first Monday are considered to be in week 0. (3)

%x Locale’s appropriate date representation.

%X Locale’s appropriate time representation.

%y Year without century as a decimal number [00,99].

%Y Year with century as a decimal number.

%Z Time zone name (no characters if no time zone exists).

%% A literal '%' character.

Notes:

When used with the strptime() function, the %p directive only affects the output hour field if the %I directive is used to parse the hour.

The range really is 0 to 61; this accounts for leap seconds and the (very rare) double leap seconds.

When used with the strptime() function, %U and %W are only used in calculations when the day of the week and the year are specified.

Taken from: http://strftime.org/

There is no python reference for this parameter.

Code Samples

There are no code samples for this tool.

Tags

Time, String, Format

Credits

David Wasserman

Use limitations

There are no access and use limitations for this item.

You are currently using the Item Description metadata style. Change your metadata style in the Options dialog box to see additional metadata content.