AJUBA SOLUTIONS

2593 Coast Avenue Mountain View, CA 94043
TEL 650-210-0100 FAX 650-210-0101
www.ajubasolutions.com

Tcl Update

Tcl'Europe, June 15th, 2000

Agenda

- Developments at Scriptics/Ajuba
- Ajuba and the Tcl Community
- Current state of Tcl
- Building the community
- Tcl roadmap

AJUBA SOLUTIONS

Ajuba Update

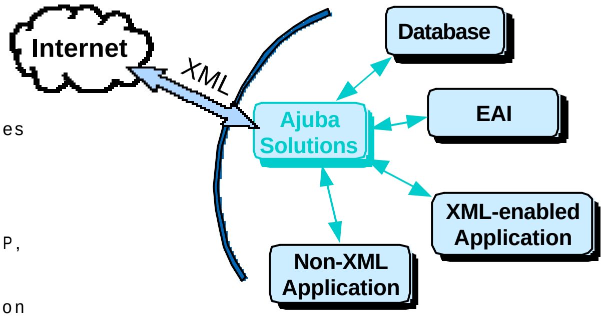
- Founded as Scriptics in January 1998 to build business around Tcl:
  - Continue development of free core
  - Create development tools
  - Build high-value integration platform around Tcl
  - Started with 8 people
  - TclPro development tools and Tcl training
- Ajuba Today
  - Name changed May 22nd, 2000
  - > 50 people
  - B2B infrastructure company
  - New product: Ajuba2

Ajuba Vision

- We build solutions for automating, managing, and customizing high-value B2B relationships over the Internet
- Focused around B2B relationships:
  - Humans create relationships, define parameters
  - We automate implementation/fulfillment of relationships (e.g. supply chain automation)
- Ajuba enables high-value relationships:
  - Integration: reduce costs, errors
  - Management: information transparency, better decisions
  - Customization: higher value, new services, customer retention

Ajuba2

- Connects enterprise resources with:
  - Internet trading communities
  - Business partners
- Software server:
  - Document transport: HTTP, FTP, etc.
  - XML parsing and generation
  - Security (X.509, etc.)
  - Task management
  - Integration with other applications
  - Tcl scripting engine for business rules
- Development tools
- Management tools



Role of Tcl in Ajuba2

- A juba2 built on Tcl
  - A juba2 Server based on tclhttpd
  - Business rules described with Tcl scripts
  - Development tools built with Tcl
  - Development tools extensible with Tcl
- Tcl provides rapid development:
  - We can build and evolve Ajuba2 rapidly
  - Customers/integrators can develop applications quickly
  - Can build powerful management tools
  - Can create highly customized B2B relationships for maximum value

Other Ajuba Products

- TclPro development tools
  - Commercial development tools for Tcl:
    - Debugger
    - Checker
    - Compiler
    - Wrapper
  - Current release: 1.3
  - Ajuba will continue to make additional releases:
    - More tools needed for Ajuba
    - Will sell tools separately as TclPro
- Professional services:
  - TclCare support contracts
  - Tcl training classes
  - Tcl consulting

Ajuba and the Tcl Community

A difficult balancing act:

- Success of company depends on commercial products, especially Ajuba2:
  - Most resources must go to commercial products
- Success of company depends on Tcl:
  - Need vibrant open source community
- Ajuba will continue to support the Tcl community …

  but we can't do it alone!

Ajuba and the Tcl Community

- Ajuba will help empower the Tcl community:
  - Tcl developers' Web site
  - CVS repositories
  - Extension archive
  - Bug database
  - Discussion groups
  - Tcl Ambassador
- Ajuba will continue to manage Tcl/Tk core:
  - New releases
  - Incorporate changes from community
- How the community can help:
  - Build new extensions and packages
  - Participate in discussion groups
  - Fix bugs, create new features

Tcl Core Development Team

Two Ajuba engineers dedicated to open source Tcl/Tk core:

- Jeff Hobbs
  - Tcl Ambassador, started 8/1999
- Eric Melski
  - Software Engineer, started 1/2000
- Ajuba will cycle additional engineers over time

Tcl Today

- Download rates from Ajuba:
  - 30,000 Tcl downloads/month
    - 55% Windows
    - 40% Unix
    - 5% Macintosh
- Steady over the last year
- Estimated developer community 500,000 - 1,000,000

Tcl/Tk Core Releases

- Tcl/Tk 8.1:
  - Development resumed Fall 1998
  - Internationalization
  - Thread support in Tcl
  - Improved regular expressions
- Tcl/Tk 8.2:
  - Released August 1999
  - Various small new features
    - Additional string commands
    - Stacked I/O channels
    - Improved TEA support
  - Completely Pure’ified by 8.2.3
  - Numerous minor bug fixes

Tcl/Tk Core Releases: 8.3

- Stable release February 10, 2000
- Numerous command enhancements
  - clock, file, lsort, glob, array, scan, regexp
- Improved out-of-the-box build on numerous platforms
- Improved error messages, especially for common programming errors
- Several more Tk commands objectified
- Improved Mac and Windows wm support
  - Support for zoomed windows
- Enhanced listbox, canvas widgets and Unix dialogs

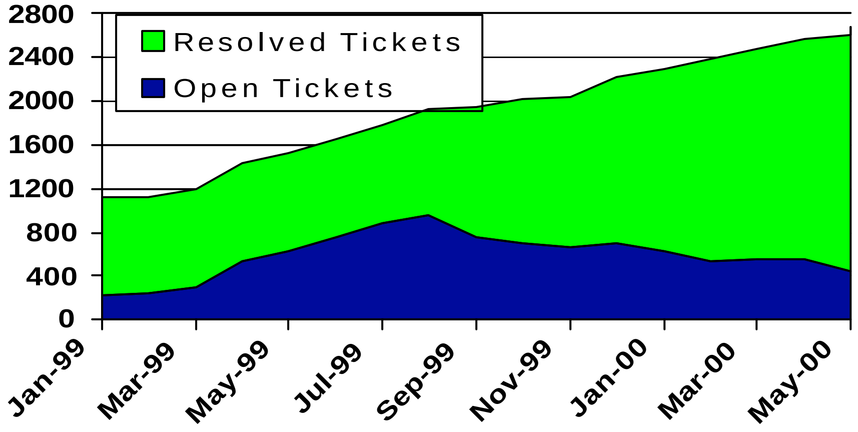
Tcl/Tk 8.3 (cont.)

- Integration of Dash and Img patches
  - support for binary data in -data image option
  - adds an alpha channel to all image types
  - GIF writing capability
  - Canvas now supports dashed items, as well as disabling items
  - Enhanced tag searching capabilities
  - Pointer warping via the event command
  - Simplified entry widget validation
  - Support for eliding in text widget (a la TkMan)
  - Postscript support for widgets and images in canvas

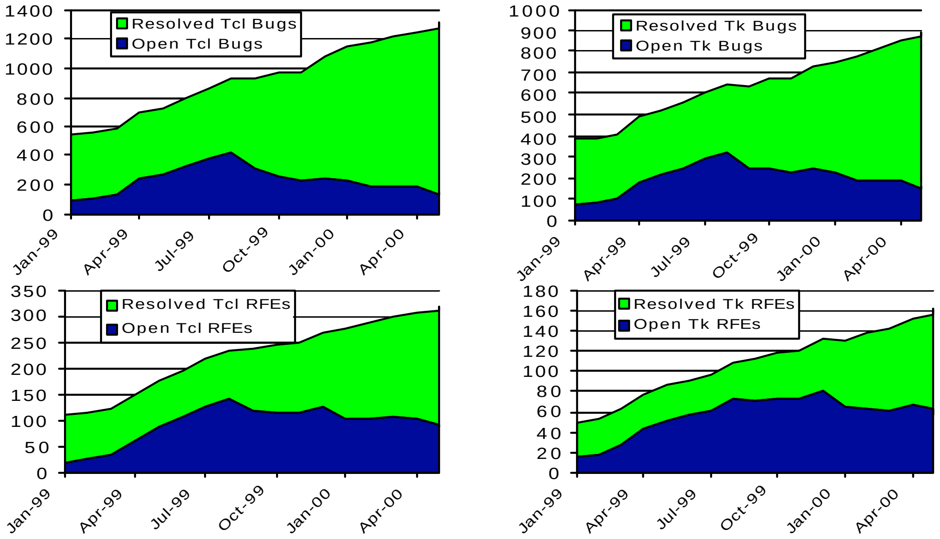
Tcl/Tk Core Releases: 8.4

- First alpha June 2000
- Numerous speed optimizations*
  - Byte-compiled string, return, ... commands
- Enhanced lsearch command
- New expr eq and ne operators
- Improved threading API (Kupries)
- Support for Scripted Documents
- New spinbox widget
- Enhanced *button, entry, label widgets
- Much more to come...

Bug Overview

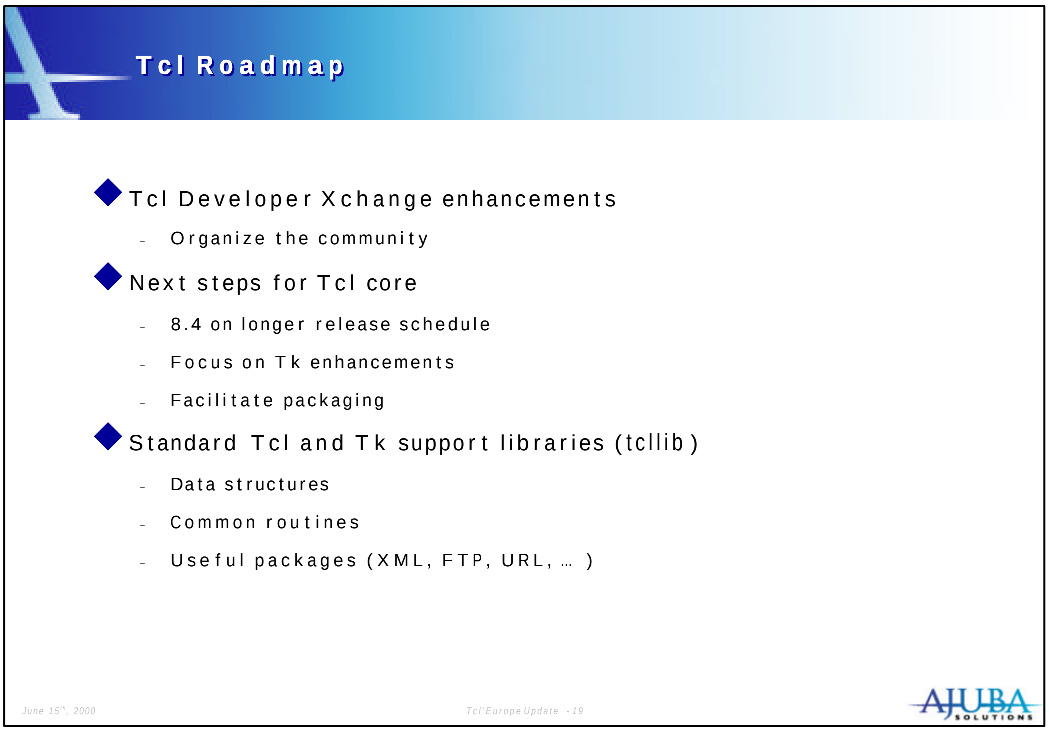


Bug Statistics



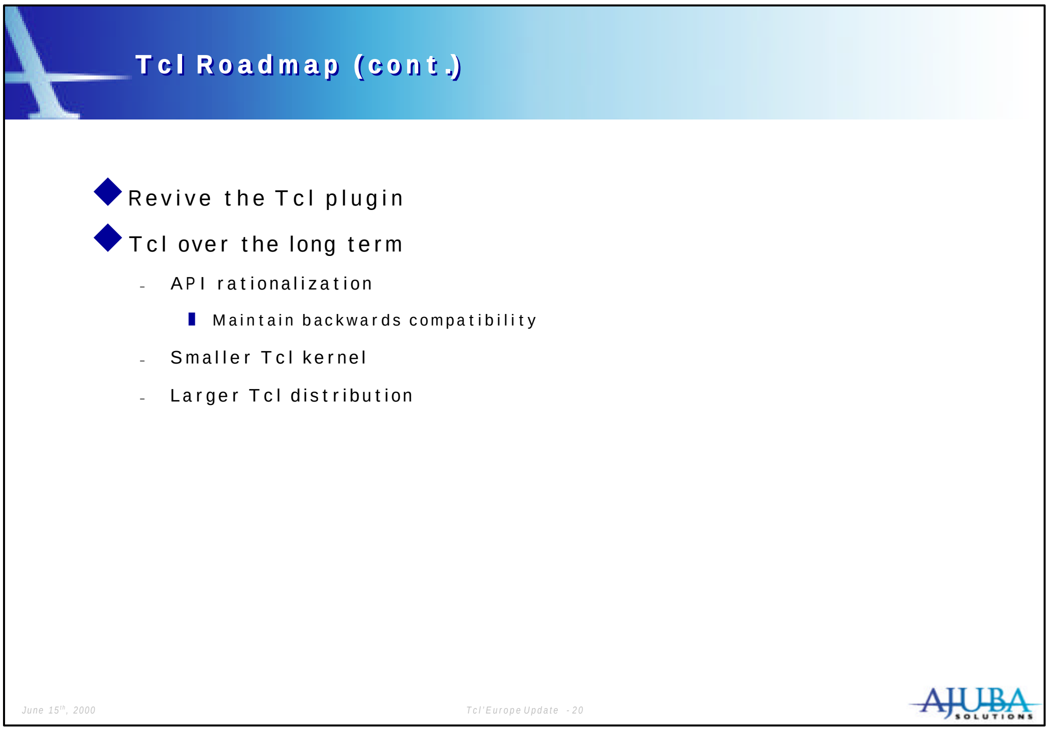
Tcl Developer Xchange

- http://dev.ajubasolutions.com/
- Home for Tcl developers of all experience levels
- New hardware and software
  - Based on AOLServer and ArsDigita community system
- Special Interest Groups (SIGs) and Q&A forums
- Bug database online
- Core downloads, plus links to extensions and resources
- Documentation, HowTos and more



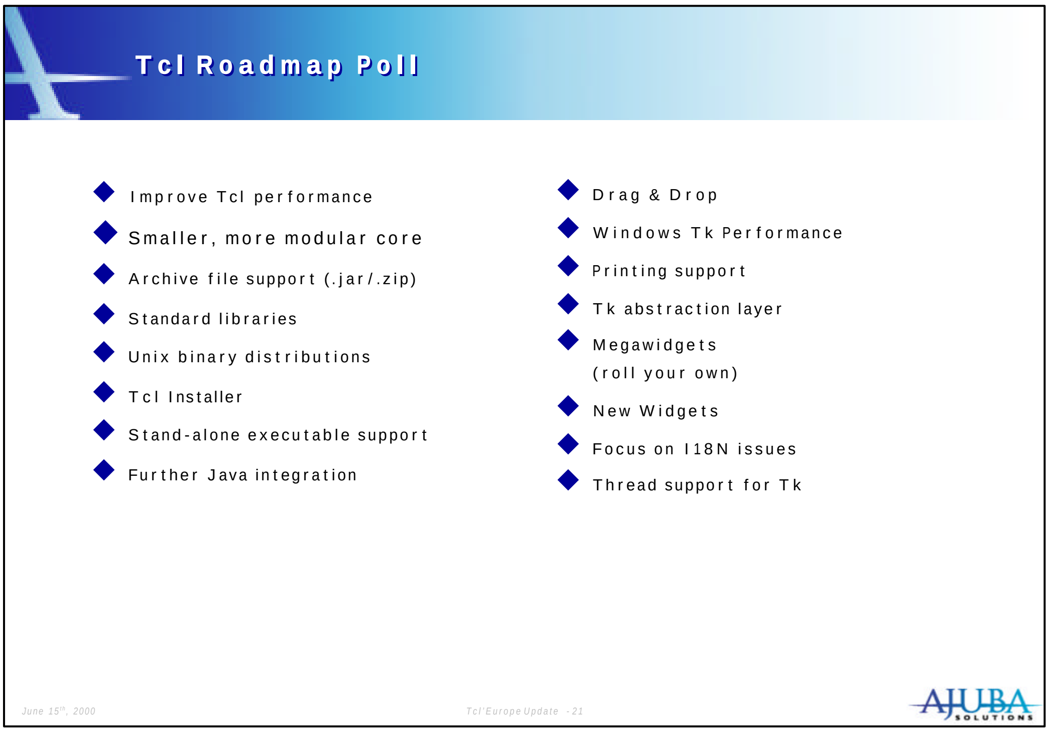
Tcl Roadmap

- Tcl Developer Xchange enhancements
  - Organize the community
- Next steps for Tcl core
  - 8.4 on longer release schedule
  - Focus on Tk enhancements
  - Facilitate packaging
- Standard Tcl and Tk support libraries (tcllib)
  - Data structures
  - Common routines
  - Useful packages (XML, FTP, URL, …)



Tcl Roadmap (cont.)

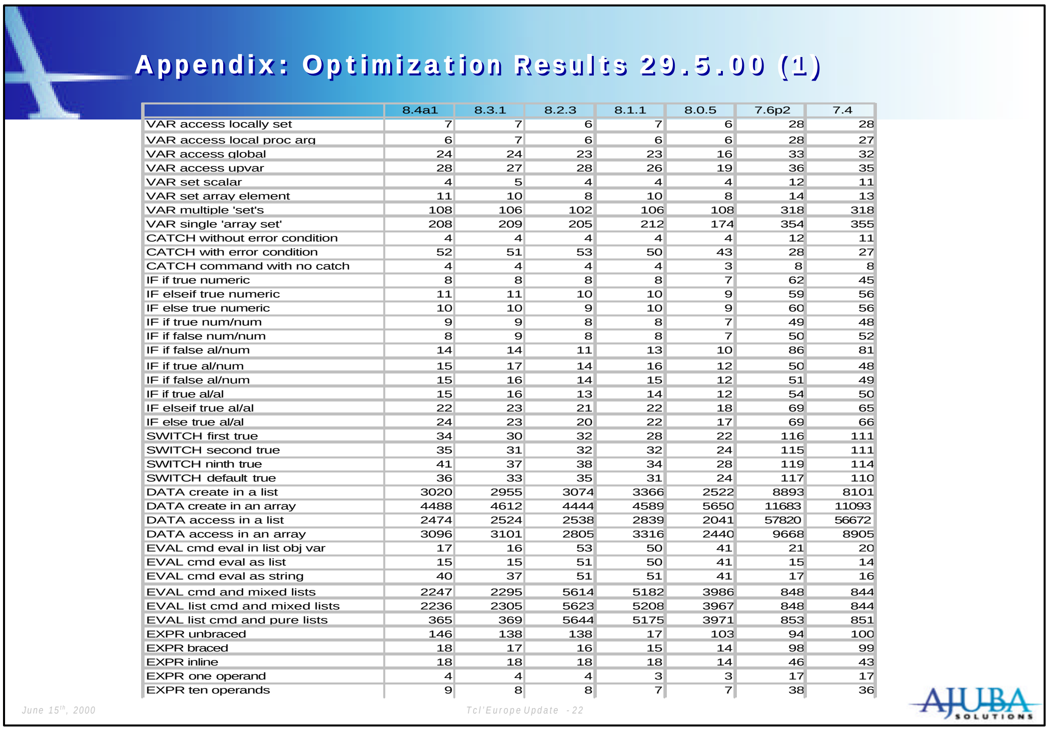
- Revive the Tcl plugin
- Tcl over the long term
  - API rationalization
    - Maintain backwards compatibility
  - Smaller Tcl kernel
  - Larger Tcl distribution



Tcl Roadmap Poll

- Improve Tcl performance
- Smaller, more modular core
- Archive file support (.jar/.zip)
- Standard libraries
- Unix binary distributions
- Tcl Installer
- Stand-alone executable support
- Further Java integration

- Drag & Drop
- Windows Tk Performance
- Printing support
- Tk abstraction layer
- Megawidgets
  (roll your own)
- New Widgets
- Focus on I18N issues
- Thread support for Tk



Appendix: Optimization Results 29.5.00 (1)

	8.4a1	8.3.1	8.2.3	8.1.1	8.0.5	7.6p2	7.4
VAR access locally set	7	7	6	7	6	28	28
VAR access local proc arg	6	7	6	6	6	28	27
VAR access global	24	24	23	23	16	33	32
VAR access upvar	28	27	28	26	19	36	35
VAR set scalar	4	5	4	4	4	12	11
VAR set array element	11	10	8	10	8	14	13
VAR multiple 'set's	108	106	102	106	108	318	318
VAR single 'array set'	208	209	205	212	174	354	355
CATCH without error condition	4	4	4	4	4	12	11
CATCH with error condition	52	51	53	50	43	28	27
CATCH command with no catch	4	4	4	4	3	8	8
IF if true numeric	8	8	8	8	7	62	45
IF elseif true numeric	11	11	10	10	9	59	56
IF else true numeric	10	10	9	10	9	60	56
IF if true num/num	9	9	8	8	7	49	48
IF if false num/num	8	9	8	8	7	50	52
IF if false al/num	14	14	11	13	10	86	81
IF if true al/num	15	17	14	16	12	50	48
IF if false al/num	15	16	14	15	12	51	49
IF if true al/al	15	16	13	14	12	54	50
IF elseif true al/al	22	23	21	22	18	69	65
IF else true al/al	24	23	20	22	17	69	66
SWITCH first true	34	30	32	28	22	116	111
SWITCH second true	35	31	32	32	24	115	111
SWITCH ninth true	41	37	38	34	28	119	114
SWITCH default true	36	33	35	31	24	117	110
DATA create in a list	3020	2955	3074	3366	2522	8893	8101
DATA create in an array	4488	4612	4444	4589	5650	11683	11093
DATA access in a list	2474	2524	2538	2839	2041	57820	56672
DATA access in an array	3096	3101	2805	3316	2440	9668	8905
EVAL cmd eval in list obj var	17	16	53	50	41	21	20
EVAL cmd eval as list	15	15	51	50	41	15	14
EVAL cmd eval as string	40	37	51	51	41	17	16
EVAL cmd and mixed lists	2247	2295	5614	5182	3986	848	844
EVAL list cmd and mixed lists	2236	2305	5623	5208	3967	848	844
EVAL list cmd and pure lists	365	369	5644	5175	3971	853	851
EXPR unbraced	146	138	138	17	103	94	100
EXPR braced	18	17	16	15	14	98	99
EXPR inline	18	18	18	18	14	46	43
EXPR one operand	4	4	4	3	3	17	17
EXPR ten operands	9	8	8	7	7	38	36

Appendix: Optimization Results (2)

	8.4a1	8.3.1	8.2.3	8.1.1	8.0.5	7.6p2	7.4
STR/LIST length, obj shimmer	568	574	567	106	10	305	285
LIST length, pure list	9	9	8	9	7	215	215
STR length of a LIST	8	8	9	55	6	94	91
LIST exact search, first item	10	9	9	8	7	191	190
LIST exact search, middle item	20	19	18	18	14	204	205
LIST exact search, last item	42	43	42	40	35	221	218
LIST exact search, non-item	69	64	66	66	57	227	226
LIST sorted search, first item	11	10	9	8	7	190	190
LIST sorted search, middle item	12	18	19	18	14	206	202
LIST sorted search, last item	11	41	43	41	35	219	217
LIST sorted search, non-item	12	63	68	66	59	228	226
LIST exact search, untyped item	44	42	42	40	34	221	216
LIST exact search, typed item	32	41	42	40	35	220	219
LIST sorted search, typed item	11	41	42	41	35	220	217
LIST sort	641	649	657	650	584	897	892
LIST typed sort	523	521	497	490	386	2086	2039
LIST remove first element	47	36	36	36	30	114	112
LIST remove middle element	47	35	35	35	29	151	146
LIST remove last element	46	35	35	34	28	161	160
LIST replace first element	48	35	35	33	27	131	131
LIST replace middle element	48	35	35	35	27	155	153
LIST replace last element	48	35	34	33	28	168	165
LIST remove in mixed list	56	534	529	44	40	148	147
LIST replace in mixed list	54	534	529	45	38	154	152
LIST index first element	8	8	7	8	7	84	89
LIST index middle element	8	8	7	7	7	105	104
LIST index last element	8	8	7	8	7	127	125
LIST insert an item at start	52	38	37	37	27	126	125
LIST insert an item at middle	50	39	36	36	26	149	148
LIST insert an item at "end"	50	37	36	36	25	180	179
LIST small, early range	17	17	17	16	10	92	91
LIST small, late range	17	18	16	16	10	126	125
LIST large, early range	23	23	22	24	16	129	131
LIST large, late range	23	24	22	22	15	141	140
LIST append to list	126	124	123	127	114	129	127
LIST join list	259	261	259	195	170	541	534
LOOP for, iterate list	3457	3435	3324	3896	3292	208686	206423
LOOP foreach, iterate list	1549	1515	1520	1594	1456	5021	4655
LOOP for (to 1000)	2065	2060	1976	2113	1952	15181	14040
LOOP while (to 1000)	2062	2022	1973	2093	1951	15066	14036
LOOP for, iterate string	4341	7013	6519	48817	4476	55746	53750

Appendix: Optimization Results (3)

	8.4a1	8.3.1	8.2.3	8.1.1	8.0.5	7.6p2	7.4
MAP string 1 val	497	5555	5590	4711	(8.2+)	(8.2+)	(8.2+)
MAP string 2 val	1062	7271	7300	6454	(8.2+)	(8.2+)	(8.2+)
MAP string 3 val	1356	8406	8401	7684	(8.2+)	(8.2+)	(8.2+)
MAP string 4 val	1662	9108	9248	8525	(8.2+)	(8.2+)	(8.2+)
MAP string 1 val -nocase	2030	7504	7283	6657	(8.2+)	(8.2+)	(8.2+)
MAP string 2 val -nocase	3649	10689	10500	10088	(8.2+)	(8.2+)	(8.2+)
MAP string 3 val -nocase	5340	14109	13908	13486	(8.2+)	(8.2+)	(8.2+)
MAP string 4 val -nocase	6455	16306	15833	15809	(8.2+)	(8.2+)	(8.2+)
MAP regsub 1 val	2979	3444	3375	3776	571	1786	1901
MAP regsub 2 val	13361	14834	14904	15429	1404	3289	3481
MAP regsub 3 val	17755	20345	20407	20761	1903	4147	4447
MAP regsub 4 val	22446	25692	25869	26062	2391	5022	5477
MAP regsub 1 val -nocase	2961	3356	3392	3771	787	2004	2091
MAP regsub 2 val -nocase	13351	14909	14911	15407	1857	3751	3905
MAP regsub 3 val -nocase	17788	20495	20605	20691	2544	5018	5059
MAP regsub 4 val -nocase	22369	25721	25798	26033	3264	5878	6213
MAP string, no match	748	8366	8385	7442	(8.2+)	(8.2+)	(8.2+)
MAP string -nocase, no match	3881	13436	13237	12827	(8.2+)	(8.2+)	(8.2+)
MAP regsub, no match	845	2160	2181	2190	770	1709	1853
MAP regsub -nocase, no match	837	2139	2176	2201	1215	2169	2252
MAP string short	26	33	33	33	(8.2+)	(8.2+)	(8.2+)
MAP regsub short	150	152	150	145	36	78	118
PROC explicit return	4	6	5	6	5	14	15
PROC implicit return	4	5	4	4	3	12	11
PROC heavily commented	4	4	4	3	3	212	173
PROC do-nothing, no args	3	3	25	23	3	2	2
PROC do-nothing, one arg	3	4	4	3	3	6	5
PROC local links with global	976	985	967	975	949	2728	2659
PROC local links with upvar	855	865	844	840	835	2714	2644
PROC local links with variable	815	825	820	814	469	2715	2629
READ 595K file with gets	278460	272660	265820	236009	123196	245947	173342
READ 595K file with read	55136	60984	54764	59449	340507	338174	46131
READ 595K file with read & size	51399	58067	53167	57077	19229	336494	45161
READ 3050b file with gets	1568	1503	1512	1317	747	1360	993
READ 3050b file with read	370	398	376	388	204	210	228
READ 3050b file with read & size	406	436	406	426	236	243	262
REGEXP literal regexp	31	31	31	34	19	21	21
REGEXP var-based regexp	32	33	32	35	22	28	27
REGEXP count all matches	123	124	425	527	315	693	647
REGEXP extract all matches	153	159	516	640	379	898	863
STARTUP time to launch tclsh	40831	40254	35977	35811	31519	23736	16699

Appendix: Optimization Results (4)

	8.4a1	8.3.1	8.2.3	8.1.1	8.0.5	7.6p2	7.4
STR str [string compare]	10	15	15	16	13	60	59
STR str [string equal]	10	16	14	28	13	59	59
STR str $a equal ""	9	16	14	28	12	98	89
STR str num == ""	10	10	13	14	12	98	89
STR str $a eq $b	16	22	21	23	18	77	74
STR str $a ne $b	15	22	21	22	17	72	70
STR length (==8010)	8	8	9	357	6	592	593
STR index 0	12	14	14	356	8	604	604
STR index 100	12	14	14	354	9	602	609
STR index 500	12	15	14	357	8	605	602
STR first (success)	11	12	12	11	8	535	536
STR first (failure)	110	128	128	128	128	615	613
STR first (total failure)	90	82	81	81	80	592	607
STR last (success)	11	364	355	355	7	632	623
STR last (failure)	146	117	115	116	114	614	612
STR last (total failure)	108	83	83	83	81	591	588
STR match, simple (success early)	10	11	9	9	7	529	531
STR match, simple (success late)	10	11	9	9	8	529	529
STR match, simple (failure)	10	11	10	9	7	530	530
STR match, simple (total failure)	10	11	10	9	8	529	531
STR match, complex (success early)	11	18	16	16	11	533	533
STR match, complex (success late)	225	1619	1533	1608	635	1197	1200
STR match, complex (failure)	193	1595	1498	1570	625	1189	1194
STR match, complex (total failure)	138	1561	1467	1542	610	1114	1113
STR range	18	17	17	365	9	618	613
STR replace, no replacement	108	488	485	473	82	1351	1352
STR replace, equal replacement	106	487	468	467	84	1366	1365
STR replace, longer replacement	108	492	475	473	83	1382	1384
STR repeat, abcdefghij * 100	56	58	56	42	429	1778	1701
STR append	484	470	464	481	452	866	837
STR info locals match	678	742	718	689	672	2445	2404
TRACE no trace set	25	23	23	23	17	34	33
TRACE read	24	23	24	23	16	33	34
TRACE write	25	23	23	23	17	34	33
TRACE unset	24	23	23	23	17	37	33
TRACE all set (rwu)	24	24	25	25	17	33	33
Unset existing variable	7	7	6	7	5	11	9
Catch invalid unset	57	57	58	55	47	31	30
Catch valid unset	7	8	7	7	5	15	14
Check existence and unset	10	11	11	16	8	21	22
Check nonexistence and unset	8	9	7	8	6	19	19