SENL

July 1997

Sprezzatura's Electronic Newsletter

For Revelation developers by Revelation developers

SPREZZATURA

Contents

MAKING DATABASES HAPPEN

SPREZZATURA

Welcome

Well it looks like the curse of Sprezz has struck again - this time at the sweatshirts. 2 months on and still no sight... bummer - sorry Eric and others! We'll get them to you as soon as possible.

The Web Site (remember that) is starting to draw closer, we've actually got somebody working on it in-house now and hope to see something real soon now. However subscribers are advised not to hold their breath!

Business continues to boom, and the RevSoft special on upgrading AREV users to OI is proving popular. We've been so busy as a company that we've had no chance to catch up on marketing activities so if you're missing written contact from us (our UK and European friends!) then our apologies!

I'd apologise for the shortness of the ramblings as well this time - But again we're busy busy busy!

Andrew P McAuley
Publisher

Ramblings

The Call For Votes has appeared and is progressing well. Many thanks to those of you who have voted already. For those of you who've been holidaying on Mars, this is the process by which a new Usenet Newsgroup is created, and we're trying to get one established called comp.databases.revelation. The great advantage of a newsgroup is that while it behaves like a Compuserve forum it is available to both Compuserve and Internet members. It is also completely free of any commercial sponsor and is unmoderated so a freer exchange of information should be possible. Naturally we hope that the group will be self policing though!

If you haven't seen the CFV then go to comp.databases or comp.databases.pick and download it and follow the instructions contained therein. Essentially you have to modify the message and mail back a YES vote to the moderator. If you can't find the CFV then you can email the Votetaker: Rebecca McQuitty <mcq@wco.com> and ask for a copy of the Revelation CFV. Note that she is taking votes for several groups so please do specify Revelation! And be nice to her, she's doing a wonderful job! <g>

If you haven't done so already, you might like to drop by comp.databases.pick in any case and review the mail there about the CFV. Essentially some Pickies are calling for a NO vote because it would detract mail from Comp.Databases.Pick. Crazies get everywhere I guess. Add your voice to the reasoned arguments of such diverse (and much appreciated people) as Jim Vaughan, Jane Gregory (brilliant posting Jane - thank you!), Dave Sigafoos, Donna Naus, Douwe Rienstra, D Craig and Jim Douglas who've all posted against the ramblings of the pick people! I must stress incidentally that I have NOTHING against Pick people, I've worked with Pick people before and will do so again, just against the attempt to block the newsgroup!

Revelation Software in the UK are tooling up for their all-dayer which is planned for September 30th at the QE2 conference centre in central London. Cameron Purdy will be a guest speaker and best of all

MAKING DATABASES HAPPEN

SPREZZATURA

it's free! It'd be nice to see some more of our foreign friends there this year - why not plan a holiday in London to coincide with the conference and perhaps we can all have a party!

OpenInsight 3.4 has just been released so we're looking forward to playing with that, I'm sure that the Quarter 2 disk will arrive real soon now... (fatal mistake labelling ANYTHING by quarters!). The 3.4 CD marks a good departure from usual style - there is now a 3rd party developers section on the CD where people can ask to have their product included for developer evaluation. Phoenix Solutions get there first with their TCL for OpenInsight - way to go RevSoft Fellas! (well Jennifer actually but I don't know the female equivalent of fellas).

If RevSoft are looking for items for my Xmas list (apart from wild parties) a Connection Object for Linear Hash would REALLY make my day.

Imagine being able to develop an application on Linear Hash using DataSets and then just plug in a different back end like say SQL Server or Oracle with little or no coding changes. I must be overlooking the complexity of the issue 'cos this is such a no-brainer!

We're working on some docs on datasets which will appear as a Tech Bulletin in a month or so - watch this space!

FEEDBACK

Chris Cordera writes

"Andrew Great newsletter! I posted the news release spoof internally and shook up a few people...was tempted NOT to tell them it was a joke...it was, wasn't it??? <g>"

I must say I thought it was seriously funny - shame the author (like Eric the Emu) has to remain anonymous - but you know who you are so good work fella!

Apologies to the people who responded about upper/lower case programming, but the way I saved the messages deleted the Email addresses so I can't credit you individually, but here goes a few comments from the floor on Programming and Newsgroup setup! (actually probably just as well for the first posting - I mean, calling your boss cheap! <g>)

+++

"RE, UPPER/lower programming, I vote for UPPER. I think the brain works by pattern recognition and mine has 20 years of uppercase conditioning <g>.

RE, Internet newsgroup. My only access to the net at the present time is through CIS (boss is real cheap). With my off-line reader, CIS is great and net access would be harder and slower... The few times I've accessed the Revsoft website, it is painfully slow to read the messages online. I realise CIS is expensive for some and the net is cheaper for them. I fear the newsgroup would fragment an already small group even more, given we already have this forum and Revsoft's website. However, IF this is progress, don't let me stand in the way <g>"

+++

MAKING DATABASES HAPPEN

SPREZZATURA

">> I think the brain works by pattern recognition and mine has 20 years of uppercase conditioning. <<

THEN WHY DON'T YOU WRITE ALL YOUR MESSAGES ON HERE IN UPPERCASE <G>,

or do you run them all through a filter prior to reading them?

>> RE, Internet newsgroup. My only access to the net at the present time is through CIS (boss is real cheap). With my off-line reader, CIS is great and net access would be harder and slower <<

Agent (or Free Agent) makes accessing Internet newsgroups as painless as CIS. I think we all hope we can attract a larger crowd on the Net, the group on here seems to shrink by the day. The WWW site message area is painful to use, in fact so painful I never use it (and I access the net with an ISDN line). When we, or I should say Andrew, get the Newsgroup up and running I would suggest that Revelation Software kill the WWW site discussion area and replace it with a pointer to the newsgroup.

BTW I find CIS to be expensive, Internet access via an ISP is cheap."

+++

"Hi There, Just read your latest SENL. Whatever you're drinking... Make mine a double!!!!"

+++

"Jokes you can see coming a mile off department..."

So this guy goes into a bar and says "Make mine a large one" and the bartender says "Sorry sir, this is a bar not a plastic surgery"

+++

And finally an attributable one from Pedro Azpurua

"I was very pleased from the senl6. But we at our firm are very concerned with the changes in the industry (client/server, SQL, and Intranets). It looks like the editors are pleased (or are in a profitable niche) with the state of OpenInsight (16 bits) and linear hash format. We wonder why they don't try to show some more insights on the future of rti in the java environment and the web. I am sure they will try to always have a comment on this. Thanks"

Well fair point Pedro, but without wishing to state the obvious, we are developers not independent consultants - even if we thought Alpha 4 was a better product than OpenInsight we wouldn't say so whilst our customer base was using OpenInsight. As it happens it is our belief that OpenInsight is currently at the best stage it has been in in its whole life. It has problems but Office 97 is hardly bug free and if Microsoft can't deliver with all that resource what do you expect from RevSoft?

As to the 16 bit element - severe pain! Especially given that RevSoft has a working 32bit engine years ago but development was stopped. The Java moves are encouraging as it shows that at least they are

MAKING DATABASES HAPPEN

SPREZZATURA

keeping up with the industry, and at the moment the web awareness of the product makes it an industry leader.

Our current Web involvement calls for Data Pumping to a SQL Server engine and it seems to work just fine so far! That's the great thing about datasets though, so easy to work with.

IT'S THAT MAN AGAIN - AARON P KAPLAN

Notes from a bloke

After spending years in Technical Support for Revelation, you get used to certain thing when meeting ARev and OI developers for the first time.

Generally, one is asked a whole bunch of questions. One of the questions I'm most frequently asked is, "Why don't you eat pork?".

After swallowing the mouthful of king prawns, I'll generally embark on a long dissertation about Leviticus, dietary halacha (basically the laws of kashrut) and other assorted chazehri. However, you're probably more interested in the technical questions I'm asked.

One of the more popular questions concerns undocumented secrets, tips & tricks and the sort of things one would expect to find out only by working at Revelation. Well, as much as I've love to tell you all sorts of things, I can't being bound by Section 42 subsection 5, paragraph 4a, part 2. Actually, I made it up There is no such law, or if there is the odds of it covering proprietary information are so slim it's statistically impossible. However, that's the first trick you need to learn. When we don't know something, we make it up.

Well, actually, that's not really true either. The correct term would be educated guess.

All kidding aside, the main point is, how can you learn about all these things you feel Revelation's been holding back, and what Revelation is holding back that you don't know about. The short answer is, not much. There are very few proprietary items in the code. All the actual code is proprietary. The encryption logic is proprietary, the LH code is proprietary, user logic is proprietary.

That's about it. Each falls under different reasons for being proprietary and we'll explain them in short turn. After that, we'll get into ways you can legally find out about internals and other things that are under or not documented.

The first type of proprietary is the security code. This is the easiest to explain, which is why I'm starting with that. Maintaining system security is an important part of the system. Allowing anyone to decrypt passwords creates a system where there is no need for a password. Anyone that works on any type of secure system knows that password and security precautions are a discipline all it's own under computer science. Just as no one expects Novell to explain how they secure passwords, Revelation should not be expected to explain how they do that as well.

Modifying @USERNAME and @ACCOUNT fall into this same category. If a user can change @USERNAME or @ACCOUNT on the fly, then system integrity has been violated. Since it's possible to

MAKING DATABASES HAPPEN

SPREZZATURA

use the LOGTO and OPENAPPLICATION features to move from application to application, that method should be used since it enforces system security.

The second style of proprietary code is 'that which makes the product unique' type. The LH hash algorithms fall under this category. Yes, all Pick derivatives are essentially the same, however, it's LH that makes Revelations implementation special. It's that speciality that makes it proprietary.

The third type is the intellectual section and that's why all source code is covered in it. Generally speaking, most people agree with the first type and slightly fewer with the second type. It's this type which causes the most contention. The idea's been explained over and over again, but this time I'm going to try a different tactic.

Revelation, as the author of a piece of software, has every right to withhold source to whatever it writes. Many arguments follow about the purchaser has the right to modify system functionality, to work around problems or limitations that effect his app, to add new functionality to a procedure, and so on. What I find humorous is that many of the same people who make these arguments secure their own systems tighter than a drum. They don't ship source code, some limit PAINT and COMPILE abilities in full development copies, encrypt windows and do all sorts of things to maintain their intellectual property. Yet, somehow, they refuse to allow Revelation that same privilege.

I know of a few sites (that have since been busted) that shipped unlimited development ARev executables to their clients. They then implemented their own user count logic to prevent users from logging in unless they order additional users from themselves. Again we see that double standard of doing one thing, but expecting Revelation to follow a different set of rules. In this case, it was immoral for Revelation to charge per user but perfectly acceptable for these developers to do so. Incidentally, one of them charged close to $1,000 per user seat. We're talking in the tens of millions of dollars in lost revenues. That would have hired many programmers and paid for plenty of marketing that had to be foregone.

This all brings up a secondary point that if everyone who pirated Revelation products on the grounds that they were going out of business soon actually paid for the stuff they stole, Revelation would probably be a much larger company at this time.

So, the next time you complain to Revelation that you need the source for LHash or BTREE.EXTRACT or WINDOW or REVCAPI.DLL, ask yourself this: Would you ask Microsoft, Borland, Lotus or Novell for the same sort of information? And if you would, do you think they would actually do it?

Just remember, Revelation is not a not-for-profit institution, and they have the right to make money on the software they've written and created.

So, soap box put aside, we still haven't answered the question on how you can find out things legally that go on inside the product. Most of you know about most of the secrets now. Many of them have already been documented through Revmedia, RevTech, CompuServe, Revelation's

Technical Bulletins and various other sources. With that in mind, we'll concentrate on OpenInsight.

OpenInsight provides two tools to help you see a good chunk of what the system is doing at any given point. These tools are the System Monitor and OpenEngine. We'll walk through how you can use these tools to find out things you never knew before, starting with OpenEngine.

MAKING DATABASES HAPPEN

SPREZZATURA

If you task over to OpenEngine, you, the first thing you'll probably notice is that it's a thin frame window. The second thing you'll probably notice is a large edit box in the centre, though some of you will probably notice that the engine icon is a mobius strip and wonder where the whole thing starts.

If you look through that edit box, you'll see all sorts of information. Here's what mine looks like when OpenInsight first starts up, starting from the bottom

```
RUN GET_APP_INFO "MYAPP"

RUN BCHECKLEGALUSER 'MYAPP','MYNAME','GRPSDFKL',0

RUN SWAP_DATABASE 'MYAPP'

RUN SETAPPID 'MYAPP'

RUN GET_SYSINFO

RUN GET_REPOS_TYPES ''

RUN GET_REPOS_TYPES 'MYAPP'

RUN GET_ALL_TYPES '',1

( WG ) RUN GET_REPOS_STORAGE_NAMES ''

( WG ) RUN GET_REPOS_TYPE_FAMILIES ''

( WG ) RUN OINOTES_GET_REPOS_TYPES 'LOTUSNOTES'

( WG ) RUN OINOTES_GET_REPOS_TYPES 'OI'

( WG ) RUN OINOTES_GET_REPOS_TYPES 'GENERAL'

RUN GET_APP_INFO 'MYAPP'
```

The lines prefixed with (WG) only show up in OpenInsight for Workgroups. You might be able to tell by this that I'm moving from OpenInsight version to OpenInsight version so you can see how it's all identical.

What are all these programs doing? Mostly, they're setting up the environment space so OpenInsight can execute property. It's attaching database files, setting up the repository outliner, checking user name and password, validating rights and setting up access permits.

Most of these programs are designed to run from SYSPROG only or on boot-up so I don't recommend trying them, but if you're like most people, the minute we tell you not to do something, you'll rush right out and do it. With that in mind, please, I beg you, don't send me any money, gifts, Van Goghs or anything else of value. (instead send them to me - Ed)

MAKING DATABASES HAPPEN

SPREZZATURA

For the rest of this section, I'm going to use OpenInsight like I normally do, and report back what the engine's doing. First thing I always do is open up the System Monitor, so that's what I'm about to do....hold on while I do just that..........................

OK. I'm back. The system monitor is now open, so, let's task over to the engine and see what it has to say for itself.

It says RUN SetDebugger 'SPY',1 while the cow says 'MOOOOOOO'

The curious amongst you are probably asking yourself what that means. Technically it's an onomatopoeia, which means a word formed from sounds associated with the object named. It comes from Latin, and is literally translated as word making.

The technical amongst you are wondering what SetDebugger 'SPY', 1 does. Simply put, it enables the display off all that neato information you see in the system monitor as you use OpenInsight. If you're like me and keep the system monitor open all the time, you might find that you don't want all that information to scroll through or you might find it annoying that you end up with all sorts of information in your window status bars. I find it annoying anyway, though not as annoying as when those little gnat like things fly all about you on a hot, sticky day and you end up breathing one or two in.

That's really annoying.

Anyway, logically we can make the deduction that the 1 is enabling something, so passing in a 0 as the second parameter should disable something. Let's see that happens if we enter RUN SetDebugger 'SPY',

0. Nothing, really from the looks of it.

RUN SETDEBUGGER 'SPY',0

SPY, 0

That's all I see in the system monitor. However, as a former president once proved, looks can be deceiving, and as my former wife proved, looks can be deceitful as well. However, something has happened. If you proceed to run a window (EXEC OR_MAIN since I know it's in all versions), you'll find that you don't get any more information in the system monitor. Personally, I prefer it like this and almost always execute this line immediately after opening the system monitor.

Next, I'm going to run the Form Designer. OpenInsight for Workgroups gives me

RUN FORMDES_SUB 'GETEVENTINFO'

while OpenInsight for LAN's returns back

RUN GET_SYSINFO

RUN GET_TABLE_TYPES

MAKING DATABASES HAPPEN

SPREZZATURA

I'm not too sure exactly what these programs do, but OpenInsight does have an advantage over ARev in that program names and parameters (when not numerals) tend to be very obvious. I'd guess that FORMDES_SUB is loaded up all the QuickEvent stuff (which is stored in SYSENV as

CFG_EVENTS and CFG_QUICKEVENTS), GET_SYSINFO looks to return user name, application name, @STATION and a few other thing and GET_TABLE_TYPES returns the listing of all the BFS's stored in the system. Entering the commands into the System Monitor confirmed my suspicion. With a bit of trial and error and guess work, plus a call to John Duquette or Cameron to ask, I could probably figure out what they all mean.

Doing a File Open calls GET_REPOS_ENTITIES which is documented. Actually reading in the window causes more entries in the log than I care to type. However, some of the more important ones have to deal with creating the temporary dictionary, locking the window entity, determining access permissions, and all sorts of things.

Right now I want you to close the window and task back to the engine. I'll sit here and wait...just let me know when you're ready....

Bim bom bim biddee bom bim biddee bim bom bim bom...

Tral la la boom dee yay...there is no school today...the teacher's gone away, we'll have the day to play....

Don't mind me, I'm just singing to myself...

the 5 o'clocks whistles on the blink, the whistle won't blow so what'd'yah think, my mother's still in the factory because she don't know what time it happens to be.....

Done? OK good...If you look at the top of the engine window, you'll find a row of buttons...button button, who's got the button...anyway...where was I...oh, yes, the buttons. One of the buttons is labelled Start Log. Click it. Watch as it magically changes to Stop Log and how View Log is disabled. We'll ignore this for a while and let's head back to the form designer.

Go ahead and open the same window you opened last time... I'll wait..

Vinteh gayt ah reygin vinteh gayt ah shnay..oy vey vey mine cup tichk

niray..nacha tunim nacha tumin...

Sorry, just singing to myself again while the form is loading....

So, know that's the form's all loaded, head back to OpenEngine. Click the Stop Log and watch it change back to Start Log and watch View Log enable itself again. Click on View Log.

If you look through, you'll see that all those programs that just had #1 and #2 have little references to what they mean. For example, on my window, I receive

RUN REPOSITORY 'ACCESS', #1

MAKING DATABASES HAPPEN

SPREZZATURA

#1: 'SYSPROG*OIWIN**EDITOR'

On a save of that window, I'll find the log reporting

RUN REPOSITORY 'WRITE', #1, #2, #3, #4, #5, #6, #7, #8, #9, #10, #11

#1: 'SYSPROG*OIWIN**EDITOR'

#2: '0'

#3: ''

#4: ''

#5: ''

#6: ''

#7: ''

#8: ''

#9: ''

#10: 'EDITOR'

#11: (Not posted because of the 8 bit chars, but it should be obvious this is the actual record stored in SYSREPOSWINS)

By using the engine log, you can find all sorts of information about repository calls just by reading, saving and compiling forms, stored procedures, inserts and events.

Through all this, it will show you how you can edit an event handler in the System Editor and compile it without having to go through the troubles of loading up a large window to make a small little change.

If you capture a log while you're actually using a window, you'll find all sorts of information on how to call RUN_EVENT and all sorts of things. If you capture a log while moving around the Database Manager, you'll find all sorts of interesting programs. Even more fun (I have a strange idea of fun) is loading up the Table Browser.

Keep in mind that some of these programs are actually ROUTINES, and not SUBROUTINES or FUNCTIONS so they might not actually return anything. You might have to set a system receiver to view the information.

Having move quickly through the Engine Log, it's time to head back to the System Monitor.

SPREZZATURA

First thing we want to do is turn back on the spy functionality. If you've been paying attention, you'll know how to do this. If you haven't been paying attention, I understand attention is getting tired of your delinquencies and will make sure you are sent to debtor's prison for eternity, or an index rebuild, which ever comes first.

Now that the SPY is re-enabled, let's run through a few things. (No, let's wait for the next issue - Ed)

CELEBRITY INTERVIEW - Tony Splaver

>> Who are you?

Tony Splaver, married, male, 26 years old.

>> Who do you work for?

Intromedix, Inc.

>> What is your role in the RevSoft community?

Created the OIPI.

>> How long have you been involved with RevSoft?

Almost 8 years.

>> Which products do you use?

OI for testing the OIPI only. I use VB, VC++ and SQL for the last 3 years.

>> What do you most like about RevSoft products?

Multivalued and variable length. Basic+ is a good language too.

>> What do you most dislike about RevSoft products?

No ODBC or 3rd party control (ActiveX controls).

>> If you HAD to use another database what would it be and why?

MS SQL Server 6.5 -- scaleable, good for multi-user, fast, industry standard, nice for n-tier C/S.

>> What are your favourite 3 books and why?

Asimov "Foundation" series

Crichton "The Andromeda Strain"

Clancy "Without Remorse"

MAKING DATABASES HAPPEN

SPREZZATURA

Because they are good science fiction and have good story-lines.

>> What are your favourite 3 CDs/Albums and why?

I only listen to classical CD while driving in traffic every morning for an hour, and I never take out my other CDs.

>> What are your three favourite films and why?

"Star Wars" - good story and good effects.

"Dirty Rotten Scoundrels" - funny.

"Men In Black" - funny.

>> What event in history would you most like to have been present at?

Landing on the moon.

>> Who is the best president/prime minister we never had and why?

Not sure.

>> Your motto/witty aphorism

Get it done, and get it right.

web site: http://www.pacifier.com/~splaver

ERIC THE EMU's AREV TIPS

My last excursion into the realms of account passwords etc. yielded a spooky fact - use the password "ABC" and the encrypted form is (do I hear psycho music???) "iqOk".

In Australia, Sausage Software is a company famous for its HotDog Web Page authoring software. And for having a recent record drop in share prices. The founder has coined the term "snaglet" to represent any piece of single-function software which is widely used on a daily basis by many people. He hopes to market such software, generated by an army of ex-university chimps sitting at keyboards all day long. As Einstein once said "Anyone can have a good idea (once). But you only make it to genius class if you can do it twice." (Eric Einstein - 1927)

Sausage has snaglets, but Eric has omelettes (schoolboy French for little female emus) I figured that the AREV community may have some omelettes of its own to share. Here are some of Eric's - for this issue's column I'll give you my top five. Perhaps others will be tempted to contribute an omelette in future editions. They must be compact code (one program and under 40 lines preferred - the smaller, the better) and useful across most / all versions of AREV. Email to feedback@sprezzatura.com - No bitching about upper or lower case, either - Eric uses Ctrl-Z to reformat all his code correctly.

MAKING DATABASES HAPPEN

SPREZZATURA

Eric's top Five Omelettes

(1) RDIR.COM

DOS-based display of REVMEDIA. If only it sorted the output by AREV filename / dosfile prefix...download the version 4 rewrite from the Simple Simon AREV Utility page at http://ourworld.compuserve.com/homepages/state_of_the_art/util8.zip

(2) DATE -

TCL utility by Eric the Emu - performs an iconv or oconv at TCL to show a date in internal/external format. Usage DATE 10678 or DATE - 3444 or DATE 07/01/74 Useful when you are verifying dates or keying in dates from the editor / at TCL

```
EXPENDABLE SUBROUTINE DATE
  * called from tcl, will convert into or out of date format the command
  * line parameter
  * will convert into or out of AREV's internal date format automatically
  * will accept multiple parameters e.g. DATE 8949 8765
  SENT = @SENTENCE
  SWAP ' ' WITH @FM IN SENT
  FIELDS = COUNT(SENT,@FM) + (SENT <> '')
  LOCATE 'DATE' IN SENT USING @FM SETTING POS THEN
    FOR I = (POS+1) TO FIELDS
      IF SENT<I> <> '' THEN
        IF NUM(SENT<I>) THEN
          IF (SENT<I> < 30000) THEN
            * adjust for your local date format
            CALL MSG(OCONV(SENT<I>,'D4/E'),'','','')
          END ELSE
            * adjust for your local date format
            CALL MSG(ICONV(SENT<I>,'DE'),'','','')
          END
        END ELSE
          *adjust for your local date format
          CALL MSG(ICONV(SENT<I>,'DE'),'','','')
        END
      END
    NEXT I
  END
RETURN
```

(3) TEDIT - checks the VOC pointer and opens an edit session in the correct library. Useful where applications have multiple libraries, especially in multi-module applications. Also useful in unfamiliar applications which break to debugger in a specific program. By Eric the Emu. Not fancy but a time saver....

```
Usage TEDIT MYPROGRAM
EXPENDABLE SUBROUTINE TEDIT
  SENT = @SENTENCE
  SWAP SPACE(1) WITH @FM IN SENT
  FMCOUNT = COUNT(SENT,@FM) + (SENT <> '')
  RECORDKEY = ''
  FOR I = 2 TO FMCOUNT
```

MAKING DATABASES HAPPEN

SPREZZATURA

```
    IF SENT<I> <> '' THEN
      RECORDKEY = SENT<I>
    END
  NEXT I
  FILENAME = ''
  OPEN 'VOC' TO V.FILE THEN
    READ V.REC FROM V.FILE, RECORDKEY THEN
      FILENAME = V.REC<3>
      PERFORM "EDIT ":FILENAME:" ":RECORDKEY
    END ELSE
      MTXT = 'Cannot read '
      MTXT<2> = RECORDKEY
      MTXT<3> = ' from VOC - routine not catalogued.'
      CALL MSG(MTXT,'','','')
    END
  END ELSE
    CALL MSG('Cannot open VOC file','','','')
  END
```

(4) DICT.VOC symbolic item COMPILEDATE.

Useful for version checking, or just putting the names of the programs you worked on into a time sheet or invoice. Nature's own time & attendance system <g>

Usage: LIST LIBRARYFILE BY-DSND COMPILEDATE COMPILEDATE

Symbolic is length 45 left justified Single value with the following spaghetti-like formula (no excuses here - this code is awful):

```
  ANS2 = ''
  IF @ID[1,1] = '$' AND @RECORD[1,1] <> CHAR(2) THEN
    FIELDS = COUNT(@RECORD,@FM) + (@RECORD<> '')
    ANS= @RECORD<FIELDS>
    ANS3=ANS
    SWAP ' ' WITH @FM IN ANS
    I = COUNT(ANS,@FM) + (ANS <> '')
    ANS2 = ICONV(ANS<I-2>:'/':ANS<I-1>:'/':ANS<I>,
    ----->'D2/E'):',':ICONV(ANS<1>,'MTH'):' ':ANS3
    IF ANS2[1,1] = 9 THEN ANS2 = '0':ANS2
  END
RETURN ANS2
```

(5) DICT.VOC ITEMS F6-F20

AREV comes pre-loaded with DICT.VOC items F1,F2,F3,F4,F5, representing the first five fields of any file. I always extend this all the way to F20, and convert all these items to Multivalue. When you want to dump a file, but you can't be bothered consulting the dictionary first, you can easily and quickly LIST F1 F2 F3 F4 F5 from TCL and get a reasonably intelligent output, working totals and no mind-numbing LISTDICT memory tests !!!

Sure as hell ain't U.S. omelets...(as per the Word 6.0 spell checker)

Eric the Emu

MAKING DATABASES HAPPEN

SPREZZATURA

THE X EVENTS - ANDREW P MCAULEY

On a recent training course one of the subjects I was explicitly asked to deal with was promoted events. This got me to thinking, because they are such a fundamental part of the product and yet their existence is all but denied by the documentation - the reason being of course that there are no supporting tools to assist with them. So I though that perhaps a little tutorial was in order for those of you who haven't yet had a chance to play with these undoubtedly powerful techniques!

So firstly, what is a promoted event? Well, if you like, you can think of it like an MFS for event handling, allowing you to intercept the handling of events at a more global level. Of course, if you don't understand what an MFS is then I've probably just served to confuse you even more - sorry! Let's try again with a more simple example - imagine you want to display a line of help on the statusline for every control that can get focus, when that control gets focus. Of course you COULD write an event handler for every control on the window, but then you could hand forge your own car parts...

What would be better would be to write a routine once and have it called in all GotFocus scenarios in our application. And unsurprisingly after such a blatant lead in, that is exactly what promoted events allow us to do.

The mechanism is actually stunningly simple and relies upon naming conventions. The system event handler is programmed to look for, and run, events in a particular order. Thus if we have tabbed off a particular control, it will first look for a lostfocus event defined for that control, then a more generic event and another until it finds a catch-all event for all system events in windows. To paraphrase, the system will say

Look for an event handler called A*A*A

If you find it run it and it will set a continue yes or no flag

If you don't find it OR the continue flag is set to yes

Look for an event handler called A*A*B

If you find it run it and it will set a continue yes or no flag

If you don't find it OR the continue flag is set to yes

Look for an event handler called A*B*A

If you find it run it and it will set a continue yes or no flag

If you don't find it OR the continue flag is set to yes

And so on using rules for naming that are laid down by the system. As an example, here's a System Monitor trace for a GotFocus on an EditField called SUBJECT in form EXAMPLE

RUN_EVENT calling:EXAMPLES*GOTFOCUS*EXAMPLE.SUBJECT ...

MAKING DATABASES HAPPEN

Looking for:EXAMPLES*GOTFOCUS*EXAMPLE ...

Looking for:EXAMPLES*GOTFOCUS.EDITFIELD.OIWIN* ...

Looking for:SYSPROG*GOTFOCUS.EDITFIELD.OIWIN* ...

Looking for:EXAMPLES*GOTFOCUS..OIWIN* ...

Looking for:SYSPROG*GOTFOCUS..OIWIN* ...

RUN_EVENT calling:SYSPROG*GOTFOCUS..OIWIN* ...

RUN_EVENT calling:SYSPROG*..OIWIN* ...

What does this actually tell us? Well let's look at it line by line

RUN_EVENT calling:EXAMPLES*GOTFOCUS*EXAMPLE.SUBJECT ...

There is an event handler defined for the GotFocus event for the control called Subject in the Window Example, so the event handler has run it.

Looking for:EXAMPLES*GOTFOCUS*EXAMPLE ...

Look for an event handler that has been defined to run for the GotFocus event for EVERY control in the Example window.

Looking for:EXAMPLES*GOTFOCUS.EDITFIELD.OIWIN* ...

As it says "Looking" not "RUN_EVENT calling" it was unable to find the previous promoted event (presumably because nobody has created it) and so it is now looking for an event which has been created to deal with the GotFocus event on all editfields on any window in the Examples application. Note that because we have now moved outside of the scope of a specific window, the event handler is qualified to all Windows WITHIN the Examples application.

Looking for:SYSPROG*GOTFOCUS.EDITFIELD.OIWIN* ...

Like with previous it is now looking for a promoted event - this time the same as before but for ALL editfields in any window in any application. That it because the first part of the event key is SYSPROG which all applications inherit from.

To muddy the waters a little here, if EXAMPLE had inherited from UTILITY and UTILITY had inherited from SYSPROG the event handler would firstly look for the EXAMPLE handler, then the UTILITY handler before settling on looking for the SYSPROG handler.

Looking for:EXAMPLES*GOTFOCUS..OIWIN* ...

Still fruitless in its searching, the event handler is now looking for a handler that has been defined to handle the Gotfocus event for ANY control in any window in application EXAMPLES.

SPREZZATURA

Looking for:SYSPROG*GOTFOCUS..OIWIN* ...

Tiring of its search then event handler tries to find a handler defined to handle the GotFocus event for ANY control in any window in any application...

RUN_EVENT calling:SYSPROG*GOTFOCUS..OIWIN* ...

and is successful!

RUN_EVENT calling:SYSPROG*..OIWIN* ...

Finally the last resort - find and run the generic handler for any event in any control in any window in any application.

It is at this final level that the quick event processing is actually done.

So nothing to it hey? But do you like the way in which I casually threw in the "System Monitor Trace"? Well, refer back to Aaron's "Notes from a bloke" and read all about it! And stop skipping ahead - read sequentially or you might miss an important clue! <g> (or not as the case may be).

The forcing of an event handler to go through the entire chain was achieved incidentally by simply returning 1 from the event handler for the GotFocus for the control.

This adds another bit of confusion into the mix - what is it with this returning 1 or 0 stuff? I mean, we're told that returning 1 carries on down the event chain, but returning 0 kills the event chain stone dead! But whenever I look at an event in the form designer it has Return 0 in so how do ANY promoted events ever get run?

The answer is fortunately simple. The Return 0 isn't actually there. You're imaging it. Well, with a little help from the event editor. You see, when you invoke the event editor it assumes that you'll want to Return something, so it suggests 0. Now if you abandon the update without saving anything you'll not actually save away a Return 0 as it wasn't there in the first place.

Ever wondered why forms take so long to save? Well actually what's happening is that as you save, the system is looking for promoted events in the chains from the controls you're using, and if it finds them it includes them in the compiled form definition. This saves it having to resolve the event chain at runtime which would slow it down quite a bit! Of course, a downside of this is that if new promoted events are added, they aren't included in this chain so windows that have already been compiled won't use these new events unless they are recompiled.

To get around this problem of events being resolved at runtime when the Return 1 command is issued (which, remember, effectively tells the system that it has not choice but to resolve the event chain) the system employs some fairly complex buffering so that once it has fully resolved an event chain for a control, it remembers that information and does not have to resolve it again. Thus the same System Monitor trace as before would read differently on the second pass through the gotfocus. Now the system would "remember" that it was unable to find certain elements and this time through it wouldn't look for them. The event log would thus look as follows

RUN_EVENT calling:EXAMPLES*GOTFOCUS*EXAMPLE.SUBJECT ...

MAKING DATABASES HAPPEN

SPREZZATURA

RUN_EVENT calling:SYSPROG*GOTFOCUS..OIWIN* ...

RUN_EVENT calling:SYSPROG*..OIWIN* ...

all elements preresolved!

Of course this means that if you do change a promoted event you will frequently have to log out to make the new promoted event logic be visible to the system as the old version will be buffered. I would suspect that the act of compiling code flushes any old versions from memory buffers, but as promoted events are infrequently compiled, this does not apply.

So let's cut to the crunch - how do we promote events ourselves? Well, we simply give them straight A's at their next staff appraisal and recommend them to the review board... oh, no, sorry, still writing the Sprezzatura Office Manual...(help me - that man Kaplan's insanity is spreading through the office like a virulent case of chicken soup).

As previously noted, tools are thin on the ground (can't you just see all those super code johnnies out their sharpening their coding tools at this blatant gauntlet slap - perhaps Phoenix will have something for us by the next issue! <g>) so it has to be a manual operation. So imagine that you want to develop a GotFocus event handler that applies to every control that can get focus in your Example window. Firstly develop the code - I use a window for this which contains all of my promoted events so that I know where to look, but for the sake of simplicity assume that we wish to use the GotFocus for the Subject prompt on the Example window. All that theoretically needs to be done is to copy the object code for the non-promoted event to have the name of the promoted event we need, in SysReposEventExes, the system will do the rest.

So log into the System Editor and at the Exec Line type

RUN COPY_ROW A,B,C,D

where

A = "SYSREPOSEVENTEXES"

B = "EXAMPLES*GOTFOCUS*EXAMPLE.SUBJECT"

C = "SYSREPOSEVENTEXES"

D = "EXAMPLES*GOTFOCUS*EXAMPLE"

simple huh? I'm sure you can figure out which event you need by looking at the System Monitor trace. Do try to avoid promoting stuff to Sysprog level though as you'll confuse your fellow developers!

Well, no article from me would be complete without a sample of code for people to savage so the same site that I was teaching at had a programming department who just LOVED promoting events. Problem was the people I was teaching had no clue what was there ('cos promoted events don't show up in the repository) so they wanted a simple way of finding out just exactly what promoted events were actually out there.

MAKING DATABASES HAPPEN

SPREZZATURA

It didn't take a genius to work out that events in their most common instantiation have a three part key made up as follows

Application*Event*Window.[Control]

where [Control] is optional - it isn't there is the event is for a Window. It also wasn't a massive leap to establish that promoted events don't have a full stop (period for our American readers, although everyone over here knows events don't have periods, periods are events). So our program to sniff out promoted events became very simple and follows!

Well, hopefully that has simplified a very powerful area of the product for those of you who didn't know all this. For those of you who did - my apologies and thanks for your forbearance!

```
Subroutine Promoted_Events(Dummy)
  Open "SYSREPOSEVENTEXES" To vSysReposEventExes Then
    Select vSysReposEventExes
    EoF = 0
    PromotedIds = ""
    Loop
      ReadNext Id Else EoF = 1
    Until EoF Do
      ThirdBit = Field(Id, "*", 3)
      Promoted = 1
      If Index(ThirdBit, ".", 1) Then
        Promoted = 0
      End
      If Promoted Then
        PromotedIds := Id : @Rm
      End
    Repeat
  End
  Call V119("S", "", "A", "L", PromotedIds, Flag)
  Convert @Rm To @Fm In PromotedIds
  PromotedIds[-1,1] = ""
  /*
    Now report on it however you want!
  */

  Debug
Return
```

Peripheral Trivia

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

- TV :Too busy

- Book: Nope, still too busy

- CD: "Time for Healing" - Sounds of Blackness (can do this whilst programming)

- WEB: See TV and Book.

MAKING DATABASES HAPPEN

SPREZZATURA

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