



Create database copies in seconds and save up to 99% of disk space





Go

Log in :: Register :: Not logged in

Search

Home Tags Articles

Editorials
Stairways
Forums
Scripts
Videos
Blogs

QotD Books Ask SSC SQL Jobs Training

Authors
About us
Contact us
Newsletters

Write for us

Daily SQL Articles by email:

your@email.com

Sign up





Get the alerts & data you need to fix unexpected schema changes





Want more great articles like this? Sign up for fresh SQL Server knowledge delivered daily.



Using IronPython to talk to SQL Server

By Gerald Britton, 2016/06/13

IronPython? What's that?

IronPython is an implementation of the Python programming and scripting language built atop .NET. It was originally created by Jim Hugunin, then kept going by Microsoft until late 2010. Since then, it has become an open source project. Currently it is compatible with Python 2.7.

Why should I care?

Many students learn Python during introductory Computer Science courses (including my alma mater, the University of Toronto) before going on to do other things. Python remains true to its mantra of "easy to learn; hard to forget" and is easier to learn than many curly-brace programming languages. Adding .NET to the mix with IronPython captilizes on Python's main advantagfes while adding all the power of .NET. In particular, we can access SQL Server Management Objects (SMOs) using Python.

So, how do I get started?

Head on over to http://ironpython.net/download/ and download the latest copy of the language. You might also like to install Python Tools for Visual Studio, so you can use VS (2012 and up) to create Python projects and programs. PTVS is available via NuGet and the VS Extensions manager.

What can I do with it?

If you are using PTVS, create a new Python project and give it any name you like. If not, any old editor will do: just create a directory for your code.

For your first project, copy and paste this code into a file in the project called helpers.py

Related Articles

BLOG Python Distributions

2016/07/01

Python Distributions Python is free and open source software. A programmer could download the execu...

ARTICLE Changing Linked Server Properties

2016/06/27

A quick guide on changing the linked server properties and settings.

FORUM connection failed, check sql server registration properties

2009/04/02

connection failed, check sql server registration properties

Fetching Remote servers properties

2009/04/01

How to Fetch Remote servers properties

forum database diagram print on one page

2010/08/10

database diagram print

Tags

ironpython ptvs python tools smo sql server visual studio



```
import re
# Pattern to match for objects with no
pattern = re.compile('^<.*>$')
def get properties(obj):
    for member in filter(lambda x: not x.:
        try:
            attr = getattr(obj, member)
            if not callable(attr):
                if not pattern.match(str(a
                    yield member, attr
        except Exception as ex:
            # Discard attributes throwing
            continue
def print properties(obj):
    for prop, value in get_properties(obj)
              .. ... ... .
```

Next, in the same directory, create a file called smo.py. It should look like this:

```
from helpers import print_properties
import clr

clr.AddReference("Microsoft.SqlServer.SMO")
import Microsoft.SqlServer.Management.Smo a
# Local database
server = Smo.Server('(localdb)\mssqllocaldb
print server.ServerType
print_properties(server.Databases['master']
```

You can run this from the command line by changing to the directory you saved the files in and typing

```
ipy.exe smo.
```

(Note: You might have to specify the complete path to ipy.exe, dependint on you you installed it.)

You should see output like this:

```
ActiveConnections
: 16
This method or property is accessible only while working against a version earlier than SQL
Server 2012.
AnsiNullDefault
: False
AnsiNullsEnabled
: False
AnsiPaddingEnabled
: False
AnsiWarningsEnabled
: False
ArithmeticAbortEnabled
```

For your second try, add these lines to the program:

Stay up to date:

Daily newsletters with brand new articles, scripts, editorials and a Question of the Day help you keep on top of SQL Server.

your@email.com

Sign up

No thanks

```
clr.AddReference("Microsoft.SqlServer.Conne
import Microsoft.SqlServer.Management.Commo

my_server = 'mhknbn2kdz.database.windows.ne
my_database = 'AdventureWorks2012'
my_userid = 'sqlfamily'
my_password = 'sqlf@m1ly'

conn = Common.ServerConnection()

conn.ConnectionString = 'server={};database
server = Smo.Server(conn)
print server.ServerType

print_properties(server.Databases['Adventure])
```

This code should access the free, Azure-based copy of AdventureWorks and report on the basic database properties. When I do this, I get:

```
ActiveConnections: unknown property.
This method or property is accessible only while
working against a version earlier than SQL
Server 2012.
AnsiNullDefault
  : False
AnsiNullsEnabled
 : False
AnsiPaddingEnabled
 : False
AnsiWarningsEnabled
  : False
ArithmeticAbortEnabled
 : False
AutoClose
  : False
... and so on
```

What Now?

Now that you have some basic examples working, you can experiment with other properties. Try some of these:

```
for table in server.Databases['AdventureWor
    print table.Name

for table in db.Tables:
    if table.Name == 'Store':
        print table.Name
        print_properties(table)

print_properties(db.Tables[9])
```

Summary

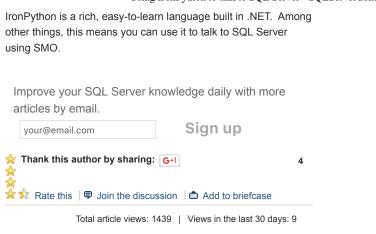
Stay up to date:

Daily newsletters with brand new articles, scripts, editorials and a Question of the Day help you keep on top of SQL Server.

your@email.com

Sign up

No thanks



Copyright © 2002-2016 Redgate. All Rights Reserved. Privacy Policy. Terms of Use. Report Abuse.

Stay up to date:

Daily newsletters with brand new articles, scripts, editorials and a Question of the Day help you keep on top of SQL Server.

your@email.com

Sign up

No thanks