Welcome to new students seminar!!

EnScript

Security is a people problem…

forensic-proof.com

proneer.tistory.com

proneer@gmail.com

@pr0neer

JK Kim

Outline

- ***Basic EnScript***
  - *What is the script?*
  - *Introduction*
  - *EnScript Navigation*
  - *Typical EnScript*
  - *User-defined EnScript*
  - *How to get a EnScript Library*

- ***EnScript Programming***
  - *EnScripts*
  - *Filters*
  - *Conditions*
  - *Queries*
  - *User-defined EnScript*
  - *How to get a EnScript Library*
  - *Write a EnScript*
  - *"Hello, Wolrd" EnScript*
  - *Advanced EnScript*

What is the script?

- A programming language that allows control of one or more software applications.
- Often interpreted from source code or bytecode, whereas the application they control are traditionally compiled to native machine code.
- Early script languages were often called batch languages or job control languages.
- Shell Script(UNIX shell), MS Batch (COMMAND.COM)
- JavaScript, VBScript, XSLT, AJAX
- Unix AWK, grep
- Perl, Python, Ruby

Basic EnScript

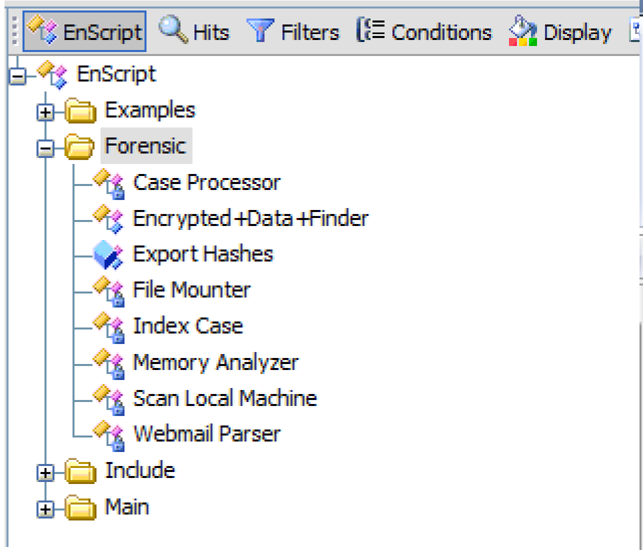
Introduction

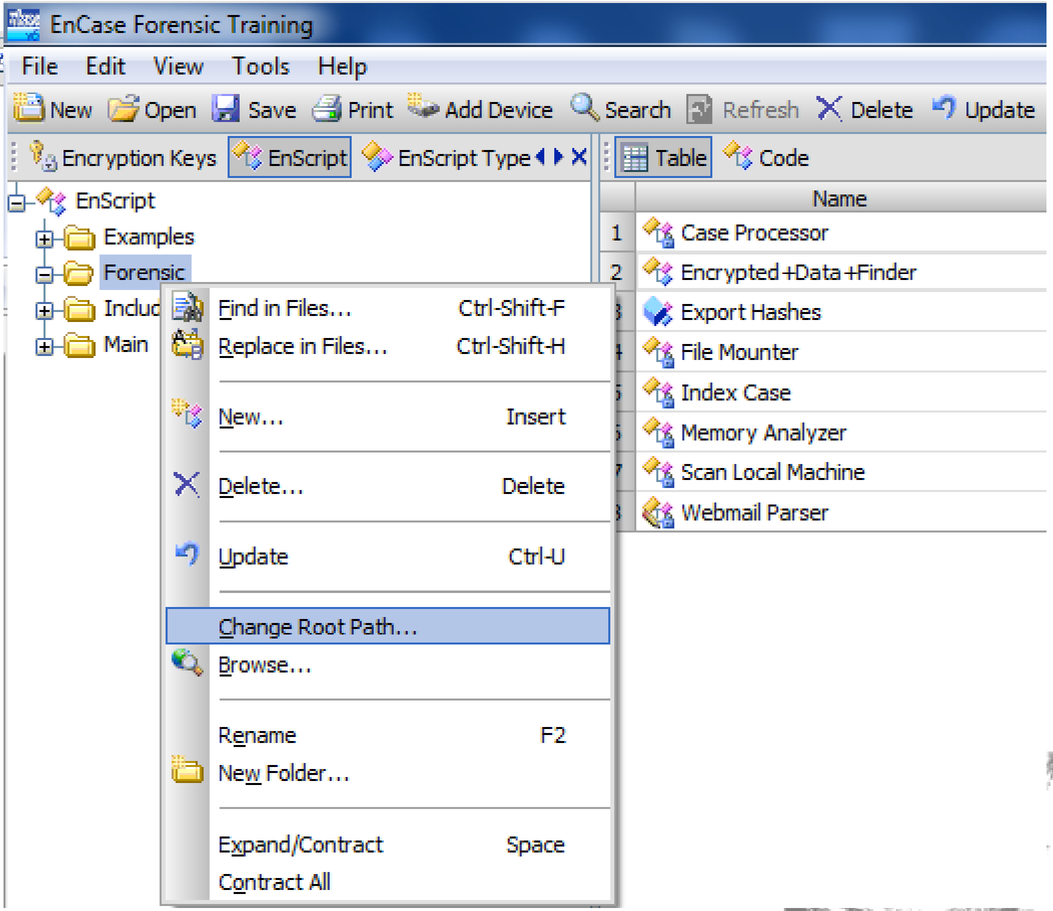
- Has nothing of GNU EnScript
  - converts ASCII files to PostScript, HTML, or RTF.
- Provided by Guidance Software.
- Similar to the ANSI C++ / Java
  - Expression evaluation
  - Operator meanings
  - Making for an easy transition

Introduction

- Even though it resemble C++/Java, just a script. no compile process.
- Like many processes in EnCase, still use the product while a script is running.
- EnScript   vs   EnPack
    - EnPack is introduced a new concept in script technology at EnCase v5.
    - EnPack is a pre-compiled version of an EnScript.
    - It’s main purpose is to render EnScripts “black-box”

EnScript Navigation





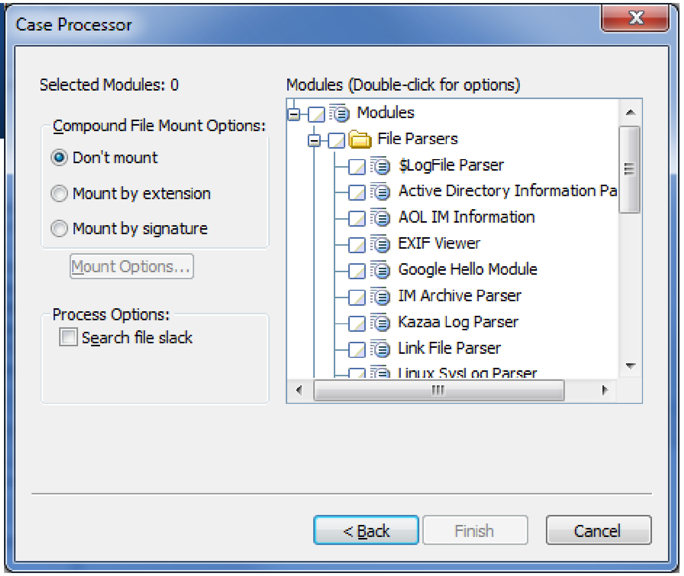
- C:\Program Files\EnCase6\EnScript\

Basic EnScript

Typical EnScript

- **Enterprise EnScript**
  - Document Incident
  - Machine Survey Servlet Deploy
  - Quick Snapshot
  - Snapshot Differential Report
  - Sweep Enterprise
- **Forensic EnScript**
  - Case Processor
  - File Mounter
  - Index Case
  - Scan Local Machine
  - Webmail Parser

Typical EnScript – Case Processor



Case Processor

Selected Modules: 0

Compound File Mount Options:

● Don't mount

○ Mount by extension

○ Mount by signature

Mount Options...

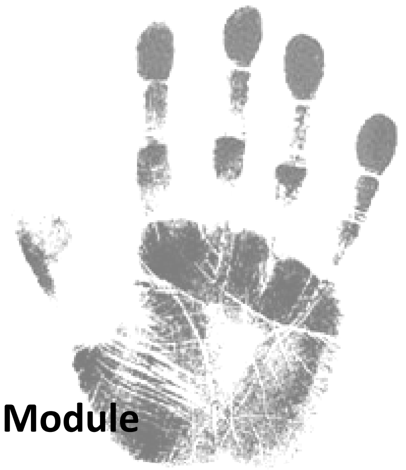
Process Options:

□ Search file slack

Modules (Double-click for options)

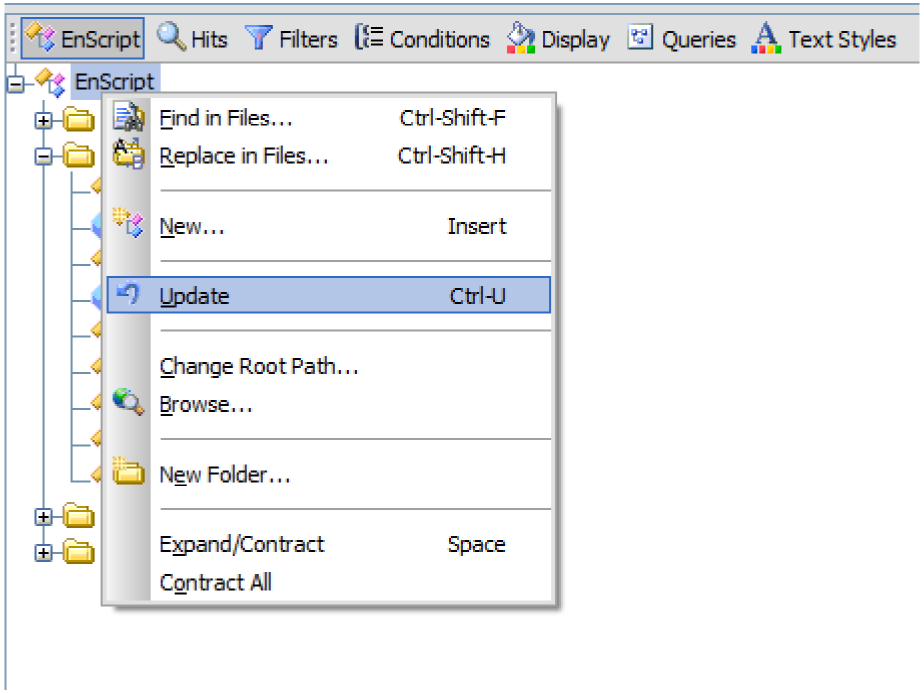
- □ Modules
  - □ File Parsers
    - □ $LogFile Parser
    - □ Active Directory Information Pa
    - □ AOL IM Information
    - □ EXIF Viewer
    - □ Google Hello Module
    - □ IM Archive Parser
    - □ Kazaa Log Parser
    - □ Link File Parser
    - □ Linux SysLog Parser

- **File Parsers**
  - $LogFile Parser, Active Directory Information Parser
  - AOL IM Information, EXIF Viewer, Google Hello Module,
  - IM Archive Parser, Kazaa Log Parser, Link File Parser, Linux SysLog Parser
  - Windows Event Log Parser, WTMP – UTMP Log File Parser
- **Information Finders**
  - Find Protected Files, HTML Carver, Partition Finder, File Finder
  - Recycle Bin Info Record Finder, Credit Card Finder, E-Mail Address Finder
- **Case Initializers**
  - Linux, Mac, Windows Initialize Case
- **App Descriptor Utility, File Report, Scan Registry, Time Windows Analysis Module**



User-defined EnScript

- Add the user-defined EnScript to that Root (C:\Program Files\EnCase6\EnScript\)



How to get a EnScript Library

- Downloads page in the support section at http://www.guidancesoftware.com/
- Message board - http://www.guidancesoftware.com/support/messageboards.asp
- Blog – ForensicKB (http://www.forensickb.com/)

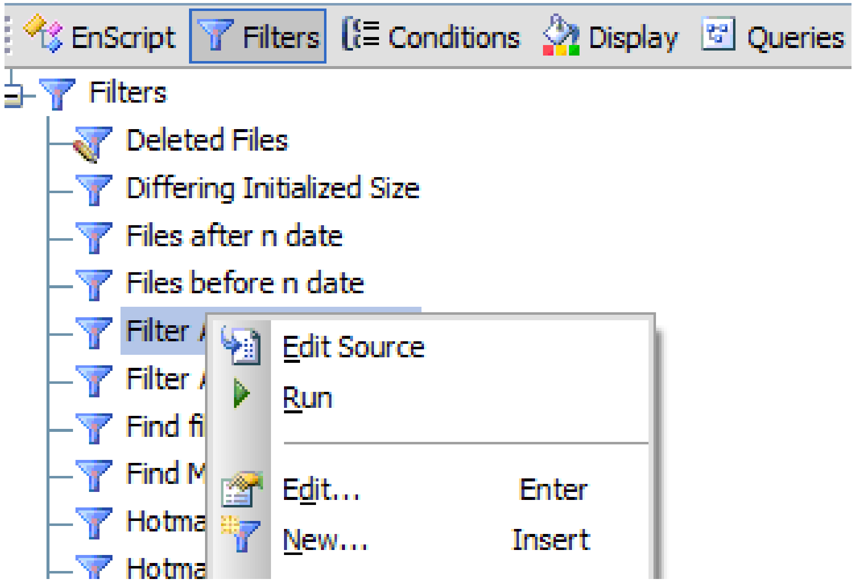
EnScripts

- Expressions(operators, functions, variables), Array and so on are coterminous C++/Java.
- Loops(for, while, do, foreach, forall), datatype and so on also are much the same.

Types	Size	Range
Char	2 bytes	0 – 65,535
String	Null terminated	
Short	2 bytes number	-32,768 – 32,767
Ushort	2 bytes number	0 – 65,535
Int	4 bytes number	-2.1 – 2.1 billion
Uint	4 bytes number	0 – 4.2 billion
Long	8 bytes number	-9.2 – 9.2 sextillion
Ulong	8 bytes number	0 – 18.4 sextillion
Double	8 bytes number	Decimal value
Void	void	
Bool	True/False	true or false

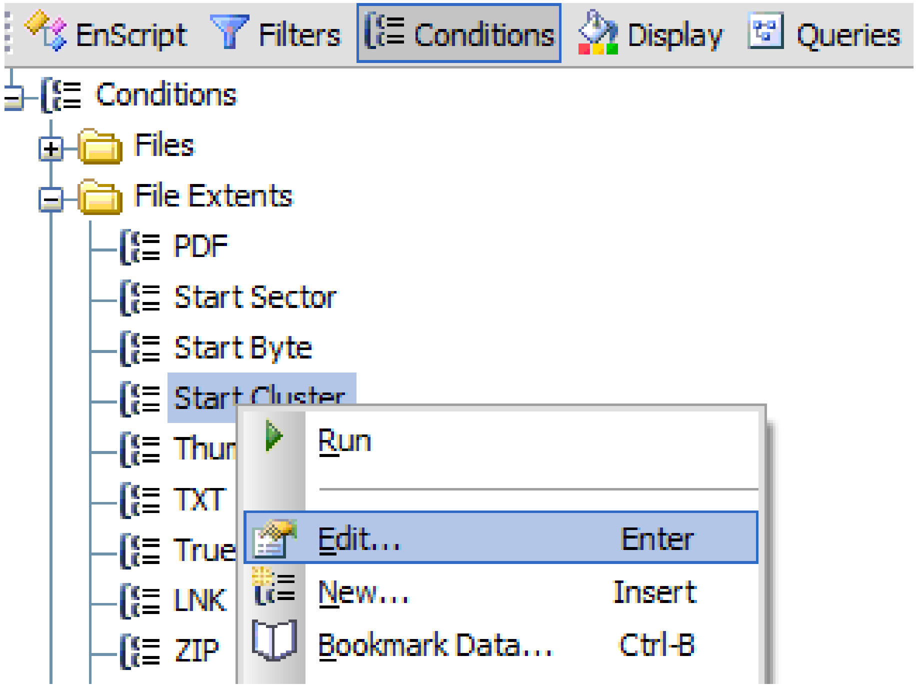
Filters

- A Filter is a special EnScript.
- The concept was to filter files/folders base on some type of criteria :
  - File extension, size, name, whatever…



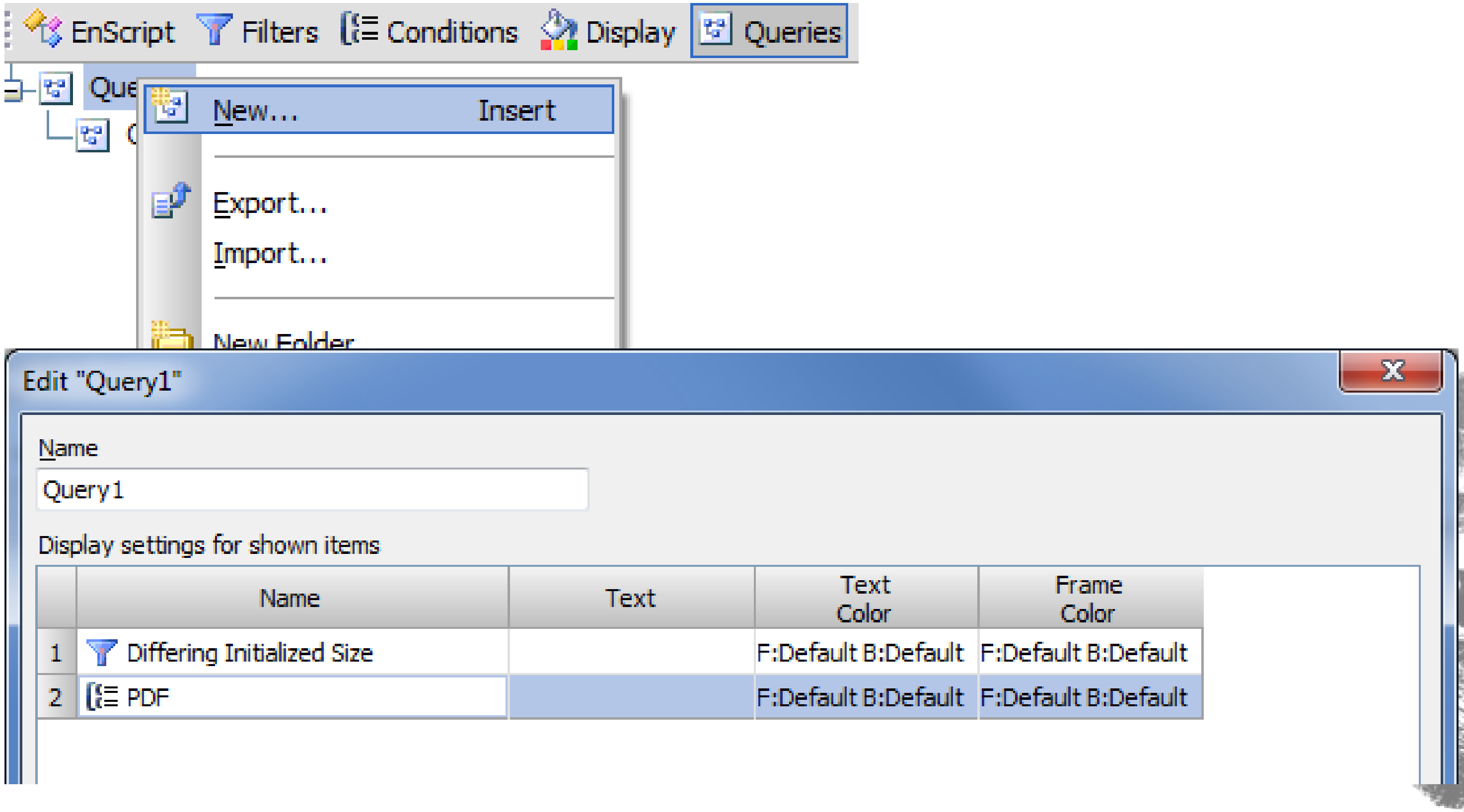
Conditions

- A Condition is exactly the same as a filter, except you don't need to know how to write EnScript programming language.



Queries

- A Query is nothing more than two or more filters and conditions put together.



Edit "Query1"

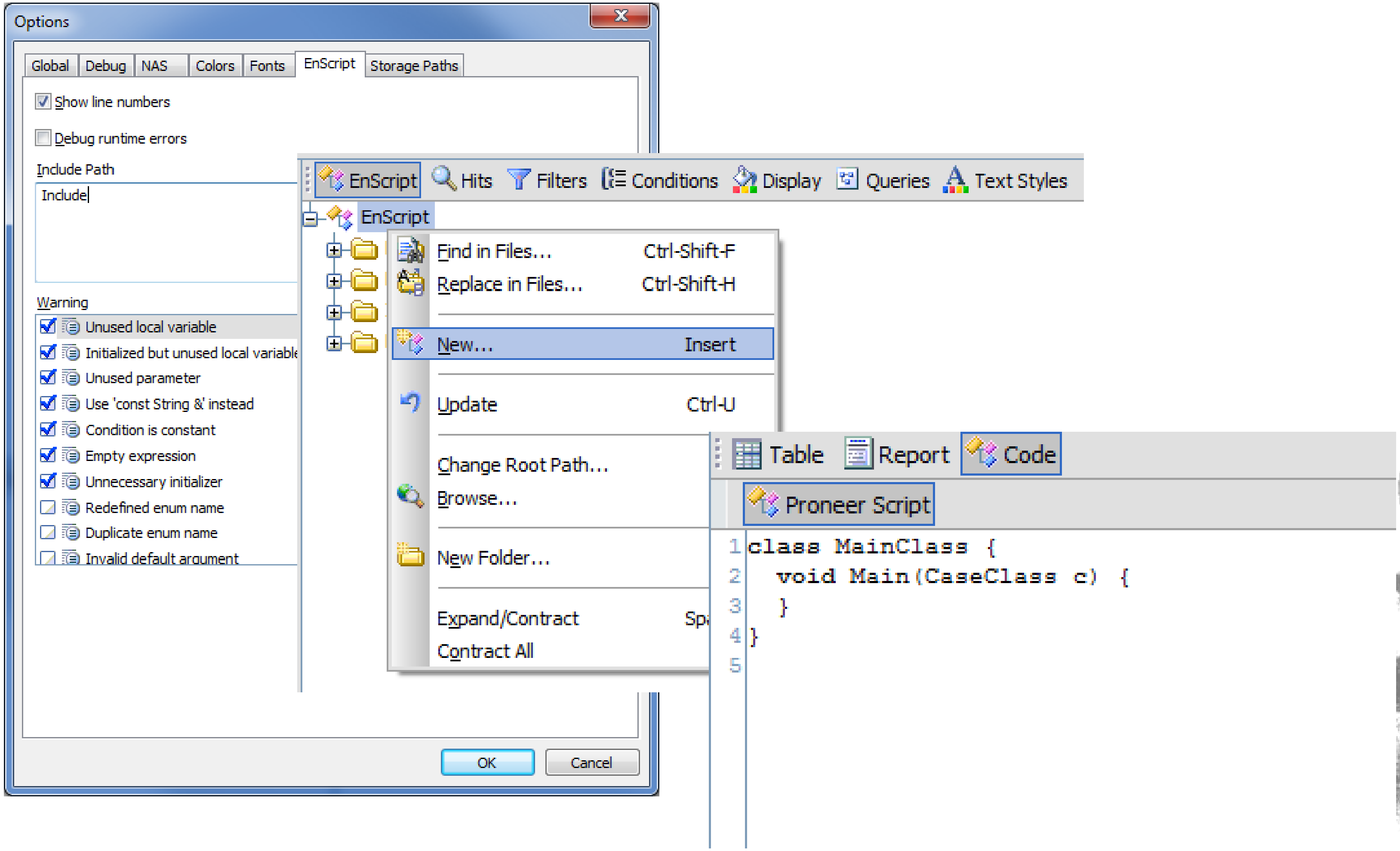
Name
Query1

Display settings for shown items

	Name	Text	Text Color	Frame Color
1	Differing Initialized Size		F:Default B:Default	F:Default B:Default
2	PDF		F:Default B:Default	F:Default B:Default

EnScript Programming

Write a EnScript



☑ Show line numbers

☐ Debug runtime errors

Include Path

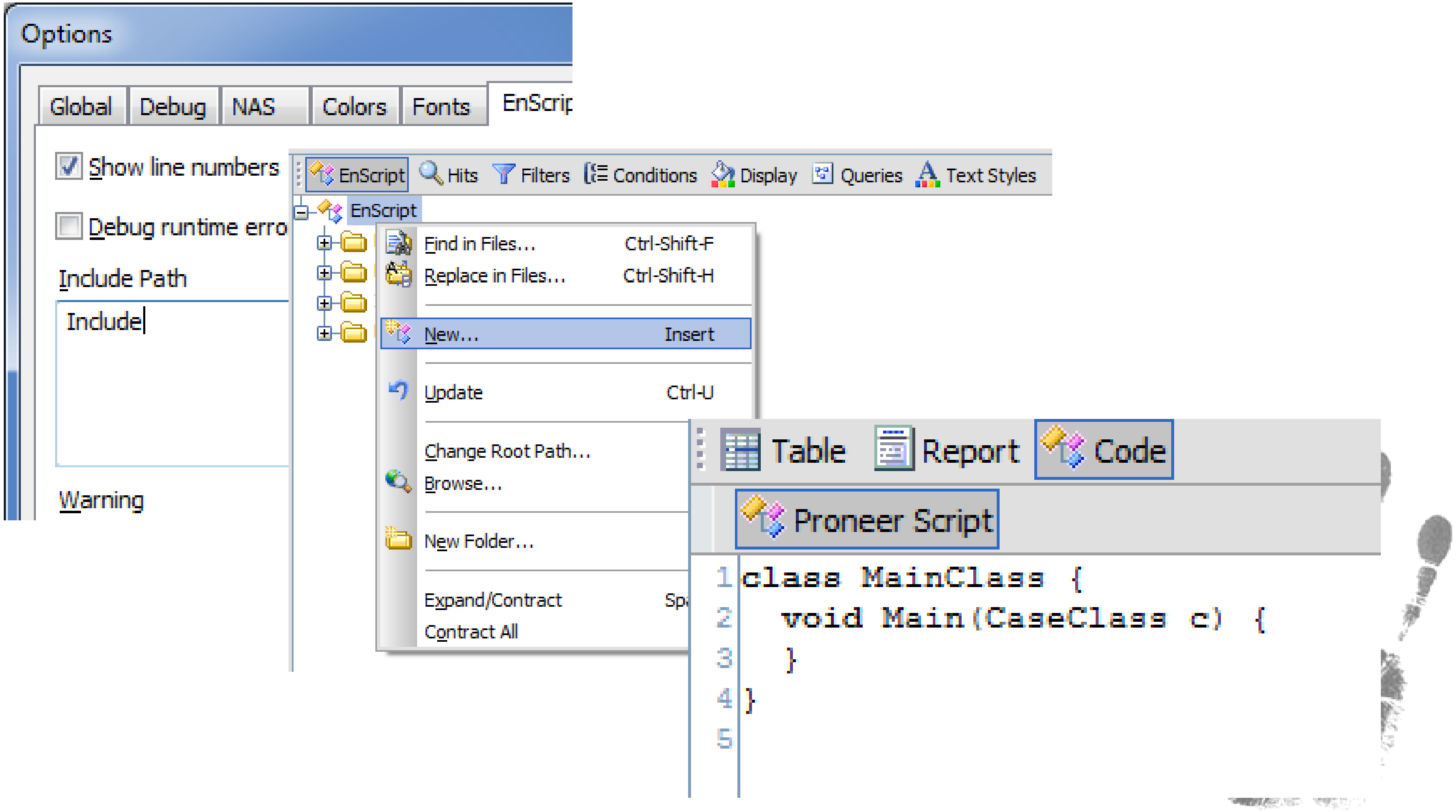
Include

Warning

☑ Unused local variable
☑ Initialized but unused local variable
☑ Unused parameter
☑ Use 'const String &' instead
☑ Condition is constant
☑ Empty expression
☑ Unnecessary initializer
☐ Redefined enum name
☐ Duplicate enum name
☐ Invalid default argument

```
class MainClass {
  void Main(CaseClass c) {
  }
}
```

Write a EnScript



☑ Show line numbers

□ Debug runtime erro

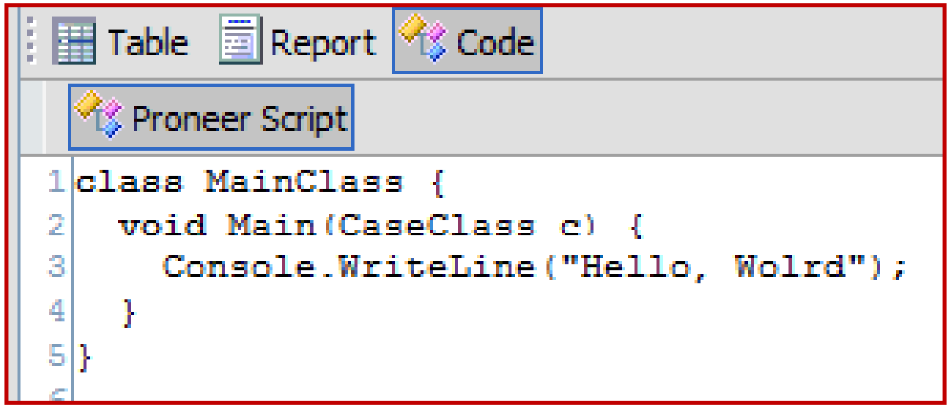
Include Path

Include

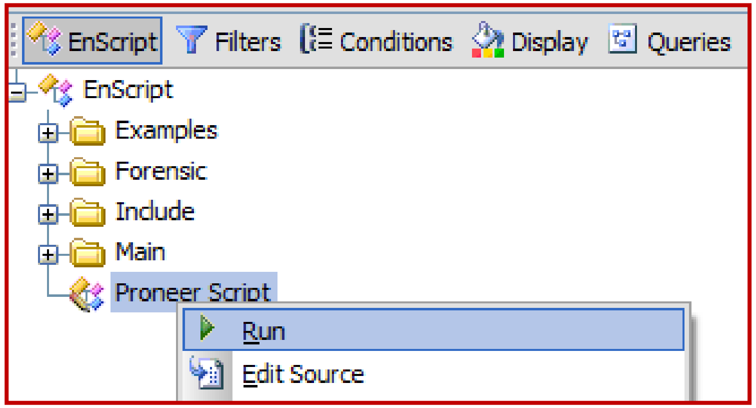
Warning

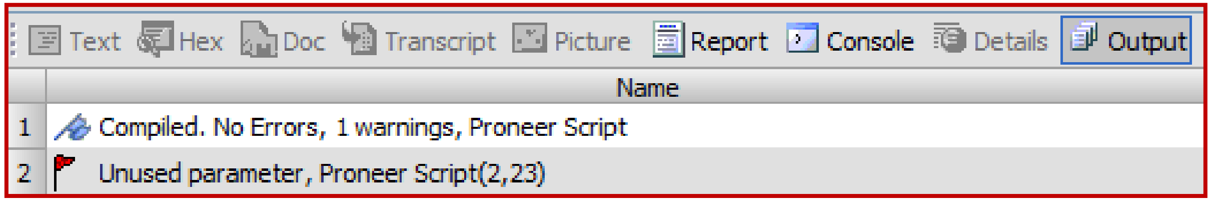
```
class MainClass {
  void Main(CaseClass c) {
  }
}
```

“Hello, World” Enscript

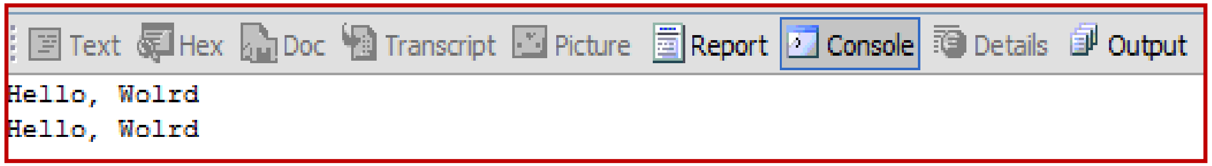


```
class MainClass {
  void Main(CaseClass c) {
    Console.WriteLine("Hello, Wolrd");
  }
}
```



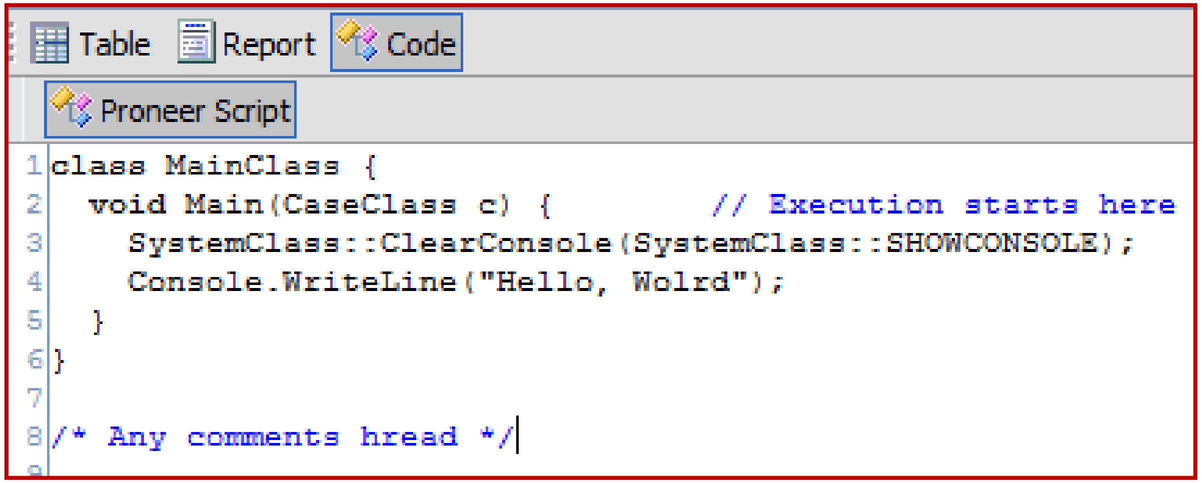


	Name
1	Compiled. No Errors, 1 warnings, Proneer Script
2	Unused parameter, Proneer Script(2,23)



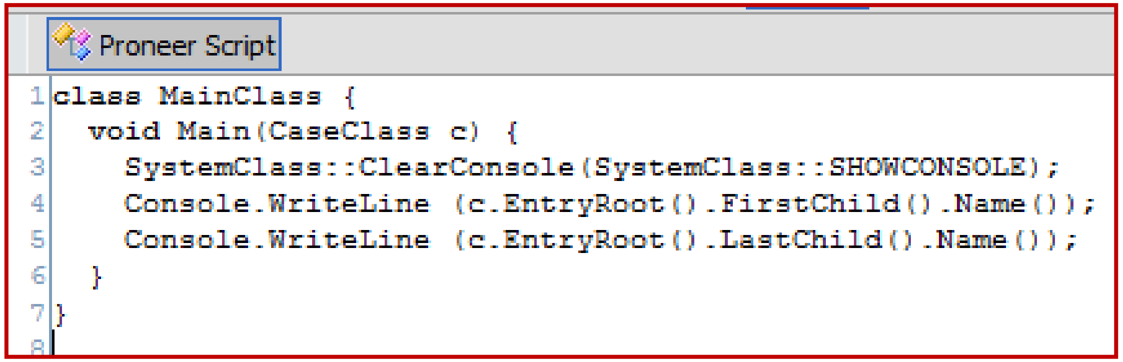
```
Hello, Wolrd
Hello, Wolrd
```

“Hello, World” Enscript

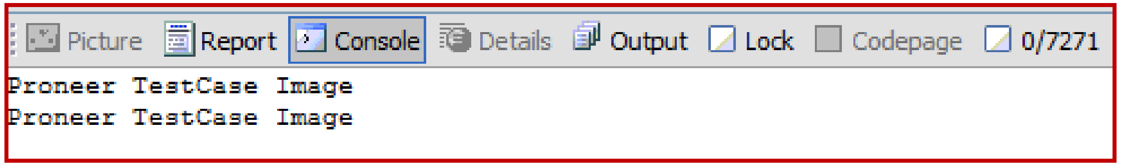


```
class MainClass {
  void Main(CaseClass c) {          // Execution starts here
    SystemClass::ClearConsole(SystemClass::SHOWCONSOLE);
    Console.WriteLine("Hello, Wolrd");
  }
}

/* Any comments hread */
```

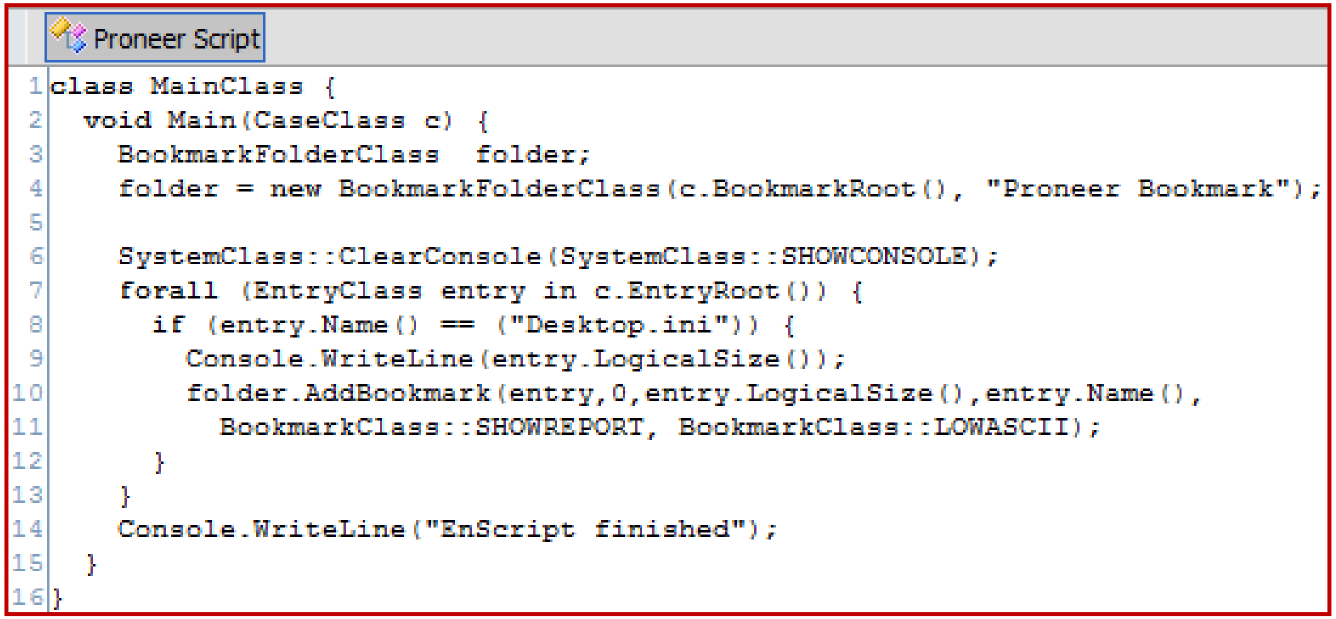


```
class MainClass {
  void Main(CaseClass c) {
    SystemClass::ClearConsole(SystemClass::SHOWCONSOLE);
    Console.WriteLine (c.EntryRoot().FirstChild().Name());
    Console.WriteLine (c.EntryRoot().LastChild().Name());
  }
}
```



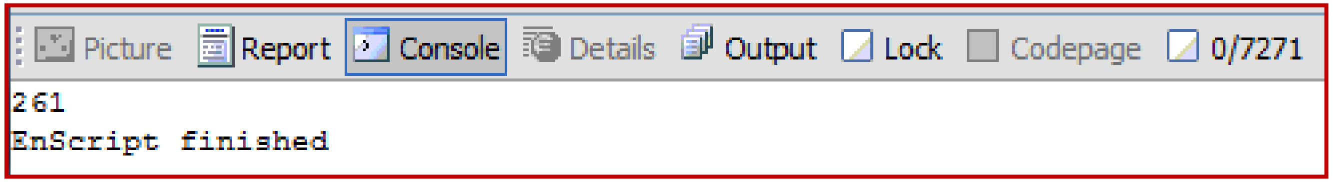
```
Proneer TestCase Image
Proneer TestCase Image
```

Advanced EnScript

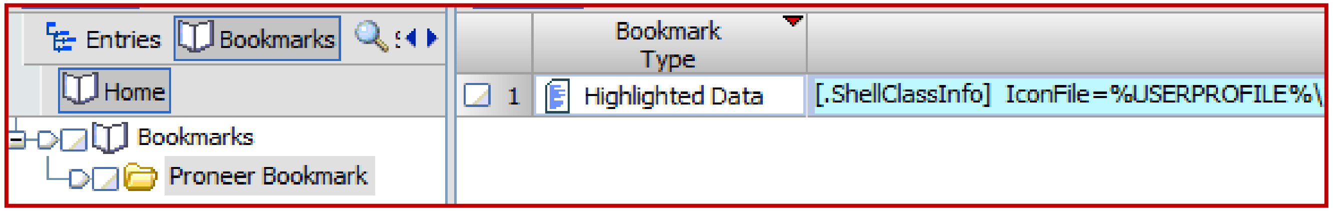


```
class MainClass {
  void Main(CaseClass c) {
    BookmarkFolderClass  folder;
    folder = new BookmarkFolderClass(c.BookmarkRoot(), "Proneer Bookmark");

    SystemClass::ClearConsole(SystemClass::SHOWCONSOLE);
    forall (EntryClass entry in c.EntryRoot()) {
      if (entry.Name() == ("Desktop.ini")) {
        Console.WriteLine(entry.LogicalSize());
        folder.AddBookmark(entry,0,entry.LogicalSize(),entry.Name(),
          BookmarkClass::SHOWREPORT, BookmarkClass::LOWASCII);
      }
    }
    Console.WriteLine("EnScript finished");
  }
}
```



```
261
EnScript finished
```

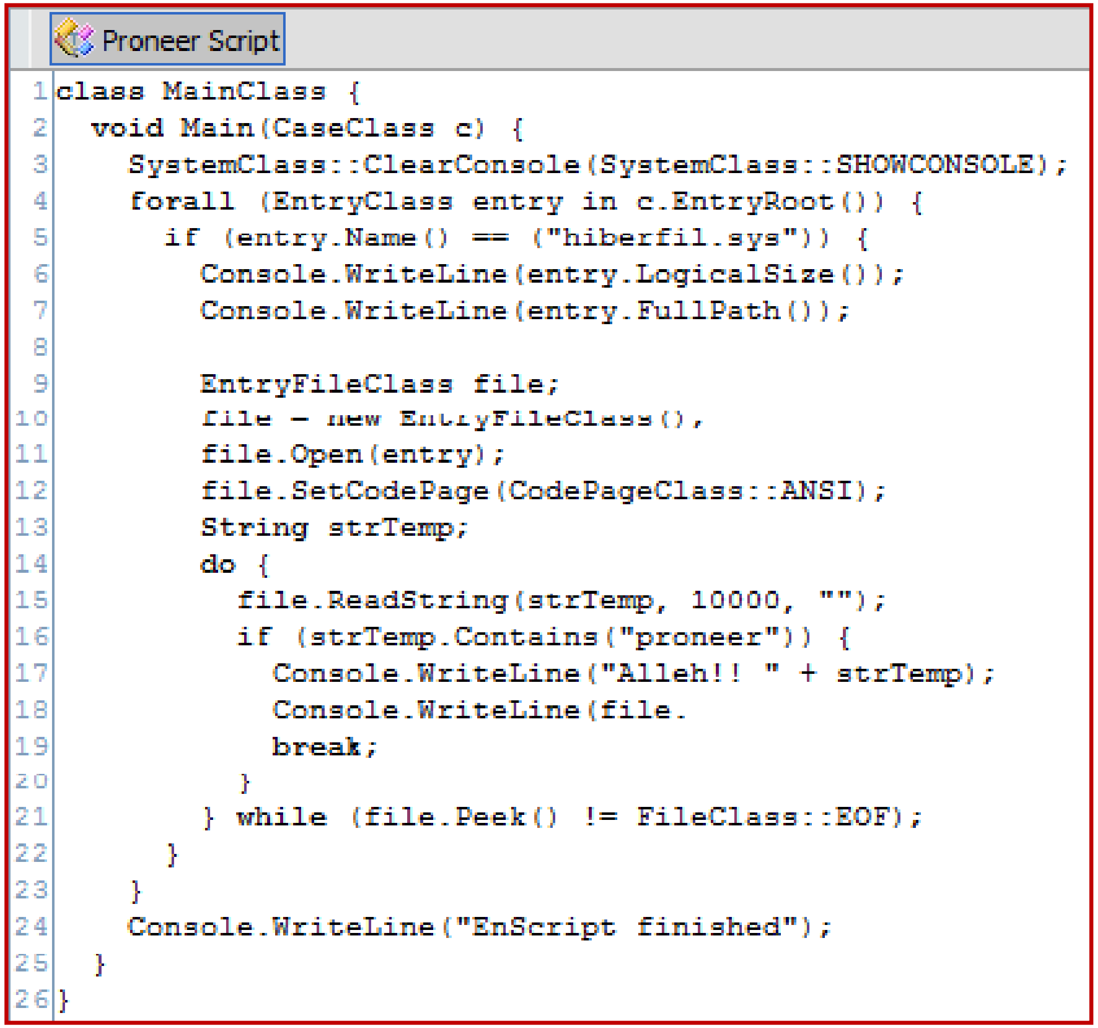


- Bookmarks
  - Proneer Bookmark

		Bookmark Type	
□	1	Highlighted Data	[.ShellClassInfo] IconFile=%USERPROFILE%\

EnScript Programming

Advanced EnScript



```
class MainClass {
  void Main(CaseClass c) {
    SystemClass::ClearConsole(SystemClass::SHOWCONSOLE);
    forall (EntryClass entry in c.EntryRoot()) {
      if (entry.Name() == ("hiberfil.sys")) {
        Console.WriteLine(entry.LogicalSize());
        Console.WriteLine(entry.FullPath());

        EntryFileClass file;
        file = new EntryFileClass();
        file.Open(entry);
        file.SetCodePage(CodePageClass::ANSI);
        String strTemp;
        do {
          file.ReadString(strTemp, 10000, "");
          if (strTemp.Contains("proneer")) {
            Console.WriteLine("Alleh!! " + strTemp);
            Console.WriteLine(file.
            break;
          }
        } while (file.Peek() != FileClass::EOF);
      }
    }
    Console.WriteLine("EnScript finished");
  }
}
```

Advanced EnScript

```
class MainClass {
  String Path;
  void Variables (uint props) {
    StorageClass storage("wordread", props);
    storage.Value("path", Path);
  }
  void Main() {
    Variables(0);
    DialogClass dialog(MainWindow, “What is the name of this program????");
    new PathEditClass(dialog, "Path", WindowClass::START, WindowClass::START, 250,
                              WindowClass::DEFAULT, 0, Path, WindowClass::REQUIRED | WindowClass::FILEOPEN);
    if (dialog.Execute() == SystemClass::OK) {
      Word::Application app;
      if (app.Create()) {
        SystemClass::ClearConsole();
        Word::Document doc = app.Documents().Open(Path);
        if (doc)  Console.Write(doc.Range().Text());
        else {
          LogClass log("Read Word document", LogClass::WARN);
          log.Warn("Unable to open " + Path);
        }
        app.Quit();
      }
    }
    Variables(StorageClass::WRITE);
  }
}
```

Conclusion

- Where EnScript API Reference can be found?

Question & Answer