

Pensieve header: Interactive assembly of my Academic Pensieve.

```
In[*]:= SetDirectory["C:\\drorbn\\AcademicPensieve\\Projects\\AcademicPensieve"];
Get["AssemblyPrograms.m"];
Get["AcademicPensieveStartup.m"];
Print["Running BatchAssembly with {PensieveDirectory, PensieveURL}=",
      {PensieveDirectory, PensieveURL}];
BeginPackage["Pensieve`"];
(*Begin["`Private`"];*)
```

```
Running BatchAssembly with {PensieveDirectory, PensieveURL}=
{C:\\drorbn\\AcademicPensieve, http://drorbn.net/AcademicPensieve/}
```

```
In[*]:= InitializeTemplates []
```

```
In[*]:= PensieveAssemble["nb"]
```

nb: People\Scherich\nb\TurboGassner.pdf

... The turbo Gassner representation. Continues pensieve://2016-06/.

```
In[*]:= Last /@ FileNameSplit /@ PensieveAssemble["one"]
```

NETLink`NET: No method named OpenHierarchy exists for the given COM object.

NETLink`NET: No method named GetHierarchy exists for the given COM object.

ImportString: First argument Pensieve`Private`XMLString\$1806 is not a string.

Out[ ]=

```
{2001-06.one, 2007-02.one, 2007-11.one, 2008-02.one, 2008-03.one, 2008-04.one, 2008-05.one,
2008-06.one, 2008-07.one, 2008-08.one, 2008-09.one, 2008-10.one, 2008-11.one, 2008-12.one,
2009-01.one, 2009-02.one, 2009-03.one, 2009-04.one, 2009-05.one, 2009-06.one, 2009-07.one,
2009-08.one, 2009-09.one, 2009-10.one, 2009-11.one, 2009-12.one, 2010-01.one, 2010-02.one,
2010-03.one, 2010-04.one, 2010-05.one, 2010-06.one, 2010-07.one, 2010-08.one,
2010-09.one, 2010-10.one, 2010-11.one, 2010-12.one, 2011-01.one, 2011-02.one,
2011-03.one, 2011-04.one, 2011-05.one, 2011-06.one, 2011-07.one, 2011-08.one,
2011-09.one, 2011-10.one, 2011-11.one, 2011-12.one, 2012-01.one, 2012-02.one,
2012-03.one, 2012-04.one, 2012-05.one, 2012-06.one, 2012-07.one, 2012-08.one,
2012-09.one, 2012-10.one, 2012-11.one, 2012-12.one, 2013-01.one, 2013-02.one,
2013-03.one, 2013-04.one, 2013-05.one, 2013-06.one, 2013-07.one, 2013-08.one,
2013-09.one, 2013-10.one, 2013-11.one, 2013-12.one, 2014-01.one, 2014-02.one,
2014-03.one, 2014-04.one, 2014-05.one, 2014-06.one, 2014-07.one, 2014-08.one,
2014-09.one, 2014-10.one, 2014-11.one, 2014-12.one, 2015-01.one, 2015-02.one,
2015-03.one, 2015-04.one, 2015-05.one, 2015-06.one, 2015-07.one, 2015-08.one,
2015-09.one, 2015-10.one, 2015-11.one, 2015-12.one, 2016-01.one, 2016-02.one,
2016-03.one, 2016-04.one, 2016-05.one, 2016-06.one, 2016-07.one, 2016-08.one,
2016-09.one, 2016-10.one, 2016-11.one, 2016-12.one, 2017-01.one, 2017-02.one,
2017-03.one, 2017-04.one, 2017-05.one, 2017-06.one, 2017-07.one, 2017-08.one,
2017-09.one, 2017-10.one, 2017-11.one, 2017-12.one, 2018-01.one, 2018-02.one,
2018-04.one, 2018-05.one, 2018-06.one, 2018-07.one, 2018-08.one, 2018-09.one,
2018-10.one, 2018-11.one, 2018-12.one, 2019-01.one, 2019-02.one, 2019-03.one,
2019-04.one, 2019-05.one, 2019-06.one, 2019-06.one, OneNote_DeletedPages.one, 2008.one,
2009.one, 2010.one, 2011.one, Alekseev - Torossian.one, Etingof - Kazhdan.one,
06-240.one, 08-1300.one, 08-401.one, 09-1350.one, 09-240.one, 10-1100.one,
10-327.one, 11-1100.one, 12-240.one, 12-267.one, 12-wClips.one, 13-Aarhus.one,
14-1100.one, 14-1350-AKT.one, 14-240.one, 15-344-Combinatorics.one,
15-475-ProblemSolving.one, 1617-257a-AnalysisII.one, 1617-257b-AnalysisII.one,
16-1750-ShamelessMathematica.one, 16-475-ProblemSolving.one, 17-1350-AKT.one,
17-1750-ShamelessMathematica.one, 17-SummerHomology.one, 18-327-Topology.one,
18S-AKT.one, 20-QGC.one, General.one, OneNote_DeletedPages.one, Vo.one, Zung.one,
AcademicPensieve.one, Citizenship.one, Discovery-2017.one, Duflo.one, ExQu.one,
FairMapping.one, K250.one, KBH.one, KnotAtlas.one, KnotTheory.one, OneCo-1604.one,
PlanetHopf.one, PPSA.one, Profile.one, Projects.one, Quadraticity2.one, Quadraticity.one,
ScatterAndGlow.one, SL2Invariant.one, SL2Portfolio.one, Tilings.one, UChair.one,
w-Computations.one, Web.one, wEnumeration.one, WKO4.one, WKO.one, BonnByWeb-1805.one,
Buffalo-170916.one, CRM-1907.one, DaNang-1905.one, Greece-1607.one, GWU-1612.one,
Hefei-1811.one, Indiana-1611.one, LesDiablerets-1608.one, LesDiablerets-1708.one,
MAASeaway-1810.one, Machines.one, Matemale-1804.one, McGill-1702.one, MIT-1612.one,
NCSU-1604.one, Niagara-1612.one, PolyPoly-1605.one, StonyBrook-1805.one,
Sydney-1708.one, Toronto-1609.one, Toronto-1811.one, Toulouse-1705.one, UNC-1610.one}
```

In[ ]:= **OneNoteLink = Pensieve`Private`OneNoteLink**

Out[ ]=

« NETObject[System.\_\_ComObject] »

```
In[*]:= FileType@"C:\\drorbn\\AcademicPensieve\\ByDate\\2019-06.one"
```

```
Out[*]=
File
```

```
In[*]:= OneNoteFilename = "C:\\drorbn\\AcademicPensieve\\ByDate\\2019-06.one";
OneNoteLink@OpenHierarchy[OneNoteFilename, "", OneNoteDocument]
```

```
NETLink`NET: No method named OpenHierarchy exists for the given COM object.
```

```
Out[*]=
$Failed
```

```
In[*]:= OneNoteFilename = "C:\\drorbn\\AcademicPensieve\\ByDate\\2019-01.one";
OneNoteLink@OpenHierarchy[OneNoteFilename, "", OneNoteDocument]
```

```
NETLink`NET: No method named OpenHierarchy exists for the given COM object.
```

```
Out[*]=
$Failed
```

```
In[*]:= NETLink`NETTypeInfo@OneNoteLink
```

```
NETLink`NETTypeInfo: Type information is not currently available for "raw" COM objects.
```

```
Out[*]//TableForm=
{ }
```

```
In[*]:= NETLink`NETTypeInfo["System.Diagnostics.Process"]
```

### ● Type

```
class System.Diagnostics.Process
```

```
Inheritance:
```

```
System.Object
```

```
System.MarshalByRefObject
```

```
System.ComponentModel.Component
```

```
System.Diagnostics.Process
```

```
Interfaces Implemented: System.ComponentModel.IComponent, IDisposable
```

```
Assembly-Qualified Name: System.Diagnostics.Process, System, Version=4.0.0.0, Culture=neutral
```

```
Assembly Location: C:\windows\Microsoft.Net\assembly\GAC_MSIL\System\v4.0.0.0__b77a5c561e34e829\System.dll
```

### ● Constructors

```
Process ()
```

### ● Properties

```
int BasePriority [read only]
```

```
System.ComponentModel.IContainer Container [read only]
```

```
bool EnableRaisingEvents
```

```
int ExitCode [read only]
```

```
DateTime ExitTime [read only]
```

```
IntPtr Handle [read only]
```

```
int HandleCount [read only]
```

```
bool HasExited [read only]
```

```
int Id [read only]
string MachineName [read only]
System.Diagnostics.ProcessModule MainModule [read only]
IntPtr MainWindowHandle [read only]
string MainWindowTitle [read only]
IntPtr MaxWorkingSet
IntPtr MinWorkingSet
System.Diagnostics.ProcessModuleCollection Modules [read only]
int NonpagedSystemMemorySize [read only]
long NonpagedSystemMemorySize64 [read only]
int PagedMemorySize [read only]
long PagedMemorySize64 [read only]
int PagedSystemMemorySize [read only]
long PagedSystemMemorySize64 [read only]
int PeakPagedMemorySize [read only]
long PeakPagedMemorySize64 [read only]
int PeakVirtualMemorySize [read only]
long PeakVirtualMemorySize64 [read only]
int PeakWorkingSet [read only]
long PeakWorkingSet64 [read only]
bool PriorityBoostEnabled
System.Diagnostics.ProcessPriorityClass PriorityClass
int PrivateMemorySize [read only]
long PrivateMemorySize64 [read only]
TimeSpan PrivilegedProcessorTime [read only]
string ProcessName [read only]
IntPtr ProcessorAffinity
bool Responding [read only]
Microsoft.Win32.SafeHandles.SafeProcessHandle SafeHandle [read only]
int SessionId [read only]
virtual System.ComponentModel.ISite Site
System.IO.StreamReader StandardError [read only]
System.IO.StreamWriter StandardInput [read only]
System.IO.StreamReader StandardOutput [read only]
System.Diagnostics.ProcessStartInfo StartInfo
DateTime StartTime [read only]
System.ComponentModel.ISynchronizeInvoke SynchronizingObject
System.Diagnostics.ProcessThreadCollection Threads [read only]
TimeSpan TotalProcessorTime [read only]
TimeSpan UserProcessorTime [read only]
int VirtualMemorySize [read only]
long VirtualMemorySize64 [read only]
int WorkingSet [read only]
```

long **WorkingSet64** [read only]

### ● **Methods**

```

void BeginErrorReadLine ()
void BeginOutputReadLine ()
void CancelErrorRead ()
void CancelOutputRead ()
void Close ()
bool CloseMainWindow ()
virtual System.Runtime.Remoting.ObjRef CreateObjRef (Type requestedType)
void Dispose ()
static void EnterDebugMode ()
virtual bool Equals (object obj)
static bool Equals (object objA, object objB)
static System.Diagnostics.Process GetCurrentProcess ()
virtual int GetHashCode ()
object GetLifetimeService ()
static System.Diagnostics.Process GetProcessById (int processId, string machineName)
static System.Diagnostics.Process GetProcessById (int processId)
static System.Diagnostics.Process[] GetProcesses ()
static System.Diagnostics.Process[] GetProcesses (string machineName)
static System.Diagnostics.Process[] GetProcessesByName (string processName)
static System.Diagnostics.Process[] GetProcessesByName (string processName, string machineName)
Type GetType ()
virtual object InitializeLifetimeService ()
void Kill ()
static void LeaveDebugMode ()
static bool ReferenceEquals (object objA, object objB)
void Refresh ()
bool Start ()
static System.Diagnostics.Process Start (string fileName, string userName, System.Security.Principal.PrincipalPolicy policy)
static System.Diagnostics.Process Start (string fileName, string arguments, string userName, System.Security.Principal.PrincipalPolicy policy)
static System.Diagnostics.Process Start (string fileName)
static System.Diagnostics.Process Start (string fileName, string arguments)
static System.Diagnostics.Process Start (System.Diagnostics.ProcessStartInfo startInfo)
override string ToString ()
bool WaitForExit (int milliseconds)
void WaitForExit ()
bool WaitForInputIdle (int milliseconds)
bool WaitForInputIdle ()

```

### ● **Events**

event EventHandler **Disposed** [arguments to delegate: (object sender, EventArgs e) ]

event System.Diagnostics.DataReceivedEventHandler **ErrorDataReceived** [arguments to delegate:  
 event EventHandler **Exited** [arguments to delegate: (object sender, EventArgs e) ]  
 event System.Diagnostics.DataReceivedEventHandler **OutputDataReceived** [arguments to delegate:

In[\*]:= **NETLink`NETObjectQ[OneNoteLink]**  
 Out[\*]=  
 True

In[\*]:= **? NETLink`\***  
 Out[\*]=

▼ **NETLink`**

<a href="#">ActivateWindow</a>	<a href="#">GetActiveCOMObject</a>	<a href="#">NET</a>	<a href="#">ReleaseNETObject</a>
<a href="#">AddEventHandler</a>	<a href="#">GetAssemblyObject</a>	<a href="#">NETAssembly</a>	<a href="#">RemoveEventHandler</a>
<a href="#">AddNETObjectListener</a>	<a href="#">GetComplexType</a>	<a href="#">NETBlock</a>	<a href="#">RemoveNETObjectListener</a>
<a href="#">BeginNETBlock</a>	<a href="#">GetNETException</a>	<a href="#">NETLink</a>	<a href="#">ReturnAsNETObject</a>
<a href="#">CallingConvention</a>	<a href="#">GetTypeObject</a>	<a href="#">NETNew</a>	<a href="#">SafeArrayAsArray</a>
<a href="#">CallsUnshare</a>	<a href="#">InstallNET</a>	<a href="#">NETNewDelegate</a>	<a href="#">SameObjectQ</a>
<a href="#">CastCOMObject</a>	<a href="#">InstanceOf</a>	<a href="#">NETObject</a>	<a href="#">SaveAssemblyAs</a>
<a href="#">CastNETObject</a>	<a href="#">KeepNETObject</a>	<a href="#">NETObjectListeners</a>	<a href="#">SendDelegateArguments</a>
<a href="#">CreateCOMObject</a>	<a href="#">LanguageSyntax</a>	<a href="#">NETObjectQ</a>	<a href="#">SetComplexType</a>
<a href="#">DefineDLLFunction</a>	<a href="#">LoadCOMTypeLibrary</a>	<a href="#">NETObjectSummaryBox</a>	<a href="#">ShowNETConsole</a>
<a href="#">DefineNETDelegate</a>	<a href="#">LoadedNETAssemblies</a>	<a href="#">NETObjectToExpression</a>	<a href="#">ShowNETWindow</a>
<a href="#">DoNETModal</a>	<a href="#">LoadedNETObjects</a>	<a href="#">NETType</a>	<a href="#">UninstallNET</a>
<a href="#">DoNETModeless</a>	<a href="#">LoadedNETTypes</a>	<a href="#">NETTypeInfo</a>	<a href="#">WrapInNETBlock</a>
<a href="#">EndNETBlock</a>	<a href="#">LoadNETAssembly</a>	<a href="#">NETUILink</a>	<a href="#">\$NETExceptionHandler</a>
<a href="#">EndNETModal</a>	<a href="#">LoadNETType</a>	<a href="#">ReferencedAssemblies</a>	
<a href="#">FixCRLF</a>	<a href="#">MakeNETObject</a>	<a href="#">ReinstallNET</a>	
<a href="#">FormStartPosition</a>	<a href="#">MarshalStringsAs</a>	<a href="#">ReleaseCOMObject</a>	

In[\*]:= ? ShortcutTarget

Out[\*]=

Symbol	
<b>Pensieve`ShortcutTarget</b>	
Definitions	<pre> ShortcutTarget[Pensieve`Private`Ink_String] := Module[   {Pensieve`Private`l, Pensieve`Private`z},   Pensieve`Private`z = FromCharacterCode[0];   Pensieve`Private`altdir = StringReplace[PensieveDirectory, C: -&gt; \\DROR-X220];   Pensieve`Private`l = StringCases[FromCharacterCode[BinaryReadList[Pensieve`Private`Ink]],     {PensieveDirectory&lt;-&gt;\\, \\DROR-X220\drorbn&lt;-&gt;Pensieve`Private`z&lt;-&gt;AcademicPensieve\} ~-     Shortest[Pensieve`Private`u_] ~- Pensieve`Private`z -&gt; Pensieve`Private`u];   If[Head[Pensieve`Private`l] === List &amp;&amp; Length[Pensieve`Private`l] &gt; 0, First[Pensieve`Private`l],     Message[ShortcutTarget::failed, Pensieve`Private`Ink];   ] </pre>
Full Name	Pensieve`ShortcutTarget

In[\*]:= ShortcutTarget [

**Ink = "C:\\drorbn\\AcademicPensieve\\2019-07\\AcademicPensieve - Shortcut.lnk"]**

Out[\*]=

Projects\AcademicPensieve