7th USENIX Tcl/Tk Conference
February 14-18, 2000
Austin, Texas, USA

Tutorial Sessions
Monday, Feb. 14 to Tuesday, Feb. 15, 2000

Please select one morning and one afternoon tutorial. Sorry, no partial registrations are allowed.

Monday, February 14, 2000

Morning Sessions (9:00 am - 12:30 pm)

M1am: Effective Tcl/Tk Programming
Michael McLennan, Cadence Design Systems, Inc.

Who should attend: Programmers and managers who are using Tcl/Tk to build real-world applications. This tutorial assumes a working knowledge of Tcl/Tk and goes beyond that to show how nontrivial Tcl/Tk programs are put together.

Topics include:

- How to use the Tk canvas widget to synthesize new types of widgets
- How to use binding tags to add new widget behaviors
- How to organize Tcl code into reusable libraries and packages.

These techniques come together to create professional-looking applications that are easier to maintain and extend.

Michael McLennan has been a Tcl/Tk enthusiast since 1992. He has taught more than 100 Tcl/Tk courses, and is co-author of Effective Tcl/Tk Programming and Tcl/Tk Tools. He also developed [incr Tcl], an object-oriented extension of Tcl/Tk. Michael, who received a Ph.D. in 1990 from Purdue University, is currently an Architect at Cadence Design Systems, Inc.

M2am: XML and Tcl/Tk
Steve Ball, Zveno Pty. Ltd.

Who should attend: Developers who wish to add XML support to their applications, use XML for storing data, or process XML documents using Tcl.

XML is set to revolutionize not only the Web but also many industries and businesses, as it allows easy, robust data exchange between machines. A number of packages now exist to assist the Tcl programmer in dealing with XML documents. These tools include XML parsers, DOM

implementations, and support for XML-based protocols. This tutorial provides an introduction to all of these tools and techniques for their use.

Topics include:

- An overview of XML
- How to write simple Tcl scripts to perform event-based processing of XML documents
- How to manipulate an XML document using TclDOM and generate XML
- How to perform procedure calls using XML-RPC.
- Event-based parsing using TclExpat and TclXML

Now that Tcl includes support for Unicode, it is fully able to process any XML document, including those written in languages other than English. Application developers can combine the benefits of Tcl-easy system integration and rapid development-with the benefits of XML-robust data exchange and structured data representation-to create the next generation of Web and business applications.

Steve Ball is Chief XML and Scripting Specialist at Australia-based Zveno Pty. Ltd. and is the chief architect of Zveno's Swish XML editor. He is the author of Web Tcl Complete (McGraw-Hill) and several Open Source software projects, including TclExpat, TclXML, TclDOM, and the Plume Web browser. Steve was first introduced to Tcl in 1993 and has hardly touched any other language since then.

M3am: Network Management with Scotty
Cameron Laird, Phaseit, Inc.

Who should attend: Network and system administrators, managers, and programmers. This class is primarily aimed at those with networking responsibilities in search of programming technologies that will help them be more effective. Familiarity with Tcl or network management concepts is required; expertise in either is not. Participants will emerge with a wealth of specific useful scripts and resources.

Topics include:

- How to manage your IP networks more easily
- How to get the benefits advertised for commercial network management products with less expense and fragility
- How to develop network management applications efficiently

Learn about the Scotty programming language and the tkined network manager built with it. This remarkable bundle of portable, open-source software both replaces and complements proprietary solutions costing tens of thousands of dollars.

The goal of the class is to make participants thoroughly comfortable with Scotty and tkined's capabilities. The method of the class is to work through several concrete operations in which Scotty

boosts a network administrator's productivity. Students receive abundant documentation that helps them apply the technology to their own situations.

Cameron Laird has over two decades full-time experience in project development. He has authored several dozen magazine articles on information technology, co-authors the "Regular Expressions" column on scripting languages twice a month for SunWorld Online, and is a Contributing Editor for Linux Magazine. He organized the Scotty BOF at the 1997 Tcl Conference in Boston. His own networking code is in production at several national ISPs and government agencies.

Afternoon Sessions (1:30 pm - 5:00 pm)

M4pm: Tcl Extension Building and SWIG
David M. Beazley, University of Chicago

Who should attend: Developers who are involved in building Tcl/Tk interfaces to C/C++ applications or building compiled extensions to Tcl 8.0 and need useful information as well as tools for automating the extension-building process. Participants should be familiar with C, but no prior experience with Tcl extension building is required.

We will explore several approaches to adding new commands and creating new shells in Tcl using C, including returning results and conversion functions. This tutorial will demonstrate the extensibility of Tcl and its strength as an embedded language.

Topics include:

- Accessing Tcl variables
- Mapping Tcl variables to C variables
- Tracing Tcl variables
- Executing Tcl commands from C
- Embedding into an application
- Creating new Tcl/Tk shells vs. dynamic loading of extensions
- Action vs. object commands
- An overview of TEA

The use of SWIG, a tool for automatically generating Tcl interfaces to C/C++ programs, will also be described. While the focus will be on SWIG, many of the topics also apply to other automated Tcl extension building tools.

SWIG-related topics include:

- Building Tcl interfaces to C libraries
- Working with objects and C++
- Exception handling
- Compilation and linking issues
- Interface-building strategies, pitfalls, and tricks

Dave Beazley is the developer of SWIG, a freely available tool for building Tcl, Perl, and Python interfaces to C/C++ applications. He has worked at Los Alamos National Laboratory and has published a variety of articles on the use of scripting environments with high-performance scientific and engineering applications. He is an assistant professor at the University of Chicago in the Department of Computer Science.

M5pm: Object-Oriented Programming with [incr Tcl]
Michael McLennan, Cadence Design Systems, Inc.

Who should attend: Programmers and managers who are using Tcl/Tk to build large, real-world applications. This tutorial assumes a working knowledge of Tcl/Tk. Some background in object-oriented programming is helpful, though not required. [incr Tcl] provides a set of object-oriented extensions for the Tcl language. Since its inception in 1993, it has become the de facto standard for object-oriented Tcl programming. It is currently used by thousands of developers and distributed in dozens of commercial applications worldwide.

Topics include:

- Class definition syntax
- Inheritance (is-a) relationships
- Compositional (has-a) relationships
- Multiple inheritance
- Public/protected/private member protection

Students will gain an understanding of how [incr Tcl] works and how it can be used to create larger, more robust Tcl/Tk applications.

Michael McLennan has been a Tcl/Tk enthusiast since 1992. He has taught more than 100 Tcl/Tk courses, and is co-author of Effective Tcl/Tk Programming and Tcl/Tk Tools. He also developed [incr Tcl], an object-oriented extension of Tcl/Tk. Michael, who received a Ph.D. in 1990 from Purdue University, is currently an Architect at Cadence Design Systems, Inc.

M6pm: Tcl Internationalization and Localization
Mark Harrison, AsiaInfo Computer Networks (Beijing), Ltd.

Who should attend: Programmers and software designers interested in writing programs that can run in multilingual environments, or who are distributing their software products in an international market.

Tcl and Tk have long had the reputation of making difficult tasks easy. Tcl/Tk 8.1 extended this tradition to internationalization (I18N) and localization (L10N). This course will show you how to create applications that can easily be ported from one (human) language to another, including 16-bit languages such as Chinese and Japanese.

Topics include:

- I18N background
- Strategies for software I18N
- Unicode and character sets: what's the big deal?
- Using the Tcl message catalog
- Tk fonts and character sets
- Environment specifics: X11, Win 9x/NT
- Why it's easy to build a pyramid if you're a pharaoh

Mark Harrison has been involved in I18N programming since 1985, when he wrote his first multi-ingual messaging system and ported it to the Japanese 5550PC display system. A Tcl/Tk enthusiast since hearing John Ousterhout speak in 1991, he developed and contributed the message catalog code to Tcl 8.1. He is the editor of Tcl/Tk Tools and coauthor of Effective Tcl/Tk. He is presently Chief Software Architect at AsiaInfo Computer Networks in Beijing, China.

Tuesday, February 15, 2000

Morning Sessions (9:00 am - 12:30 pm)

T1am: Building Applications with BLT
George Howlett, Cadence Design Systems, Inc.

Who should attend: Applications developers and programmers building real-world applications with Tcl/Tk. Participants should be familiar with Tcl/Tk, but expertise is not required.

The BLT toolkit is a treasure chest of useful components for Tcl/Tk applications. It contains essential widgets, such as a tabbed notebook and a hierarchical listbox, which are not part of the standard Tk distribution. For scientific or business applications, BLT has plotting widgets that support X-Y graphs, strip charts, and bar charts. BLT also adds support for drag-and-drop operations, handling "busy" windows, smooth image scaling, compositing encapsulated PostScript, and much more. Like Tcl/Tk, BLT runs cross-platform on UNIX and Windows 95/98/NT machines.

In this tutorial, students will get an overview of the BLT components and see how those components can be used to build better Tcl/Tk applications.

George Howlett has been using Tcl/Tk for more than 10 years as a Member of Technical Staff at Bell Laboratories, developing software for Electronic Computer-Aided Design (ECAD). He is an ardent fan of Tcl and has taught more than 50 Tcl/Tk courses. He is also the author of the BLT Toolkit, an extension that adds widgets and commands to Tcl/Tk. His chapter on "The BLT Toolkit" is part of Tcl/Tk Tools, published by O'Reilly & Associates. George is currently a Senior Consultant at Cadence Design Systems, Inc.

T2am: Embedding Tcl in C/C++ Applications
D. Richard Hipp, Hwaci

Who should attend: This tutorial is useful for anyone who wants to mix Tcl or Tcl/Tk with C or C++ in a single application. Participants should be familiar with C/C++ programming and have at least some experience in writing Tcl/Tk scripts.

These days, most people think of Tcl/Tk as a programming language, but Tcl was originally created as a C library, and it remains by far the best C library ever invented for building user interfaces.

This tutorial will describe how to use Tcl/Tk as a library instead of as a language.

Topics include:

- A roadmap of the C API to Tcl/Tk
- Procedures for creating, initializing, and controlling Tcl interpreters from within C/C++ programs
- Ways to invoke Tcl procedures from C/C++ code and to invoke C/C++ procedures from within Tcl scripts
- Tricks for embedding Tcl/Tk scripts in C/C++ programs as static strings
- Guidelines for deciding when to use Tcl/Tk and when to use C/C++
- Simple techniques for writing C/C++ programs that employ graphical user interfaces
- Methods for using the Tcl/Tk library as a portability layer so that identical source code can be compiled to run on all flavors of UNIX and on Windows95/98/NT
- Compilation procedures to turn Tcl/Tk scripts into standalone executables that are not readable by the end user and that run on machines which do not have Tcl/Tk installed

Since 1994, **D. Richard Hipp** has been writing C/C++ programs that embed Tcl/Tk. He has written numerous publicly available Tcl extensions, libraries, and programming tools, including Embedded Tk, MkTclApp, PtTcl, TkHtml, and Plot3D. He is one of the co-authors of Tcl/Tk Tools (O'Reilly & Associates). Richard has a Ph.D. in Computer Science from Duke University (1992) and is a founder of Hwaci, a software development consulting firm based in Charlotte, N.C.

T3am: Web Tcl Complete
Steve Ball, Zveno Pty. Ltd.

Tcl is used for every aspect of Web programming from server-side right through to client-side, including Web servers and Web browsers, and all the tools in between.

This tutorial discusses how Tcl may best be used for all of these purposes. Steve Ball's Web Tcl Complete (McGraw-Hill) is used as a guide.

Topics include:

- Safe-Tcl
- Server-side CGI programming
- CGI-less scripting using NeoWebScript, the Tcl Web Server, and AOLServer
- The Tcl plug-in
- Building your own Web application: the HTTP package and HTML libraries
- TclBlend and Jacl

Steve Ball is Chief XML and Scripting Specialist at Australia-based Zveno Pty. Ltd. and is the chief architect of Zveno's Swish XML editor. He is the author of Web Tcl Complete (McGraw-Hill) and several Open Source software projects, including TclExpat, TclXML, TclDOM, and the Plume Web browser. Steve was first introduced to Tcl in 1993 and has hardly touched any other language since then.

Afternoon Sessions (1:30 pm - 5:00 pm)

T4pm: Regular Expressions and Other Parsing Mysteries
Tanya Gallagher, Scriptics Corporation

Who should attend: Beginning to intermediate Tcl programmers with some exposure to basic regular expressions who are interested in learning the new text parsing commands in recent releases of Tcl.

For years even Tcl enthusiasts were forced to admit Tcl's lack of extended regular expression (RE) support. In Tcl 8.1, Henry Spencer revitalized Tcl with a more powerful RE engine that supports advanced regular expressions. Tcl 8.2 extended the string command to help Tcl programmers write clear, concise string processing code.

This tutorial focuses on writing efficient and readable string processing scripts in Tcl with the new RE and string features. If you have ever struggled to locate a single line of text, to match a string with a specific number of digits, or to validate a string as a double, this tutorial will teach you what you need to know.

Topics include:

- Using character classes and non-greedy quantifiers to say what you mean in writing RE patterns
- Validating input with 8.2 string commands
- Writing scripts that work with Unicode
- Performance analysis of commonly written Tcl string processing code
- Tips for maximizing efficiency

Tanya Gallagher is Scriptics' Training Manager. She has updated and written curriculum for Scriptics' Tcl training and teaches Tcl to training customers. Before joining Scriptics, she was a Currriculum Developer/Technical Instructor at Informix Software, where she specialized in data warehousing tools, primarily those that deal with data extraction and transformation. Tanya

holds a BA in industrial/organizational psychology and an MA in instructional design with a curriculum development emphasis, both from San Jose State University.

T5pm: Tcl Internals: Raw and Exposed
Lee Bernhard, Scriptics Corporation

Who should attend: Tcl extension developers and Tcl programmers interested in learning about the Tcl C internals. Students should have a solid background in Tcl and C programming.

In Tcl, everything is a string-sort of. Tcl 8.x allows you to write scripts which pretend that everything is represented as a string, while it silently uses a dual-ported object system to cache native types. Tcl's byte compiler allows script writers the convenience of an interpreted language during the development process, while generating byte codes to speed evaluation of procedures and loops. This tutorial looks behind the curtain to explain the Tcl internals, and shows how to write Tcl and C code that takes advantage of the new architecture.

Topics include:

- How to write extensions that make use of Tcl_Obj'ects for caching native types
- How to invent new object types to aid you in representing your own structures in Tcl
- How to avoid writing scripts that cause data type shimmering that negates the advantages of the object system

The second part of the tutorial will focus on additional issues in extending the newer Tcl interpreters. Additional topics include:

- How to write code that takes advantage of the byte compiler
- How to add a Tcl interpreter to an existing C application
- How the stubs architecture allows you to write extensions that port across Tcl versions without requiring serious modifications

The course materials will include code samples that illustrate the proper use of these features. After completing this tutorial, participants will be aware of how to extend Tcl in a way that takes maximum advantage of the new internal architecture.

Lee Bernhard has been involved in the Tcl community for five years as an instructor and developer. He has taught thousands of hours of Tcl/Tk courses at Bell Labs, and he founded the Scriptics Tcl training organization. He has recently joined the Scriptics engineering team to again concentrate on development. His engineering projects include the Windows port of [incr Tcl], and adding multi-threaded support to Tk. Lee's computing interests include scripting, object-oriented languages, and distributed systems. Lee holds a BS from Duke University in Computer Science and Cognitive Psychology.

T6pm: Expect-Automating Interactive Applications
Don Libes, NIST

Who should attend: This tutorial is appropriate for all users, system administrators, and system programmers. Tcl knowledge is helpful. Attendees will come away with practical knowledge that can be immediately applied to solve problems and save time in your day-to-day UNIX use.

This tutorial explains how to automate interactive programs such as telnet, ftp, passwd, rlogin, and hundreds of other applications that normally require human interaction. Using Expect to automate these applications will allow users to speed up tasks and, in many cases, create solutions that would never have been considered before.

The tutorial will also cover how to test interactive programs and how to connect interactive applications together with no changes to the underlying programs or access to the original source-a common problem for legacy applications or sites without source code-but of value even with source code.

The tutorial will demonstrate how to wrap interactive programs with Motif-like front ends using Tk to control applications by buttons, scrollbars, and other graphic elements. Attendees will also learn how to reuse interactive programs in Web applications without rewriting existing code.

Both total and partial automation will be covered. Attention will be paid to showing how to automate and move interactive tasks into the background securely and reliably.

Don Libes is the creator of Expect as well as the author of its definitive text, Exploring Expect (O'Reilly, 1995). Don has written over 80 computer science papers and articles plus two classic UNIX books: Life with UNIX (Prentice Hall) and Obfuscated C and Other Mysteries (Wiley). In his day job, Don is a computer scientist at the National Institute of Standards and Technology.