

Introduction to Using Perl with ProDiscover[®]

~HTCIA Intl. 2005 ~

Christopher L. T. Brown, CISSP
Technology Pathways, Founder & CTO
clbrown@techpathways.com
619-435-0906 / 888-894-5500 x111

Copyright © 2005, Technology Pathways, LLC

Purpose of this Presentation

- Review the basic architecture of ProDiscover family of computer forensics products
- Provide attendees a basic understanding of the Perl language
- Introduce attendees to writing, debugging, and running ProScripts

ProDiscover[®] Architecture & Product Family

Tools For The Process

- The ProDiscover[®] family of products are designed to support the forensics process for specific markets
 - ProDiscover[®] – for Windows
 - ProDiscover[®] – Forensics (includes Perl)
 - ProDiscover[®] – Investigator (includes Perl)
 - ProDiscover[®] – Incident Response (IR) (includes Perl)
 - More to come...
- Basic features and UI are common to all family tools
- Customized functionality in each tool to suit the user

ProDiscover[®] Core Architecture

- Read the disk at the sector level in a Read-Only mode
- Perform all display and functions through it's own trusted, read-only file system
- Currently supports all versions of FAT and NTFS including Dynamic Disk, Software RAID and Volume Sets, SUN Solaris UFS on SPARC and X86 Platforms, LINUX Ext2/3

Independent Validation (1)

- ProDiscover® has been utilized by professionals to find and present evidence for civil and criminal proceedings
- An Independent Review of Forensic Tools was recently published by Mark Scott on the SANS web site validating the ProDiscover imaging process
<http://www.sans.org/rr/special/forensicimaging.php>

Independent Validation (2)

- The Dartmouth Institute for Security Technology Studies report Law Enforcement Tools and Technologies for Investigating Cyber Attacks - Gap Analysis Report ranks ProDiscover® as meeting the greatest number of needs for Preliminary Investigation and Data Collection.
- http://www.ists.dartmouth.edu/TAG/gap_analysis.htm

Independent Validation (3)

- Digital Investigations Journal Vol. 1 No. 4 (December, 2004)
<http://www.compseconline.com/digitalinvestigation>
 - *Remote Forensics'*, by Philip Sealey
 - *Tool review - remote forensic preservation and examination tools*, by Eoghan Casey & Aaron Stanley
- Network Computing Magazine (December 2004)
<http://www.networkcomputing.com/>
 - *Network Forensic Tools - Elementary, My Dear Watson*, by Marisa Mack

Scripting

Introduced Scripting in v. 4.0

- Why Script
 - Provides the power to automate repetitive tasks
 - Standardize Investigations
 - Low level analysis (data carving)
 - Integrate with other tools
 - New data analysis (latest version of ???)

ProScript & Perl

- How is scripting implemented:
 - By exposing low level functions of ProDiscover in the form of the ProScript API (around 150 functions)
 - Functions are made available to ActiveState Win32 Perl in the form of a Perl Module
 - ActivePerl 5.8.6 is included in the standard ProDiscover Distribution (All editions except “for windows edition)
 - All Scripts are executed through the ProDiscover Interface

Function Categories

- General Functions (37)
- Project File Functions (6)
- Data Carving Functions (11)
- Disk and Folder Functions (10)
- File Functions (8)
- Search Result Functions (6)
- Registry Functions (8)
- ACL Functions (3)
- Event Log Functions (7)

Advanced API

- Remote System State Functions (16)
- Search Functions (18)
- Auditing Functions (18)

Installation Notes

- ProDiscover will install Perl upon completion of base installation
- Use default directory for Perl installation
- Choose ProDiscover 'Start' menu option to install Perl module after Perl and ProDiscover base installation is complete

Balance of Power

- The power provided by Perl and ProScripts can be dangerous
 - Perl has extensive “non-write protected” I/O capabilities natively and through the over 500 included modules
 - ProScript also provides the power to crash ProDiscover
- Dangers can be mitigated by testing, experience, and standard forensics methodologies

The 5 Commandments of ProScript

- **WARNING:** While ProScript will not write to evidence disk or images in any way
- **BE ALERT:** the use of Perl File I/O to work with an image file can damage the image.
- **ALWAYS:** Users should ensure all image read functions are performed from ProScript API's.
- **ALWAYS:** Users should understand and test all scripts prior to use on actual evidence images.
- **REMEMBER:** Hardware write blockers can help to protect image files from external applications."

What Makes a Perl script a ProScript?

- Simply adding the directive *use ProScript;*
- And using the ProScript API as needed
- Full Access to the Perl API and other modules is provided

Two ProScript Demos & API Reference Walkthrough

Perl Primer

A Bit of Perl History

- Perl - Practical Extraction and Report Language
- Created by linguist Larry Wall in 1987
- Overall Structure taken from “C”, but a very forgiving implementation
- Integrates regular expressions and associated arrays found in scripting languages such as ‘shell’, ‘sed’, and ‘awk’

A Bit of Perl History (2)

- Today Perl includes the ability to manipulate complex data structures, pointers and a cult like following

But why choose Perl?

- Perl was created for text manipulation
- Perl's simplicity in manipulating *Scalar Data* is unmatched
- Extensive body of code, modules, and training already available
- Many people are already using Perl to script other forensics tools

Basic Perl Data Types

\$HoldScalarDataHere

a scalar (number, string, or reference)

@HoldListDataHere

a list (ordered collection of scalars or array)

%HoldAssocArrayHere

a hash, map or lookup table

Associative Arrays

- While a regular array maps integers to arbitrarily typed objects (integers, strings, pointers, and, in an OO sense, objects), an associative array maps arbitrarily typed objects to arbitrarily typed objects.
- For example, if the value associated with the key “evidence” is 7, we say that our array maps “evidence” to 7.

Number and String Assignment

```
$n = 42;
```

```
$name = "chris";
```

```
$color = 'red';
```

Lists

- Assignment

`@Scores = (32, 45, 16, 5);`

- Accessing individual elements

`$Scores[2]` # an element of `@Scores` (zero based)

- Number of elements in an array

`$ScoreCount = @scores`

Control Structures

- Perl provides standard control structures:
 - while, for, foreach
 - if, unless, until
- Subroutines
 - function();
 - sub function { do stuff here... }

Some Useful Perl Resources

- search.cpan.org/
- aspn.activestate.com/ASPN
- www.activestate.com/support
- www.activestate.com/support/enterprise
- perl.oreilly.com
- www.perl.org
- use.perl.org
- The Gecko book from O'Reilly

Putting Perl to use with ProScript

ProScript Directories

- **\ProScript\Documentation** - contains ProScript API documentation
- **\ProScript\Examples** - contains ProScript API and Perl example scripts
- **\ProScript\Output** – an output directory for script output
- **\ProScript\PerlModules** - contains ProScript Perl module setup files and installation batch file.
- **\ProScript\User Scripts** - contains working ProScripts

Writing and Debugging ProScripts

- The most difficult issue when programming in any language is often identifying syntax errors
 - typing "PSCloseHandel(\$nHandle);" when you meant to type "PSCloseHandle(\$nHandle);"
 - Forgetting a trailing ;
 - Case specific issues such as typing 'Use' or 'use'

Troubleshooting Syntax Issues

- Two techniques to help identify and overcome syntax errors:
- Smart GUI based editors
 - *PrimalScript* from www.primalscript.com/
 - *Komodo* from www.activestate.com
- Using Perl from command line to identify syntax issues

Running ProScripts

- ProScripts must be run from ProDiscover
- Four ways to run ProScripts
 - Button Bar Item
 - Right-Click
 - Tools Menu
 - Command Line
 - ProDiscover\IR.exe project.dft script.pl VarToPass

Testing the ProScript Module

- From ProDiscover main window click on “Run ProScript” on the button-bar
- Choose “browse” and naavagate to “<installation directory>\ProScript\Examples”
- Then choose the script “colors.pl”
- Can be tested without an open project

Writing a Simple ProScript

- A simple hello world example

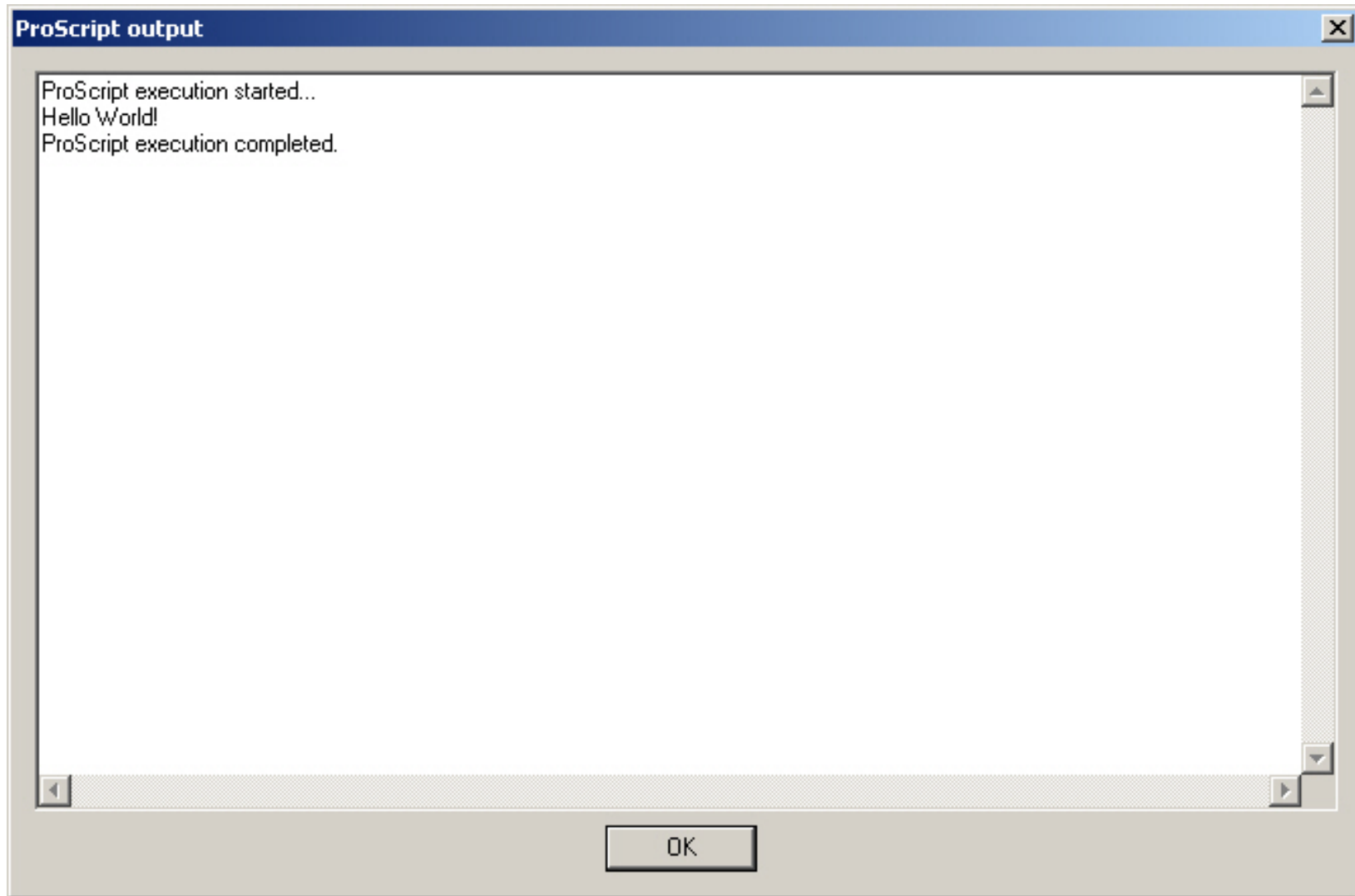
Use ProScript;

```
PSDisplayText("Hello World!");
```

Debugging “Hello World”

- Is the output as expected?
- Debug steps

Corrected Hello World



More Script Walkthroughs

- [ReactHash.pl](#)
- [CarvJPGArtifacts.pl](#)

COMPUTER EVIDENCE

Collection & Preservation

- Teaches investigators how to ensure case integrity when dealing with computer evidence
- Provides a practical resource for collection and preservation that will help ensure legal acceptability
- Covers key areas such as rules of evidence, evidence dynamics, network topologies, collecting volatile data, imaging methodologies, and forensics labs and workstations
- Includes a CD-ROM with shareware and commercial demo software tools as well as document templates, worksheets, and references



Networking & Security Series

CHRISTOPHER L.T. BROWN

Thank You Questions?


Technology
Pathways

**703 First Street
Coronado, Ca. 92118**

**Phone: 888-894-5500
FAX: 619-435-0465**

www.TechPathways.com

clbrown@TechPathways.com



**Are your
tools
keeping
pace with
the
criminals?**