The following paper was originally published in the
Proceedings of the Fifth Annual Tcl/Tk Workshop
Boston, Massachusetts, July 1997

Web Enabling Applications

Brent Welch and Stephen Uhler
Sun Microsystems Laboratories
Mountain View, CA

plates. They can develop the surrounding HTML in a WYSIWYG environment and easily add and tune the calls to the aliases. Open APIs are used to create the aliases and integrate them into the authoring environment.

The demo features SNMP device management [SNMP] via a Web browser. The server uses the Scotty SNMP Tcl extension to access and control SNMP-based network devices [Schoenwaelder95]. This application uses the HTML+Tcl templates. It defines a content type, `application/x-snmp`, that processes the query data in an SNMP-specific way and then uses the regular template mechanism. It uses sessions to record which device is being accessed. Groups within a session are used to control the data-driven SNMP module. SNMP uses a database, or MIB, that describes the device. Session groups are used on different pages to record what part of the MIB is being displayed and how it is displayed. The application includes a general MIB browser that lets you navigate the MIB to examine different parts of the SNMP state of the device.

The Tcl Web server is designed to be a flexible mechanism that facilitates web-access to new and existing applications. It leverages the embeddability and extensibility of Tcl so it is easy to add the server to your application. The architecture supports a variety of ways to map URL accesses to the application. In particular, the HTML+Tcl templates for dynamic page generation provide a rich infrastructure for building the web-based interface to your application.

Availability

The server is available for evaluation purposes. We hope to have a freely available version available soon. Please contact the authors for more information, or check the web site at `http://sunscript.sun.com/products/tclhttpd/`.

References

MIME93 RFC-1521. MIME (Multipurpose Internet Mail Extensions). N. Borenstein.

Ousterhout94 J. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley, April 1994, ISBN 0-201-63337-X.

Schoenwaelder95 Schoenwaelder, Langendorfer, *Tcl Extensions for Network Management Applications*. Proc. of the 3rd Tcl/Tk Workshop. ftp://ftp.ibr.cs.tu-bs.de/pub/local/papers/tcltk-95.ps.gz.

SNMP RFC 1155, RFC 1157. Simple Network Management Protocol.

Welch95 B. Welch, *Practical Programming in Tcl and Tk*. Prentice Hall, June 1997 (2nd Ed.), ISBN 0-13-616830-2.

Welch96 B. Welch and S. Uhler. *Tcl/Tk HTML Tools*, Proc. of the 4th Tcl/Tk Workshop. July 1996. pp 173-182.

Web Enabling Applications

Brent Welch
Stephen Uhler
{bwelch,suhler}@eng.sun.com
Sun Microsystems Laboratories
2550 Garcia Ave. MS UMTV29-232
Mountain View, CA 94043

Abstract

This demo shows a Tcl-based Web server that can be extended with application-specific modules to support a variety of applications. The server is written as a pure Tcl script[Ouster94][Welch95] that is modular, extensible, and embeddable. Clients use a regular Web browser to make HTTP requests to the Tcl server. The server can be directly embedded into existing applications, or it can act as a stand-alone server that includes application-specific modules. The server web-enables your application so it can be accessed through a standard Web browser.

The architecture of the server allows interception of URL requests at two levels. At the top-level, the application can implement a whole sub-tree of the URL hierarchy. In this case the application is free to implement whatever semantics it wants for those URLs. The server provides an `Direct` module that works at this level. It maps URL requests directly into calls on Tcl procedures. The name of the URL determines the name of the Tcl command, and query data from forms is bound to procedure arguments. The return value of the Tcl procedure is returned as HTML data to be displayed in the browser.

For example, the URL `/mail/bugreport` is bound to `Mail_/mail/bugreport`, and this procedure sends its form data to the registered webmaster for the server. Note that it is not possible to invoke arbitrary Tcl commands in the server via this interface.

The second level of URL processing supports handlers for different document types. This level is supported by the `Doc` module that implements file-based URLs. The file name suffix is used to select a MIME content type [MIME93], and handlers can be defined for different content types. This level is used to support standard CGI, server-side includes, and imagemaps. Applications can define new file content types and register handlers by defining appropriately named Tcl commands.

The `application/x-tcl-subst` content type is used for hybrid HTML+Tcl documents. These pages are HTML templates that include inline Tcl commands. The server processes the templates in Safe-Tcl interpreters. Side effects from the Tcl processing control the application, and the results are inserted into the Web page viewed on the clients. HTML forms allow input, and specialized displays can use Tclets or Java applets.

The server supports a series of page accesses by the same client, which are called a session. A session is identified by query data posted with the URL. Each session maps to an instance of a Safe-Tcl interpreter. The HTML+Tcl templates are processed in the interpreter for the session. Creating a session creates an interpreter and installs application-specific aliases. The aliases are intended to be the primary Tcl commands used in the HTML+Tcl templates.

Sessions are further subdivided into groups (for lack of a better name) that collect session state information. A group maps to a Tcl array within the per-session interpreter. A group may be associated with a page or shared by different pages, and a page could have several groups. Query data can be bound to groups, automatically updating the Tcl arrays based on query data.

The server supports other standard features like logging and authentication mechanisms. These are implemented by modules that can be tuned for application-specific needs.

The system includes an authoring environment for the HTML+Tcl templates that is based on the WebTk Tcl/Tk HTML editor [Welch96]. Content developers are presented with a simple property sheet interface to aliases that are used in the tem-

plates. They can develop the surrounding HTML in a WYSIWYG environment and easily add and tune the calls to the aliases. Open APIs are used to create the aliases and integrate them into the authoring environment.

The demo features SNMP device management [SNMP] via a Web browser. The server uses the Scotty SNMP Tcl extension to access and control SNMP-based network devices [Schoenwaelder95]. This application uses the HTML+Tcl templates. It defines a content type, `application/x-snmp`, that processes the query data in an SNMP-specific way and then uses the regular template mechanism. It uses sessions to record which device is being accessed. Groups within a session are used to control the data-driven SNMP module. SNMP uses a database, or MIB, that describes the device. Session groups are used on different pages to record what part of the MIB is being displayed and how it is displayed. The application includes a general MIB browser that lets you navigate the MIB to examine different parts of the SNMP state of the device.

The Tcl Web server is designed to be a flexible mechanism that facilitates web-access to new and existing applications. It leverages the embeddability and extensibility of Tcl so it is easy to add the server to your application. The architecture supports a variety of ways to map URL accesses to the application. In particular, the HTML+Tcl templates for dynamic page generation provide a rich infrastructure for building the web-based interface to your application.

Availability

The server is available for evaluation purposes. We hope to have a freely available version available soon. Please contact the authors for more information, or check the web site at `http://sun-script.sun.com/products/tclhttpd/`.

References

MIME93 RFC-1521. MIME (Multipurpose Internet Mail Extensions). N. Borenstein.

Ousterhout94 J. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley, April 1994, ISBN 0-201-63337-X.

Schoenwaelder95 Schoenwaelder, Langendorfer, *Tcl Extensions for Network Management Applications*. Proc. of the 3rd Tcl/Tk Workshop. ftp://ftp.ibr.cs.tu-bs.de/pub/local/papers/tcltk-95.ps.gz.

SNMP RFC 1155, RFC 1157. Simple Network Management Protocol.

Welch95 B. Welch, *Practical Programming in Tcl and Tk*. Prentice Hall, June 1997 (2nd Ed.), ISBN 0-13-616830-2.

Welch96 B. Welch and S. Uhler. *Tcl/Tk HTML Tools*, Proc. of the 4th Tcl/Tk Workshop. July 1996. pp 173-182.

Web Enabling Applications

Brent Welch
Stephen Uhler
{bwelch,suhler}@eng.sun.com
Sun Microsystems Laboratories
2550 Garcia Ave. MS UMTV29-232
Mountain View, CA 94043

Abstract

This demo shows a Tcl-based Web server that can be extended with application-specific modules to support a variety of applications. The server is written as a pure Tcl script[Ouster94][Welch95] that is modular, extensible, and embeddable. Clients use a regular Web browser to make HTTP requests to the Tcl server. The server can be directly embedded into existing applications, or it can act as a stand-alone server that includes application-specific modules. The server web-enables your application so it can be accessed through a standard Web browser.

The architecture of the server allows interception of URL requests at two levels. At the top-level, the application can implement a whole sub-tree of the URL hierarchy. In this case the application is free to implement whatever semantics it wants for those URLs. The server provides an `Direct` module that works at this level. It maps URL requests directly into calls on Tcl procedures. The name of the URL determines the name of the Tcl command, and query data from forms is bound to procedure arguments. The return value of the Tcl procedure is returned as HTML data to be displayed in the browser.

For example, the URL `/mail/bugreport` is bound to `Mail_/mail/bugreport`, and this procedure sends its form data to the registered webmaster for the server. Note that it is not possible to invoke arbitrary Tcl commands in the server via this interface.

The second level of URL processing supports handlers for different document types. This level is supported by the `Doc` module that implements file-based URLs. The file name suffix is used to select a MIME content type [MIME93], and handlers can be defined for different content types. This level is used to support standard CGI, server-side includes, and imagemaps. Applications can define new file content types and register handlers by defining appropriately named Tcl commands.

The `application/x-tcl-subst` content type is used for hybrid HTML+Tcl documents. These pages are HTML templates that include inline Tcl commands. The server processes the templates in Safe-Tcl interpreters. Side effects from the Tcl processing control the application, and the results are inserted into the Web page viewed on the clients. HTML forms allow input, and specialized displays can use Tclets or Java applets.

The server supports a series of page accesses by the same client, which are called a session. A session is identified by query data posted with the URL. Each session maps to an instance of a Safe-Tcl interpreter. The HTML+Tcl templates are processed in the interpreter for the session. Creating a session creates an interpreter and installs application-specific aliases. The aliases are intended to be the primary Tcl commands used in the HTML+Tcl templates.

Sessions are further subdivided into groups (for lack of a better name) that collect session state information. A group maps to a Tcl array within the per-session interpreter. A group may be associated with a page or shared by different pages, and a page could have several groups. Query data can be bound to groups, automatically updating the Tcl arrays based on query data.

The server supports other standard features like logging and authentication mechanisms. These are implemented by modules that can be tuned for application-specific needs.

The system includes an authoring environment for the HTML+Tcl templates that is based on the WebTk Tcl/Tk HTML editor [Welch96]. Content developers are presented with a simple property sheet interface to aliases that are used in the tem-