

TeraTech News

Tools for Programmers

100 Park Avenue Suite #360 Rockville, MD 20850
800-447-9120 • 301-424-3903 • Fax 301-762-8185
<http://www.teratech.com> or info@teratech.com

Lucy Mwinamo, Editor

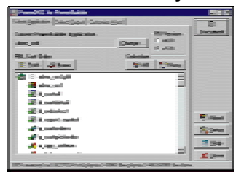
May 2000

This month we cover:

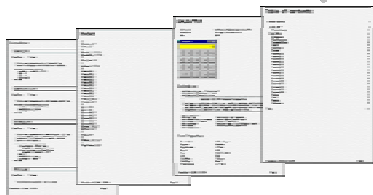
- PowerDOC for Visual Basic
- Replacement for win32api.txt
- Switching from Access to SQL Server
- Bug avoidance
- LCLint
- MD ColdFusion Users Group
- Quote of the Month

PowerDoc For Visual Basic

PowerDOC generates high quality technical documentation in Microsoft Word, including table of contents and index. Select the Visual Basic application to document, and PowerDOC does the rest! PowerDOC automatically extracts all objects, controls,



attributes, pictures, events, script, functions, variables, structures, and comments from the project and produces comprehensive technical



documentation in Microsoft Word. Users can customize PowerDOC by selecting the level of documentation they desire. More Information:

http://www.powerdocinc.com/Business/products/business/productcatalog/business/productpage/business/productpage_VB.htm

Replacement for win32api.txt

Every Win32 API-programmer knows that the win32api.txt-file that comes with Visual Basic is totally out of date. This file is based on the Windows95 header files, and with the arrival of Windows98 and windows2000, many new API-functions,

types and constants were created. That's why the KPD team released a new version of the win32api.txt that's based on the latest Windows2000 header files. This new file does not contain the latest functions and types, because converting types and function-declarations from a header file to Visual Basic isn't as easy as converting constants. At this moment, they're working on a new version that also contains the latest functions and types. So, what's new in this file? Well, this file contains 25535 new constants! The new file comes with a new API-Viewer, API-Toolshed. Why didn't they release a new win32api.txt-file so you can view it with the Microsoft API-Viewer? You may not believe it, but Microsoft's API-Viewer gives an overflow error when you try to load the new API-file. For More info: <http://kpdteam.tripod.com/index.htm?apits/index.htm>

Switching from Access to SQL Server

Some of you may have on your Web servers Access databases that provide some sort of dynamic content or storage facilities for data. There are several reasons for converting to SQL Server:

- i. SQL Server can better handle large amounts of data. Access databases tend to bog down when the files are getting over 50 to 100 MB, whereas SQL Server databases can easily handle that much data and far more.
- ii. SQL Server operates in a different manner than Access and can handle requests much faster and more efficiently. SQL Server databases can also be backed up without

- having to take down the server, which means you'll get better uptime.
- iii. SQL Server is a lot easier to administer remotely than is Access. Instead of having to download and upload the entire database every time, SQL Server lets you make all your changes via Enterprise Manager. If you need help converting Access to SQL, gives us a call.

Bug Avoidance

It is often faster in the long run to spend more time in design and careful coding than extra time debugging. Here are some tips I found on the Internet on bug avoidance.

Chasing down a bug in your code is never fun or easy. In a perfect world, you'd find these bugs before they ever hatch, but that's just not practical. There are a few guidelines you can follow that may help you prevent error prone code:

- Break your code into small, manageable pieces; don't write functions and subs that are hundreds of lines of code.
- Comment abundantly, and make those comments count.
- Explicitly declare all your variables; a misspelled variable can be very difficult to find.
- Adopt and consistently use a naming convention.
- Code defensively to catch bugs where they first occur (i.e. use ELSE and CASEOTHER)

LCLint

LCLint is a tool for statically checking C programs. With minimal effort, LCLint can be used as a better lint. If additional effort is invested adding annotations to programs, LCLint can perform stronger checks than can be done by any standard lint. LCLint does many of the traditional lint checks including unused declarations, type inconsistencies, use-before-definition, ignored return values, execution paths with

no return, likely infinite loops, and fall-through cases. The main focus, however, is on more powerful checks that are made possible by additional information given in source code annotations. Some problems detected by LCLint include:

- Violations of information hiding. A user-defined type can be declared as abstract, and a message is reported where code inappropriately depends on the representation of the type.
- Inconsistent modification of caller-visible state. Functions can be annotated with information on what caller-visible state may be modified by the function, and an error is reported if the modifications produced by the function contradict its declaration.
- Memory management errors. Instances where storage that has been deallocated is used, or where storage is not deallocated (memory leaks).

More info: <http://lclint.cs.virginia.edu/>

MD ColdFusion Users Group.

We host the MDCFUG meeting every second Tuesday of the month. For more information on past meetings and next meetings topics visit:

<http://www.cfug-md.org/>

Quote of the Month

"The net interprets censorship as damage and routes around it."

TERATECH CUSTOM PROGRAMMING

- ◆ Visual Basic, Quick Basic, C/CC++, ASM
- ◆ Access, FoxPro, Clipper, SQL
- ◆ Numeric Analysis, Statistics, Telephony
- ◆ Web HTML, ColdFusion, ASP, Java

Copyright TeraTech 2000

TeraTech, Inc.

100 Park Ave Ste 360

Rockville, MD 20850

800-447-9120 301-424-3903

fax 301-762-8185

<http://www.teratech.com>

info@teratech.com