SPREZZATURA

SENL

February 1999

Sprezzatura's Electronic Newsletter

For Revelation developers by Revelation developers

MAKING DATABASES HAPPEN

SPREZZATURA

Contents

MAKING DATABASES HAPPEN

SPREZZATURA

Welcome

WELCOME - to this issue of S/ENL, the Sprezzatura Electronic Newsletter.

Well I think that it is becoming increasingly apparent that SENL does not have a publication schedule! I think if we aim for every two months we should be able to manage it! Sorry about that.

This issue contains a short but seminal article by Carl Pates on a subject close to the heart of all OI developers – how do I abort an index rebuild? I strongly recommend it to you, we've been using it internally for a while now (specifically for our index rebuild utility that not only keeps you informed of progress but also allows cancellation).

Another couple of hectic months for the Sprezzatura Group. We're doing some sterling work with Web Enabling applications for our clients and in one particularly memorable occasion late last year we were able to get a client onto the Internet with a working solution in five days start to finish! What made it all the more pleasant was the speed with which the project evolved. We got a 'phone call in the office "What do I need to get a working web site?" "A Web license and a T1 line and a bit of programming".

The next day the 'phone goes again "The software's here and the T1 is being installed, when can you be here?" So the week following we had a working application! All parties involved were ludicrously happy, none more so than the client who was able to pint out to his parent company that he'd achieved in three weeks what they'd had a team of four working full time for nine months on!

Speaking of licensing, this issue contains a very informative article on OpenInsight licensing. This is a vexed issue and we hope that our article will help you to ensure that you're in compliance. This is a real issue these days as Revelation Software is understandably keen to protect its development revenue. Just recently a UK-based Rev G-based recruitment software house was found to be pirating Revelation software to a great degree. We helped out on the investigation and found hundreds of thousands of pounds worth of pirated software. Revelation Software aggressively pursued the issue and that company is no longer trading.

The RevSoft road show returns with a vengeance next month and as ever Sprezzatura will be represented at the six locations! In fact as we update this there are only two venues left! Nice to see so many friends again, and Aaron and I look forward to seeing friends old and new again on the East Coast

The work we're doing with TL Creates is approaching fruition and everybody who has seen it cannot believe just how sophisticated the software is for something just written with OICGI, javaScript and HTML! We're assembling quite a little code library that we hope to be able to offer for eventual sale to developers.

At last we've got round to lavishing some TLC on our own web site! It's still not finished but we've tidied it up a lot, checked the links, added new links etc. etc. The US office now has a T1 line so we'll be able to put example apps up and we hope to put up some user stories shortly. Why not check it out and give us some feedback?

MAKING DATABASES HAPPEN

SPREZZATURA

Much joy in the Sprezz office as we all upgrade our kit! The server has gone to a twin PII 350 with 13.5 GB of disk and 256 MB of RAM. We've got new workstations too – all in preparation for jREV! We figure we'll need the higher spec to run it so let's bite the bullet and invest in new kit.

We've just had our favourite assignment of the year so far. A prestigious client calls – "Can you please come in and do a Y2K audit?" "No problem" we think, "chance to try out FinSol's excellent looking package". One of our consultants visits the site to view this AREV application and ten minutes later calls the user across.
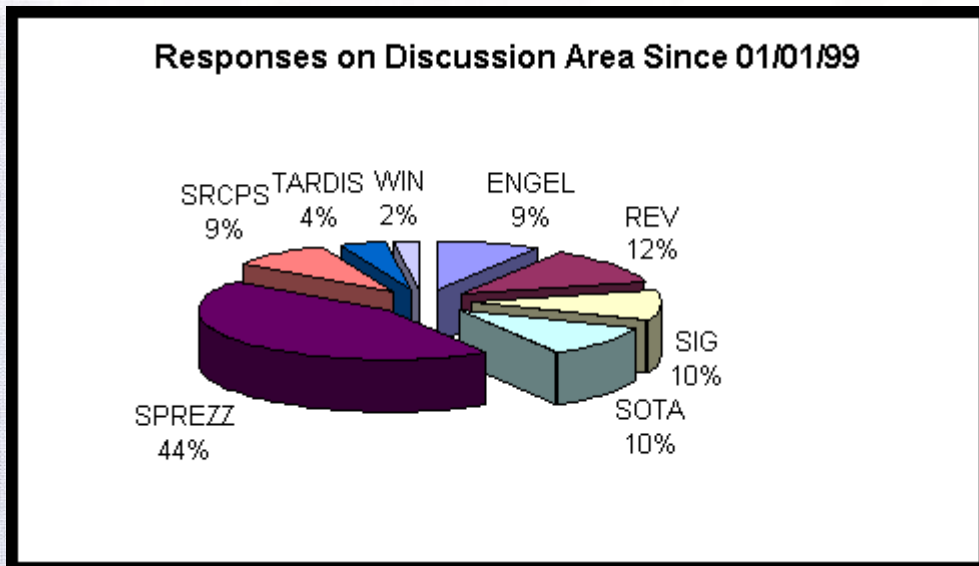
"Ermmm, excuse me." ,"Yes?"

"Well, it seems that this app doesn't actually have any date fields", "Yes, that's right."

"Ermmm... well, we'll gladly certificate it as Y2K compliant then", "Ooh good!"

Cha-ching!

One of our colleagues recently commented on the fact that Sprezz have an awfully large Web presence on the discussion database. Obviously we like to spend our time working as much as the next person so we thought it's be interesting to see just how the various players in the community faired. We downloaded all responses since 1/1/99 and broke them down by company just to see how it'd look. It makes interesting viewing as I'm sure you'll agree! Perhaps this explains why we're picking up so many support contracts recently! If you'd like to talk to us about contracts, either fixed price or "pay per call" just drop us a line on sales@sprezzatura.com or visit http://www.sprezzatura.com/tecsuprt.htm for details! We support all RevSoft products back to Revelation C!

Note that the statistics for the WORKS section invert this relationship with RevSoft doing the lion's share and Sprezzatura sitting more in the wings! And upon that interesting set of statistics - I look forward to seeing a lot of you on the forthcoming RevSoft road show, 'til next issue, ciao for now!

Andrew P McAuley

MAKING DATABASES HAPPEN

SPREZZATURA

Publisher

Feedback

Aaron Kaplan from Sprezz Inc writes (<g>) "I keep getting these beeps from OI on a random basis and it really started to tick me off. Then, I noticed that the system monitor keeps clearing itself as opposed to scrolling off as I originally thought. So, I start to wonder if whenever the system monitor clears, it beeps. Sure enough, when a buffer is reached, the monitor clears and plays the default sound.

Wonder if this is the cause of many a random beep that people experience."

Mykl writes (in non-joke mode for once)

You must follow the rules on this one exactly, otherwise it won't work.

It's really scary how this works out.

NO CHEATING!!!!

First, get a pen and paper.

Second, write the numbers one through six.

Next to number one, write any number...

Next to number two, write the name of anyone to which you are really attracted...

Next to three, write down the first color you can think of...

Next to number four, write the name of your first pet....

Next to number five and six write down the name of a family member...

Remember...no cheating.....

Keep scrolling down........

Don't cheat, or you'll be upset.......

Here's the answers....

The number next to number one show how many times you should be smashed over the head with a baseball bat for thinking that stupid e-mails like this actually mean anything....

The person named next to number two is someone who will never sleep with you because you're stupid enough to waste your time on something like this....

MAKING DATABASES HAPPEN

SPREZZATURA

The color you picked means nothing. It's a friggin' color for *****sake..

Number four gives you the name of a dead animal....

Numbers five and six represent family members who are embarrassed to be related to you....

Meanwhile in response to the rhetorical query "What is meta for?", Philly's finest offered the following Meta is a programming language written by the Ostby brothers that is essentially a rules based engine. As a processing language, the programmer is expected to establish, maintain, manage, manipulate and destroy all memory, stacks and heaps.

With the exception of the above manipulation, each statement is a definition of a rule. A rule can be made up of a series of other rules. By combining these rules, the input is parsed, processed and validated. Optionally, output can be generated based on the inputs. This feature enhanced metas use as a compiler.

Meta's primary function is to validate processes that are based on fixed, rigid rules, for example a sentence parser, a compiler or other syntax checker.

In Revelation's products, meta is used to valid and process OCONVs, generate source code for R/LIST statements and index transaction code, generate object code and symbol tables, handle SQL protection generation, and many other things.

I hope this clarifies what meta is for.

If you need any further information, please don't hesitate to contact us."

A nice little communication from Peter Richards somewhere deep in the Philippines "Many thanks for the articles in the Sprezz SENL newsletters, especially on ODBC/ SQL information. By going through your step by step guide, I was able to (seamlessly <g> <g>) do some SQL queries on MS-Access and Dbase DB's. Thank you !! :)"

Meanwhile in other news – do any of our antipodean readers have a new address for IES Ltd in Rotorua? The information we have seems to be out of date.

A client in Brussels writes to tell us of some exciting work - in his words "You can check out some of our static html pages generated from AREV via AREV on http://www.uia.org/uiademo/demohome.htm. Try our experiments with VRML, which we plan to generate dynamically as well: http://www.uia.org/uiademo/vrml/vrmldemo.htm.

A client from the US writes "I've been seeking an answer from somebody concerning our HRMS software. It's an AREV product, with approx 1500+ fields; anyway, a disaster has occurred and I've got nowhere to go.

The %FIELDS% record in the DICT.EMP file seemingly has lost some data (field names, FMCs, etc...) - I know this, because the length of the data on the field name line (3) is approx 8900 chrs when the "old" good database is 16,960+. Is there any way to "re-compile" the DICTIONARY file so that the fields are somehow magically retabulated or gathered?

MAKING DATABASES HAPPEN

SPREZZATURA

Any help is appreciated..."

In answer -

```
"Delete %FIELDS%

:DELETE DICT YOU*******YOUKILLEDKENNY %FIELDS%

then FORCE a rebuild of %FIELDS%

:COPY DICT YOU*******YOUKILLEDKENNY * TO:(DICT YOU*******YOUKILLEDKENNY (O
```

Note that anything over 1000 fields is going to become WAY too unstable and we wouldn't recommend it even if we were being paid to."

MYKL'S JOKE SECTION (He's back, and this time it's terrible...)

Engineer jokes

An architect, an artist and an engineer were discussing whether it was better to have a wife or a girlfriend. The architect said he enjoyed spending time with his wife, building a solid foundation for an enduring relationship.

The artist said he enjoyed time with his girlfriend because of the passion and mystery he found with her. The engineer said, "I prefer both. With a wife and a mistress, when I go out, they each assume I'm with the other woman, which makes it easy for me to spend all night at the lab!"

A pastor, a doctor and an engineer were waiting one morning for a particularly slow group of golfers. The engineer fumed, "What's with these guys? We must have been waiting for 15 minutes!" The doctor chimed in, "I don't know, but I've never seen such ineptitude!" The pastor said, "Hey, here comes the greenskeeper. Let's have a word with him." Hi George. Say, what's with that group ahead of us? They're rather slow, aren't they?" The greenskeeper replied, "Oh, yes, that's a group of blind firefighters. They lost their sight saving our clubhouse from a fire last year, so we always let them play for free anytime." The group was silent for a moment. The pastor said, "That's so sad. I think I will say a special prayer for them tonight." The doctor said, "Good idea. And I'm going to contact my ophthalmologist buddy and see if there's anything he can do for them." The engineer said, "Why can't these guys play at night?"

An engineering student was walking across campus when another engineer rides up on a shiny new motorcycle. "Where did you get such a great bike?" asked the first. The second engineer replied "Well, I was walking to class yesterday when a girl rode up on this motorcycle. She threw the bike to the ground, took off all her clothes, smiled at me and said 'Take whatever you want.'" The second engineer nodded approvingly, "good choice; the clothes probably wouldn't have fit."

MAKING DATABASES HAPPEN

SPREZZATURA

OpenInsight Licensing Issues – Andrew P McAuley with help from Aaron Kaplan

Originally this article was to have been a reasoned discussion of OpenInsight licensing with great attention spent to ensuring that the information contained herein was as balanced as possible and that RevSoft's viewpoint was clearly but fairly represented. Due to the complexities of the licensing we have opted to simply state our current understanding of RevSoft's stance on their runtime licensing and to conclude with a description of how to implement a single user developer copy on a multi-user runtime system.

According to RevSoft sources, runtime users may not:

- Modify OpenInsight or Application components

- Create/Update/Delete dictionaries

- Create/rename/delete files

- Create/Modify/Delete index definitions

- Create OpenInsight or Application components from scratch

- Create custom ad hoc reports by any means other than a licensed Revelation Reporter

- According to the same RevSoft sources, runtime users may

- Modify OpenInsight or Application components if they are only pre-existing controls within a form.

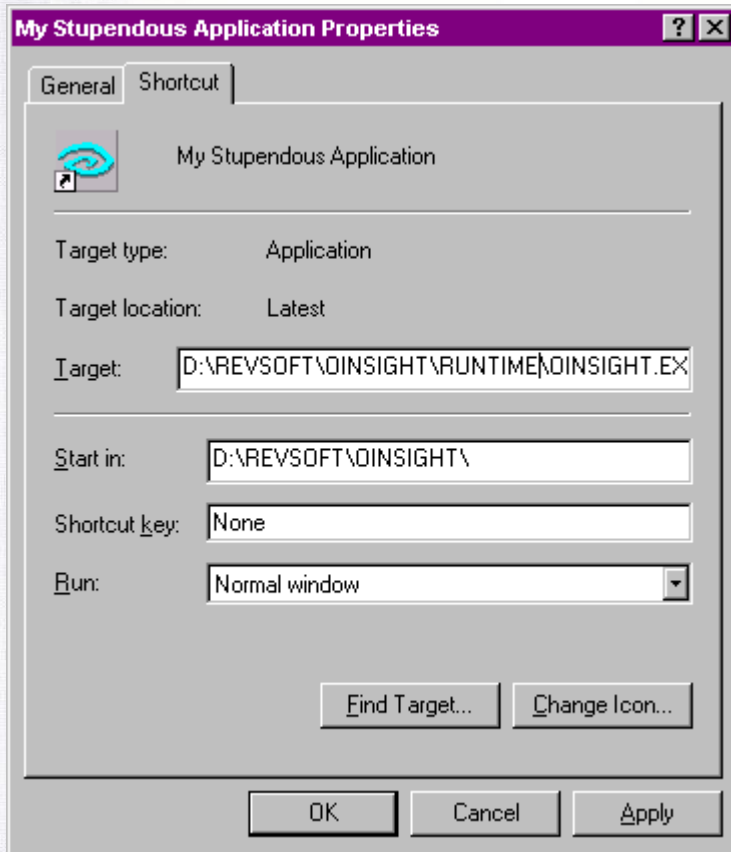- Change the values in selection criteria in a pre-defined select statement

Note that this not a legal opinion and that if you want a definitive answer the only official source is Revelation Software. In addition it does not claim to be a comprehensive list, merely to highlight a number of areas which we believe will be perceived as grey by the majority of developers.

The good thing is, RevSoft recognise that these restrictions are strict and they have provided a way whereby you can just install a single development copy on a multi-user runtime system for occasional development related logins. In other words it is possible to have a development licence (or bumped development licences) running against the system that is networked for X runtime users. We demonstrated this recently at a road show and users were blown away. The choice is NOT just runtime and no file maintenance or full development server deployment packs! In true boutique style it is mix and match. If your users only require occasional developer access, deploy on runtimes and install a single user developer. If development tasks are a part of everyday life for all users, deploy on development ready packs.

MAKING DATABASES HAPPEN

SPREZZATURA

How do we do it? We'll show you how to do it in a 95/98 environment but we're sure you can extrapolate for your own environment.
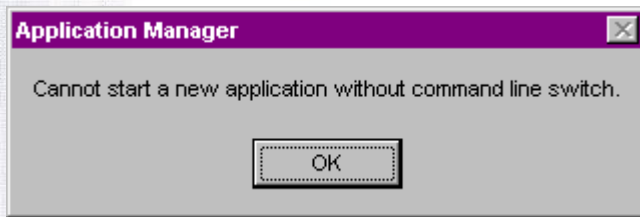

The quick explanation is deploy your application as usual. Next, create two new subdirectories, one called RUNTIME and the other called DEVELOP. Move all DLLs and EXEs from REVBOOT into the DEVELOP directory. Copy the DEVELOP directory into RUNTIME and rename the .RUN files overwriting their EXE equivalents. To run the application, use C:\REVSOFT\OINSIGHT\RUNTIME\OINSIGHT.EXE as the program, with C:\REVSOFT\OINSIGHT as the Start in directory.

To run in development mode, use C:\REVSOFT\OINSIGHT\DEVELOP\OINSIGHT.EXE as the program with C:\REVSOFT\OINSIGHT as the Start in directory. Ensure of course that you are properly licensed in all respects. Speak to RevSoft about this one and if you're buying the development license for deployment like this for occasional use, I'm sure they'll do you a deal! They'll probably even work with you to make deployment as painless as possible

Qtips Revisited– Aaron Kaplan

In this issue, we're going to break with tradition so we can go back to an old tradition, QTIPs, or as our friends in the UK call them, Cotton Buds.

MAKING DATABASES HAPPEN

SPREZZATURA

Application Manager ☒

Cannot start a new application without command line switch.

OK

How many times have you seen this message?

This seems to imply that you can start a new application someway, somehow. And, as it turns out, you can.

If you load up the application properties dialog, you'll notice that one of the options you can change is the database. If two different applications share the same database, then they can both run at the same time. The only restriction is that both applications must be in run mode, not development mode.

To load up OpenInsight in this manner, you must use application switches. These are

/AP = ApplicationName

/UN = UserName

.PW=PassWord

/DV=DevelopmentMode where DevelopmentMode is 1 for Development, 0 for 'runtime'

Revelation Trivia: MERGE was originally a custom application written for one specific customer. It was generalized and added to the product a little later. The final author, for extremely valid reasons, still wishes to remain anonymous.

Like Advanced Revelation, OpenInsight supports multiple language sets. Unlike Advanced Revelation, there is not a dialog box allowing you to modify the default values. However, you can set these values programmatically. By modifying the environmental variable record, you can set the ENV_LND_DEFAULT$ field (15) to the default language set while the ENV_LND_LOAD$ field (16) determines which fields are loaded into memory. Valid language sets that ship with OpenInsight 3.7 are LND_DEFAULT, LND_ENGLISH_INT, LND_ENGLISH_UK, LND_ENGLISH_US, LND_FRENCH_CN, LND_FRENCH_F, LND_GERMAN_D, LND_INTERNATIONAL.

Revelation Trivia: Advanced Revelation 3.12 was written using OS/2 2.1

The ENV_STARTUP_PROC$ field (32) allows you to have an initial login process, much like the ARev logon process. This is slightly different from having the logon process as part of the create event of the main window.

The procedure seems to take 2 parameters. The first appears to do nothing, but there are indications that seem to indicate a proper return value will shut down the system. The second parameter is the station ID. You can modify this value to alter the current station ID.

MAKING DATABASES HAPPEN

SPREZZATURA

(According to Revelation Software, the first parameter returns 1 when called initially from the engine. It would seem that you could use this to determine if the function was called during initial load or later in the program.)

The ENV_SHUTDOWN_PROC$ field (33) indicates a procedure that is executed when the engine is terminated. However, after a series of attempts, I was unable to get any procedure to execute.

I'm sure everyone know about this one by now, but if you load up the OpenInsight about box, and double click on the OpenInsight logo, a puzzle game pops up. Hitting the Fn function keys while the game is running will give you some important words to live by.

Revelation Trivia: Cameron Purdy learned C++ to write the DSO.

The BMPS and ICONS directories contain dozens of files that have not been loaded into the repository.

The system monitor converts all entries to upper case. However, if you prefix run with an underscore (_run), the command line will be passed as entered.

Revelation Trivia: A bottle of DiGel used to be passed down the line from to successive Advanced Revelation product managers

The undocumented CENTER_WINDOW function will center a window.

Center_Window(WindowName [, Xpos, Ypos, Width, Height])

The PURGE.CACHE() subroutine is still valid in OpenInsight. So is the Calculatex function.

Swap.Cursor allows you to assign one cursor's value to another.

SWAP.CURSOR(SourceCursor, DestCursor, Mode)

Where mode determines to copy (1) or move (2)

Revelation Trivia: Get and Set property was originally going to be implemented in Advanced Revelation 3.0, but was shelved because of a user revolt.

By modifying the CFG_QUICKEVENT record in SYSENV, you can add your own quickevents. (Thanks Aaron - for next month - how? - Ed)

Setting DICT_PROTECT$ on any dictionary field will protect that field from changes, even without requiring an index. It does require a !file, but DICT_MFS handles that for you.

Belay that Indexing order Mister! – Carl Pates

OK, Index builds...hands up anybody who has wanted to do any of the following from within an OI form:

MAKING DATABASES HAPPEN

Stop an index build

See exactly what records the index build routine is processing

See exactly what was being processed when the build crashed (but that never happens right? <g>)

Duplicate the index build progress dialog box found in the Database Manager

Carry on working and entering data in your OI system while building an index (!?)

If you have any hands in the air right now then carry on reading, otherwise hit 'Page Down' to go to something hopefully more interesting…

Still here? OK then, here's the deal….

Back in the distant days of ARev there was a very important little function called ESC.TO.EXIT which returned TRUE$ if the user pressed the Escape key. Very handy. The index building function (called BUILD.INDEX surprisingly enough) checked this as it processed each record within a file and allowed you to abort the process if necessary. It also displayed a progress indicator so you could get an idea of how the system was doing.

BUILD.INDEX is also used in OI, but with a few small changes. One of those changes is the removal of ESC.TO.EXIT, as this code is not used in OI. It does however, have a sort of replacement called TEST_ABORT which returns TRUE$ of FALSE$ in the same way as ESC.TO.EXIT did. So, with each iteration in OI the TEST_ABORT function is called, and if that returns TRUE$ then the index build is stopped (TEST_ABORT actually checks to see if the client (in this case OI) that made the request hasn't told the OpenEngine to stop processing, but you can only really set this if you write your own front-end to OpenEngine).

So how can this be exploited? Simple really - We write our own TEST_ABORT function that performs whatever processing we need and returns TRUE$ to abort if necessary. Just follow the simple recipe below:

Make a copy of the real $TEST_ABORT. We still need to call this in our replacement function so save it with a new name such as $TEST_ABORT_REVSOFT

Create a new function called TEST_ABORT that first checks the real TEST_ABORT that you saved in step 1 above. If this returns FALSE$ then carry on to your code that checks to see if the user has cancelled the request. If they have then return TRUE$, otherwise return FALSE$

A simple example of this is shown below

```
Compile Function Test_Abort( Void )

Declare Function Test_Abort_Revsoft

$Insert Logical

Abort = Test_Abort_Revsoft()
```

SPREZZATURA

```
If Abort Else

Call Yield()

* // Insert your cancel checking code here and set the Abort

* // variable to TRUE$ if the process is to be cancelled eg:

If UserHasCancelled Then

Abort = TRUE$

End

End

Return Abort
```

However, the problem here is that we need to call Yield() so that the user can actually click a cancel button, but this may also trash a system variable that the building process uses so we need to make sure we save these off.

Hence we should modify the code like so:

```
Compile Function Test_Abort( Void )

Declare Function Test_Abort_Revsoft

$Insert Logical

Abort = Test_Abort_Revsoft()

If Abort Else

* // Save off critical system variables

SaveAtDict = @Dict

SaveAtRecord = @Record

SaveAtID = @ID

SaveAtRecCount = @RecCount

SaveAtRnCounter = @Rn.Counter

Call Yield()

* // Insert your cancel checking code here and set the Abort

* // variable to TRUE$ if the process is to be cancelled eg:
```

SPREZZATURA

```
If UserHasCancelled Then

Abort = TRUE$

End

Transfer SaveAtDict To @Dict

Transfer SaveAtRecord To @Record

Transfer SaveAtID To @ID

Transfer SaveAtRecCount To @RecCount

@Rn.Counter = SaveAtRnCounter

End

Return Abort
```

Yield and you shall have mercy!

There is one further modification that we can perform to make our code run smother, and that is in the frequency of the Yield() request. There is probably no need to yield every cycle as this is wasteful, but what we want to do is ensure that we call it frequently enough so that we get a good response.

The Windows API function GetTickCount() is a good way to measure this. Windows keeps a count of how many milliseconds have passed since it was started, and this is updated approximately every 55 milliseconds. If we use GetTickCount() to retrieve this value and only Yield() when it has changed then we are only polling the cancel request approx. 18 times a second, which allows the indexing code to run faster. Also GetTickCount() is one of the fastest API functions around so it shouldn't slow you down too much.

GetTickCount() should already be defined in your copy of OpenInsight. It is exported form USER.EXE, and if you don't have it you should prototype it in DLL_USER in SYSPROCS as

```
LONG PASCAL GetTickCount(VOID)

So the new version of Test_Abort works like this:

Compile Function Test_Abort( Void )

Declare Function Test_Abort_Revsoft, GetTickCount

$Insert Logical

Common /TEST_ABORT/ LastTickCount@

Abort = Test_Abort_Revsoft()

If Abort Else
```

MAKING DATABASES HAPPEN

SPREZZATURA

```
If ( GetTickCount() # LastTickCount@ ) Then

* // Save off critical system variables

SaveAtDict = @Dict

SaveAtRecord = @Record

SaveAtID = @ID

SaveAtRecCount = @RecCount

SaveAtRnCounter = @Rn.Counter

Call Yield()

* // Insert your cancel checking code here and set the Abort

* // variable to TRUE$ if the process is to be cancelled eg:

If UserHasCancelled Then

Abort = TRUE$

End Else

* // Tick count could have changed during Yield() so

* // current value

LastTickCount@ = GetTickCount()

End

Transfer SaveAtDict To @Dict

Transfer SaveAtRecord To @Record

Transfer SaveAtID To @ID

Transfer SaveAtRecCount To @RecCount

@Rn.Counter = SaveAtRnCounter

End

End

Return Abort
```

You may want to bear all this is mind when designed other long running processes too.
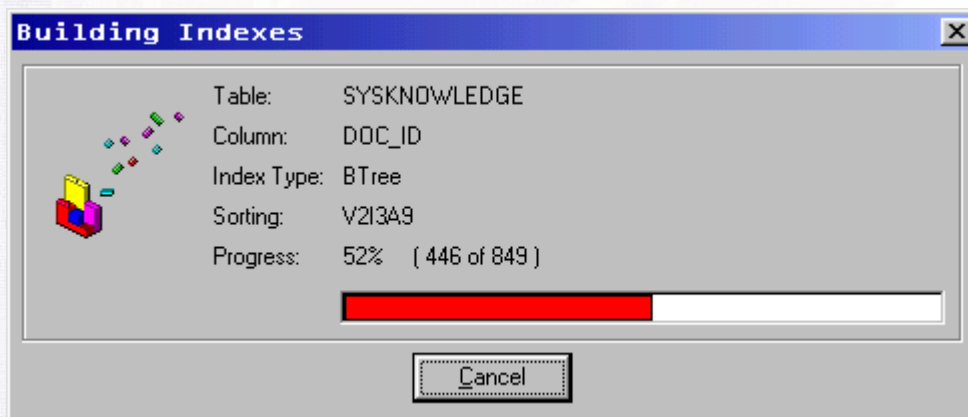
MAKING DATABASES HAPPEN

SPREZZATURA

Where were we again?

So that should give you an insight into how to intercept and abort an index rebuild request. But what about a progress indicator I hear you cry?

Well, If you look at the code above you'll see that the system variables @Rn.Counter and @RecCount are saved off. This is because they are used by BUILD.INDEX and are updated accordingly with every record processed, so if you monitor these variables you should be able to figure out how far along the process you are. Of course, this is not totally straightforward, as index building involves several stages that you need to be aware of, but if I went into that here this would take weeks to write!

Anyway, here's a screenshot of one I did earlier, which shows exactly what records are being processed, along with a progress bar and the facility to cancel the process, and yes, the animation does work just like the one in Defrag <g>!

 Of course, putting in this extra processing does make an index build take a little longer but if you need this extra functionality without having to expose your customers to the wonders of the Database Manager then it's well worthwhile!

One final point, you may have noticed earlier that I mentioned entering data while building an index. Well, if you're insane enough to try it (and I am) the Index Building window you see above will let you do exactly that if you launch it non-modally, because of the Yield() functionality in our replacement TEST_ABORT.

Next stop a 'multi-threaded' OpenEngine!!

ERIC THE EMU'S AREV TIPS

Eric recalls with some fondness the old Timex wristwatch slogan "It took a lickin', but it kept on tickin'!" The PC's finest reliable time interval is the timer tick, which updates low memory at the 0000:046C hex memory address in RAM. The tick also invokes interrupt 8, which many TSR's check periodically.

There comes a time in AREV or REVG when you want to measure time elapsed in increments of less than a second, and AREV offers DOSTIME() as a kind of a micro-timer. REVG has no such native facility.

MAKING DATABASES HAPPEN

SPREZZATURA

At best DOSTIME is messy to use, because it attempts to turn timer ticks into hundredths of a second. Timer ticks occur 18.2 times every second (every 54-55 ms). Such intervals are useful for obtaining raw timings, and also for time delays.

A different implementation of a timer would be to return the ticks since midnight, with an optional implementation of a delay parameter (again in ticks) so that a brief interval can be simply programmed with one line of code. Since the erudite emu has been having a "fine time" with REVG lately, there is a utility to perform this task.

The function FINETIME() has the following syntax:-

```
DECLARE FUNCTION FINETIME

* get the time in ticks, with zero delay

TICKS.SINCE.MIDNIGHT=FINETIME()

DECLARE FUNCTION FINETIME

* delay by 5 ticks = 5/18 of a second

WAIT=FINETIME(5)

A=\0200000000000000000000EB0590000000\

A:=\002EC706130000002EC706110000002E\

A:=\803E0500007415B80013BA01002EFF1E\

A:=\0A002E893613002E890E1100F8065352\

A:=\BA40008EC2BB6C00268B0F268B77025A\

A:=\5B07F82E010E11002E11361300F80653\

A:=\52BA40008EC2BB6C00268B0F268B7702\

A:=\5A5B072E390E1100720AF82E39361300\

A:=\7202EBDAB80014BA00002EFF1E0A00CB\

OPEN '','BP' TO BP.FILE THEN

WRITE A ON BP.FILE,'$FINETIME'

END
```

The emu is intending to fly (!) to Melbourne for the February 25th Revelation Software inaugural drinkathon. See you at the bar!

MAKING DATABASES HAPPEN

SPREZZATURA

Peripheral Trivia

As this issue of S/ENL was put to bed we fed the inner man with:

- TV : Dilbert – the cartoon series

- Book: The Woman and the Ape - Peter Høeg

- CD: War & Peace – Vol 1 – Ice Cube

- WEB http://www.dogpile.com

Join us :	Send Mail to Admin@Sprezzatura.com with subject SUBSCRIBE SENL or complete our online registration form.
Leave Us:	Send Mail to Admin@Sprezzatura.com with subject UNSUBSCRIBE SENL
Change of Address:	Leave at the old address & join at the new one
Web Info:	http://www.sprezzatura.com/
Tell us what you'd like to see in S/ENL:	info@sprezzatura.com

COPYRIGHT NOTICE

TRADEMARK NOTICE

MAKING DATABASES HAPPEN