
Article

[Veerarajan Karu...](#) · Jul 12, 2022 10m read

Time zone conversion utility using embedded python

Date and Time is an important factors in our life. Because all applications are engaging based on Date/Time. But our world is split into multiple time zones. if our product is launched in the world, to maintain the history of events we are definitely required to convert all times to our local time or UTC (Coordinated Universal Time). As I know, many known programming languages of C#, JavaScript, Java, etc., provided the library to convert the date and time. i.e with a time zone name we can be able to convert without knowing the UTC offset.

TZ database doesn't exist in InterSystems products. In past, I have written the function in C# (C# has a library for this) and used it inside InterSystems as a wrapper class.

But now, IRIS is started to support Embedded python. So, we can use the timezone conversion from python in InterSystems easily.

Libraries required

1. datetime - Basic date and time type support - <https://docs.python.org/3/library/datetime.html>
2. pytz - tz database library - <https://pypi.org/project/pytz/>

How to install

Once install the IRIS, go to the InterSystems bin folder (\Intersystems\IRIS\bin) and execute

Windows :

```
irisip install --target <source>\Intersystems\IRIS\mgr\python datetime
```

```
irisip install --target <source>\Intersystems\IRIS\mgr\python pytz
```

Linux :

```
pip3 install --target /InterSystems/IRIS/mgr/python datetime
```

```
pip3 install --target /InterSystems/IRIS/mgr/python pytz
```

How to handle library in IRIS

```
Class Utility.TimezoneConversion
{
  /// Convert the date time and return as DateTime object
  /// CTime - Input date and time
  /// ToTimezone - Which format we need to change the date and time
  ClassMethod Convert(CTime As %String, ToTimezone As %String) As %String
  {
    Set ret = $$$OK
    try {
      #;datetime library to calculate the date or time
      Set DateTime = ##class(%SYS.Python).Import("datetime")
      #;timezone library to convert the one timezone to another timezone
      Set Pytz = ##class(%SYS.Python).Import("pytz")
      #; timezone apply
      Set DateTimeConverted = DateTime.datetime(+$(CTime,":",1),+$(CTime,":",2),+$( $(CTime,":",3), " ",1),+$( $(CTime,":",3), " ",2),+$(CTime,":",4),+$(CTime,":",5)).now(
        Pytz.timezone(ToTimezone))
      Set ret = DateTime.datetime.now(Pytz.timezone(ToTimezone))
    }
    catch(ex) {
      Set ret = ex.AsStatus()
    }
    Quit ret
  }
  /// Convert date time into one timezone format to another timezone format
  /// Input Format - YYYY:MM:DD HH:MM:SS
  ClassMethod DisplayConversion() As %Status
  {
    Set ToFormat = "%Y:%m:%d %H:%M:%S"
```

Time zone conversion utility using embedded python

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
Set CTime = "2022:07:12 12:00:00"

Set ToTimezone = "America/Mexico_City"

Set ret = ..Convert(CTime,ToTimezone)

If $IsObject(ret){

Write "InputDateTime:"_CTime,!

Write "ConvertedDateTime:"_ret.strftime(ToFormat),!

Write "Timezone Name:"_ret.tzname(),!

Write "Timezone detail:"_ToTimezone,!

}Else{

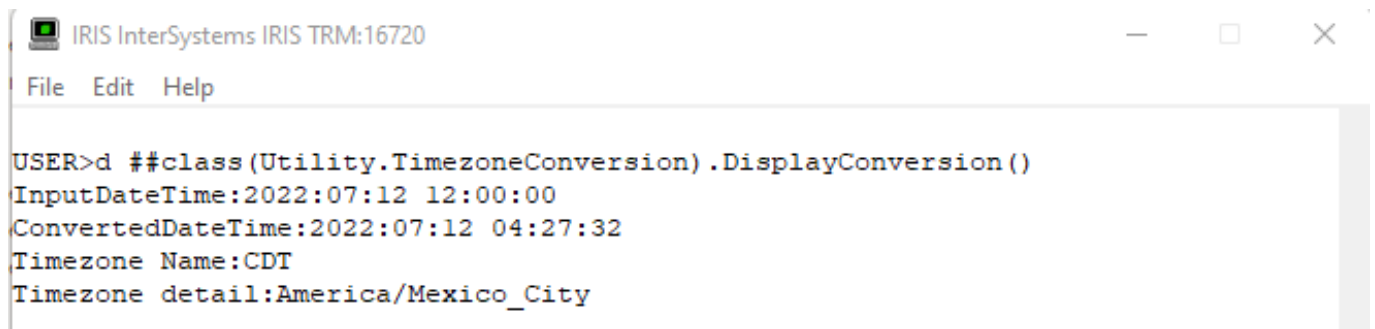
Write "Error:"_ret,!

}

}

}
```

Output

A screenshot of a terminal window titled "IRIS InterSystems IRIS TRM:16720". The window has a menu bar with "File", "Edit", and "Help". The terminal output shows the execution of a class method:

```
USER>d ##class (Utility.TimezoneConversion).DisplayConversion()
InputDateTime:2022:07:12 12:00:00
ConvertedDateTime:2022:07:12 04:27:32
Timezone Name:CDT
Timezone detail:America/Mexico_City
```

List of All Python Time zones

Spoiler

Performance

Performance-wise compared to the wrapper, embedded python provides a faster response and easy to implement.

[#Embedded Python](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

Source URL: <https://community.intersystems.com/post/time-zone-conversion-utility-using-embedded-python>
