

SPREZZATURA

45 St Mary's Road • Ealing • LONDON W5 5RQ

Tel: 020 8912 1010

Fax: 020 8912 1011

info@sprezzatura.com

SEN

November 2004

Sprezzatura's Electronic Newsletter

For Revelation developers by Revelation developers

MAKING DATABASES HAPPEN

SPREZZATURA

Contents

Welcome	3
What have we been up to?	3
Office Move	3
Revelation's Movements	4
In This Issue	5
Skeleton for OI – Delapsus Resurgam	6
Commuting Your Execution	6
Walking the Walk	7
Talking the Talk	21
The Universal Driver Hints and Tips	27
Errors commonly associated with the Universal Driver	28
S/List 3.0 – Changes to License – Martyn Phillips	28
Volume MFSs – Aaron Kaplan – Managing Director Sprezz Inc	30
OpenInsight XP – Carl Pates – Technical Director – Sprezzatura Group	34
Enabling Visual Styles in OI	34
Turning off visual style for a control	35
Securing Menus via S/MS	38
An important note about licensing	39
Eric the Emu's AREV Trips	41
Celebrity Interview – Bill Caisley – Revelation EMEA Support	42
Peripheral Trivia	44
COPYRIGHT NOTICE	44
TRADEMARK NOTICE	44

SPREZZATURA

Welcome

Welcome – to the latest SENL from Sprezzatura... although arguably that should be the Enl from Sprezzatura or the latest SENL. I suspect I've been hanging around comp.databases.theory too long and the normalization debates have gotten to me!

What have we been up to?

Sprezzatura have been very busy over the preceding months – a lot of the sort of work we were doing has been invalidated by the introduction of 32 bit OI (you'll recall that we were successfully marketing a set of add on tools for OI that delivered true 32 bit functionality in 16 bit OI) so we've been refocusing somewhat. Our core business of being the consultant's consultancy has continued of course – we've solved long standing GFE issues for clients, customized aspects of the tool set in a way that is possible only with a very in depth knowledge of the internals, provided tools for tracking Universal Driver performance, delivered AREV compatibility tools for new converters, provided in depth XML tools for clients allowing them to go above and beyond that which is possible with the core toolset and encapsulated some very cool OLE technology in OI systems.

Of course we haven't been neglecting our existing user base especially our S/Web users. For those of you who aren't familiar with S/Web it is our scalable middleware solution for high performance web sites back ended with AREV or OI. We have new versions planned for Q2 next year that will deliver even more bang per buck but in the meantime we've revised the pricing to make it more attractive to existing developers with a lower upfront cost and a weighting towards actual deployment. Our biggest S/Web site is currently delivering upwards of 2 million hits a day so we're very confident of the scalability! You can read about a couple of our success stories at:

[TV Shopping Network: Huge Growth With Small Costs](#)

[Courier Services Firm Brings Mission-Critical "AREV" Application To The Web Using OpenInsight](#)

[Sprezzatura and The United States National Weather Service](#)

To speak to someone about how S/Web can help your web enablement just drop us a line at sales@sprezzatura.com.

Office Move

For about a decade we've been based in the same building in Ealing – a town on the outskirts of London situated conveniently for London's Heathrow airport. We love the location – not least because it IS so convenient for Heathrow – enabling us to get to most of our US customers for less than the

SPREZZATURA

cost of internal US domestic flights! This has made our people such regulars on Virgin Airlines that they now help out on moderating some Virgin Websites online! However with recent growth we've finally been forced to move but we refused to give up the convenient location without a struggle so we've moved just up the road! We're still in Ealing but in newer more modern offices with a new VOIP switchboard, faster internet access and all mod cons. The best thing (from the perspective of your editor) is that we now ALL have Windows (yours truly used to be in a box with no windows) and air conditioning! I know it will come as a surprise to some of you but a lot of UK offices don't have air conditioning! Those naughty Elizabethans just didn't think ahead when designing national monuments! The view from outside makes stepping out onto the roof a delight so courtesy of a staff member here's one of the views!



Revelation's Movements

Again we must apologise for the highly irregular publication schedule for SENL – the problem is (as any consultancy knows) these sorts of endeavours are normally fitted in around more normal paying work and there has been quite a lot of this about with the continuing success of OI 7.01 and the forthcoming release of OI 7.1. The community is continuing to expand, attracting former AREV developers along with some stalwarts who up until recently were still using RevG! We must also give a

SPREZZATURA

warm welcome to those of the Pick community who have joined the Rev community as a result of Revelation's efforts to remind them of our existence!

There was a time when Pick was a four letter word in the Rev community as the old management strove to make something of the product that it was not – do the words “We are a Lotus Notes Software Company” ring any bells?! The simple truth is – without Pick there would have been no Revelation and we owe that community a debt of gratitude.

In the UK and Europe it has been gratifying to watch the efforts of Revelation UK bearing fruit – especially in relation to raising market awareness. It seems that a month doesn't go by without some new mention in the press. One of the most impressive achievements recently was a two page article celebrating the success of one of Sprezzatura's favourite VARs – Social Software. The success of the jointly developed XML pilot scheme for the Government linking Youth Offending Teams into the Criminal Justice system was covered in a two page article in Government Computing, the UK's leading magazine for the Government Sector.

In the States Revelation have been making headway as well with a number of new User Stories and additional press coverage.

Planning is already well under way for the next conference in Las Vegas and as ever Sprezz plan to be there in force. New Orleans was a great success for all concerned and there were some stunningly informative presentations. This year the management team at Revelation are pulling out all of the stops to ensure that the quality of presentation is higher than ever and standards have been set to ensure that all contributors will be even better than last year. The bar has been raised at Sprezz Towers and preparations are already under way to ensure that our professionalism is not drowned out by our spontaneity even if this means slightly more scripted presentations! Our flights are already booked so we look forward to seeing as many of you there as possible.

As an adjunct to the article by Bill Caisley don't forget to check out Revelation US's interesting article on using the Universal Driver to hide the DOS files from users at <http://www.revelation.com/knowledge.nsf/07dbcbabb6b3e379852566f50064cf25/eac117ad062bc8f785256f4a004e04de?OpenDocument>. This complements Revelation UK's article (another Bill Caisley assisted by Kevin Ruane) at <http://www.revsoft.co.uk/Downloads/hidinglkov.pdf>.

In This Issue

One of the incidental giveaways at the last conference was a reincarnation of our favourite utility for AREV – Skeleton. In this issue we discuss this utility and attach a checkout for you to try yourself.

We've continued to work closely with both end users and developers globally. The past 12 months have seen Sprezz personnel in the UK, Europe, America and the Caribbean working with some of the

SPREZZATURA

biggest and smallest names in our community! We always try and ensure that our pricing is competitive with market rates and we make sure that if you're not geographically adjacent to our resource that you don't suffer because of it! So if you've got any projects that you feel Sprezz expertise would be useful on, drop us a line at sales@sprezzatura.com!

As ever we've been busy creating new product – sometimes as an internal project that just grew and other times as a joint project with an existing client where the client pays a lesser amount to develop a utility they need so that we can then offer it to others in the Rev community. One such routine is our Menu Security Utility S/MS which we review in an article later in this issue.

We're also pleased to finally have caught up on documenting our S/List 3.x product. This coincides with an alteration to our licensing and support policies so we'd refer you to the article on this too. Those of you who aren't yet S/List users might like to follow the links to check the documentation to see if it could be of interest to you.

Finally we're pleased to welcome back our usual stalwart contributors to SENL – Aaron Kaplan with an article on the use of Volume MFSSs and an apparently resurrected Eric the Emu who is playing with keyboard trapping. For those of you who've been dealing with EMEA support we also provide an introduction to Bill Caisley, Revelation Software UK's front line support and as ever the redoubtable Carl Pates contributes a cute but useful tip to those of you wanting to improve the look of your OI apps under XP. (As an aside if you've recently moved to 7.0+ from earlier versions you might want to take advantage of SRP's free utility to help with this at www.srpcs.com). Finally Andrew McAuley provides a couple of articles on Sprezz product and in the next S/ENL a guide to Context Menus for programmers – just in time to have this replaced by a 4GL method in OI 7.1!

In closing - a query for those of you making heavy use of a Rev system. When running end of year type reports against your system, do you find that data entry slows even when using Network products? We don't have an answer yet we're just interested in your real life experience.

Thanks for continuing to be interested in our outpourings – if you have any suggestions for information you'd like to see documented then don't hesitate to ask!

Regards

Your friends at Sprezzatura

Skeleton for OI – Delapsus Resurgam

Commuting Your Execution

Way back in the day it seemed that everyone agreed that writing commuter programs (one program to encapsulate all of the logic associated with an entry form) was a great idea. Sprezzatura even

SPREZZATURA

released a utility for subscribers to REVMEDIA called SKELETON that made this ideal easily achievable. Then with the introduction of OpenInsight it seemed that people moved away from this idea. Firstly people used lots of individual events. Then when they realized what a bear this became from a deployment perspective they moved to using Omnievents as a pseudo-commuter (we were guilty of this too). Finally they came to the realization that the old commuter idea was actually still as valid as ever and commuters are once again with us.

With this in mind Sprezzatura decided to revisit the idea of the commuter program generator only this time with a slight difference. Rather than make the utility available only to those who subscribe to a paid journal we've decided to make it available as a source code release to the Revelation Community under a modified version of the GNU GPL license. We're doing this because we believe that there are others in the community who could add value and by so doing help us all in our mission to deliver better software whilst becoming more productive.

But before looking further at the distribution of the software let's take a look at what it does...

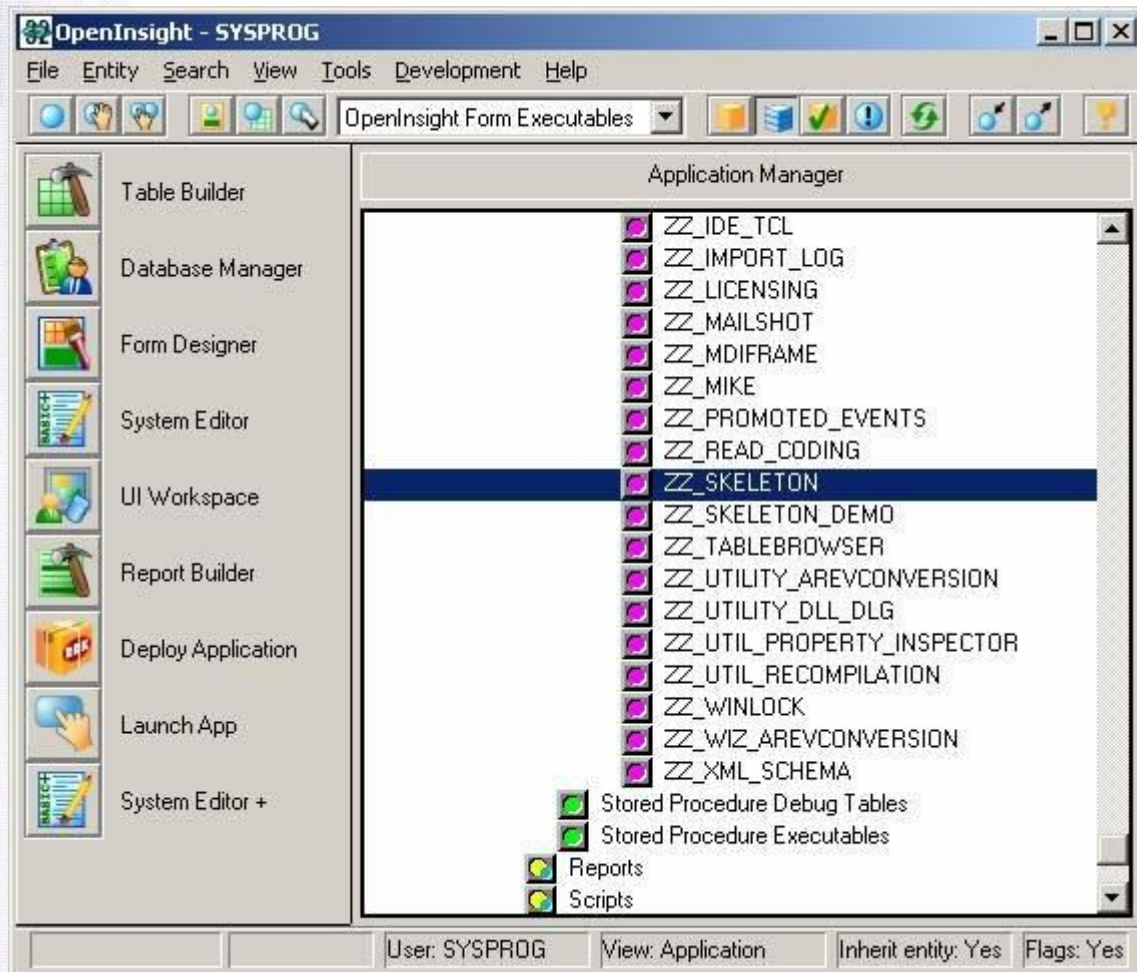
Walking the Walk

In this instantiation of Skeleton we've opted to ship as a Windows Executable called ZZ_Skeleton. It is supported by an underlying commuter program called, yes you've guessed it, ZZ_Skeleton. When the window is run it shows you a list of windows in the current application and asks which one you'd like to add a commuter module to. Once you've chosen the window it then asks you for the window and for each prompt which events you'd like to add. It then adds quickevent calls into the window and generates the source code for the supporting routine. Of course there are caveats associated with this but let's look at the routine in action before discussing these.

So to launch ZZ_Skeleton we navigate to the OI window executable ZZ_SKELETON

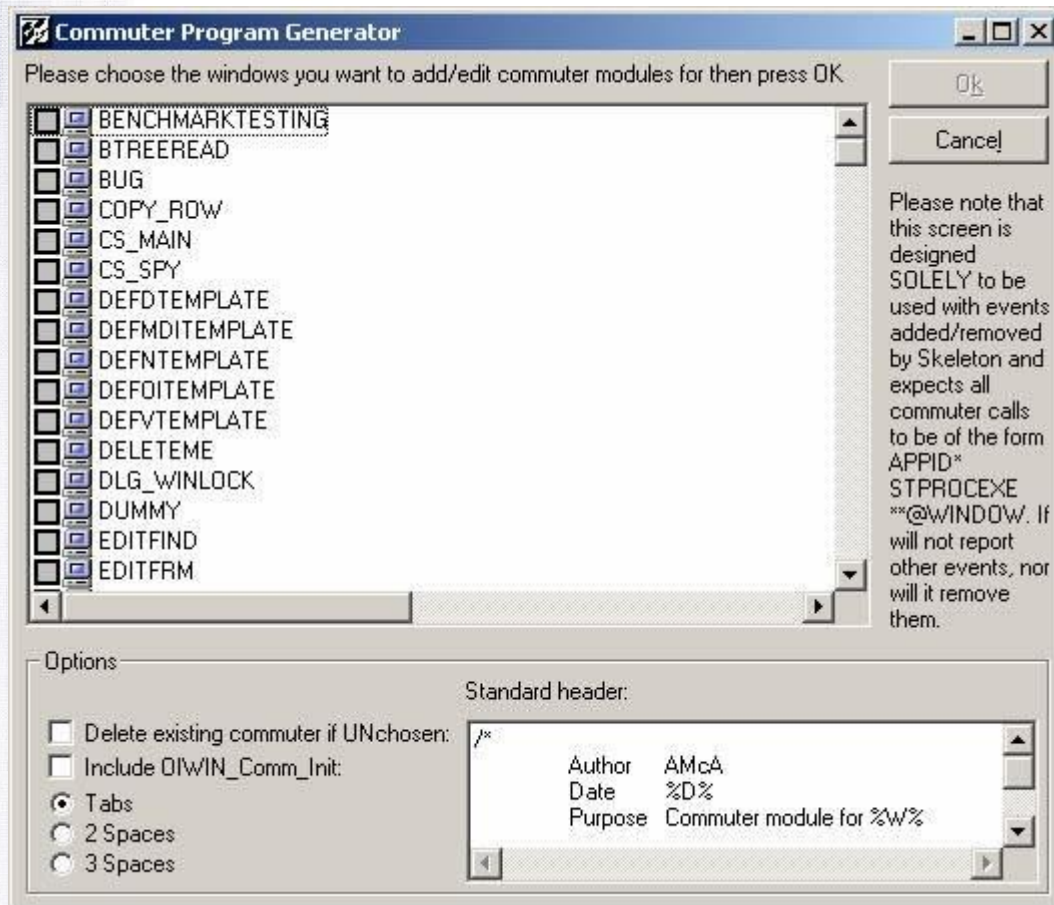


SPREZZATURA



And Shift-Enter or Shift-DbClick...

SPREZZATURA

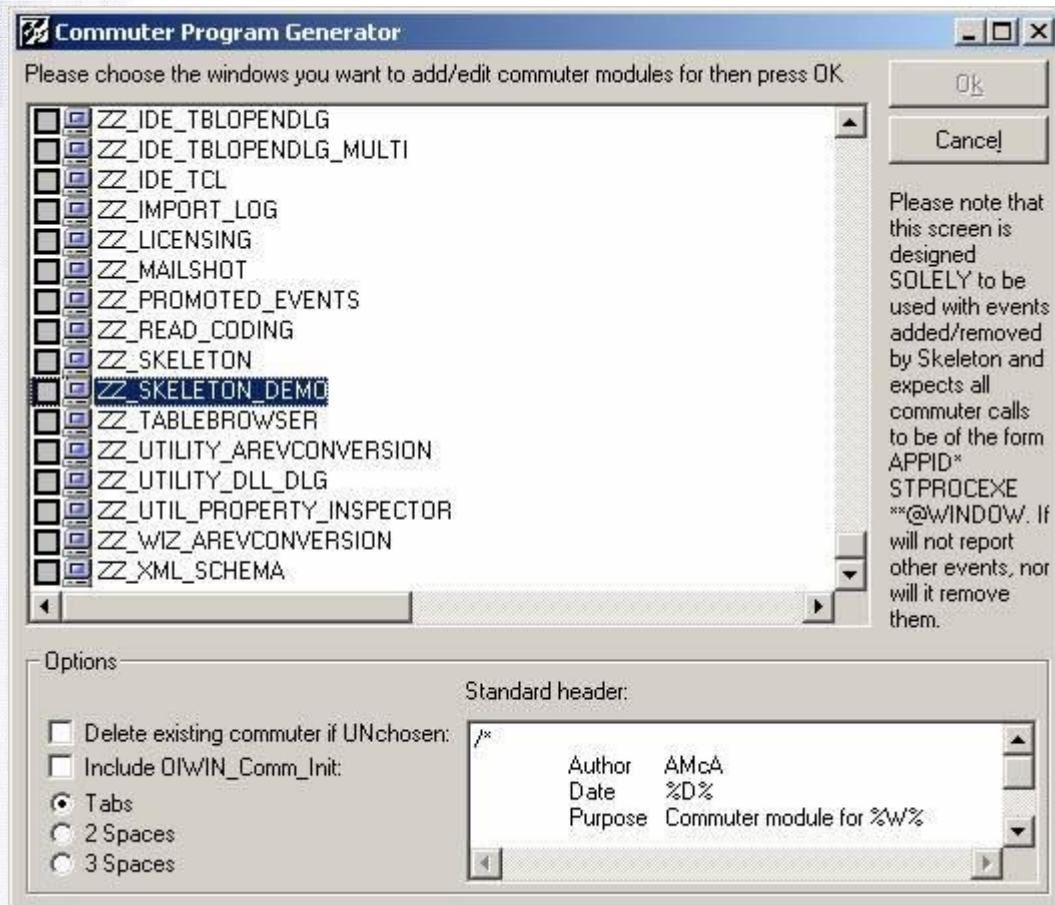


The ZZ_SKELETON window appears and the main listbox populates with all of the Windows in the current application. The Options as the bottom allow you to determine whether any existing commuter logic should be overwritten (although this is not implemented currently), whether OpenInsight Window Common ought to be included in the program (See http://www.sprezzatura.com/senl/senl18.htm#_Toc447361865) and whether the resultant commuter program should use tabs or spaces for indentation. In addition it allows the specification of a custom header for the program.

At this stage the OK button is disabled as there is nothing to do.

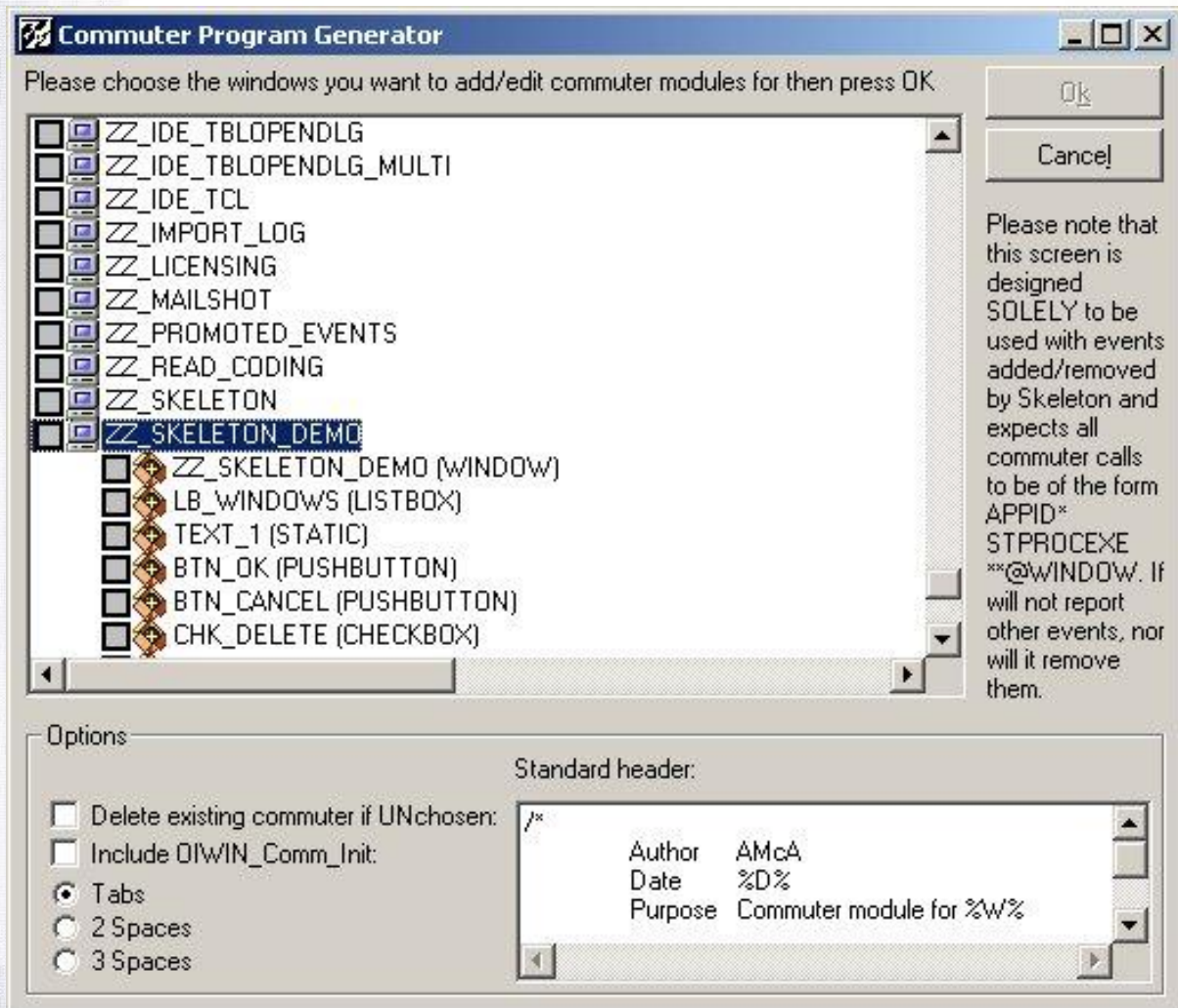
To add events in and to generate a commuter program we firstly navigate to the Window we wish to generate for – in this example we'll be all self referential and use a copy of ZZ_SKELETON called ZZ_SKELETON_DEMO

SPREZZATURA



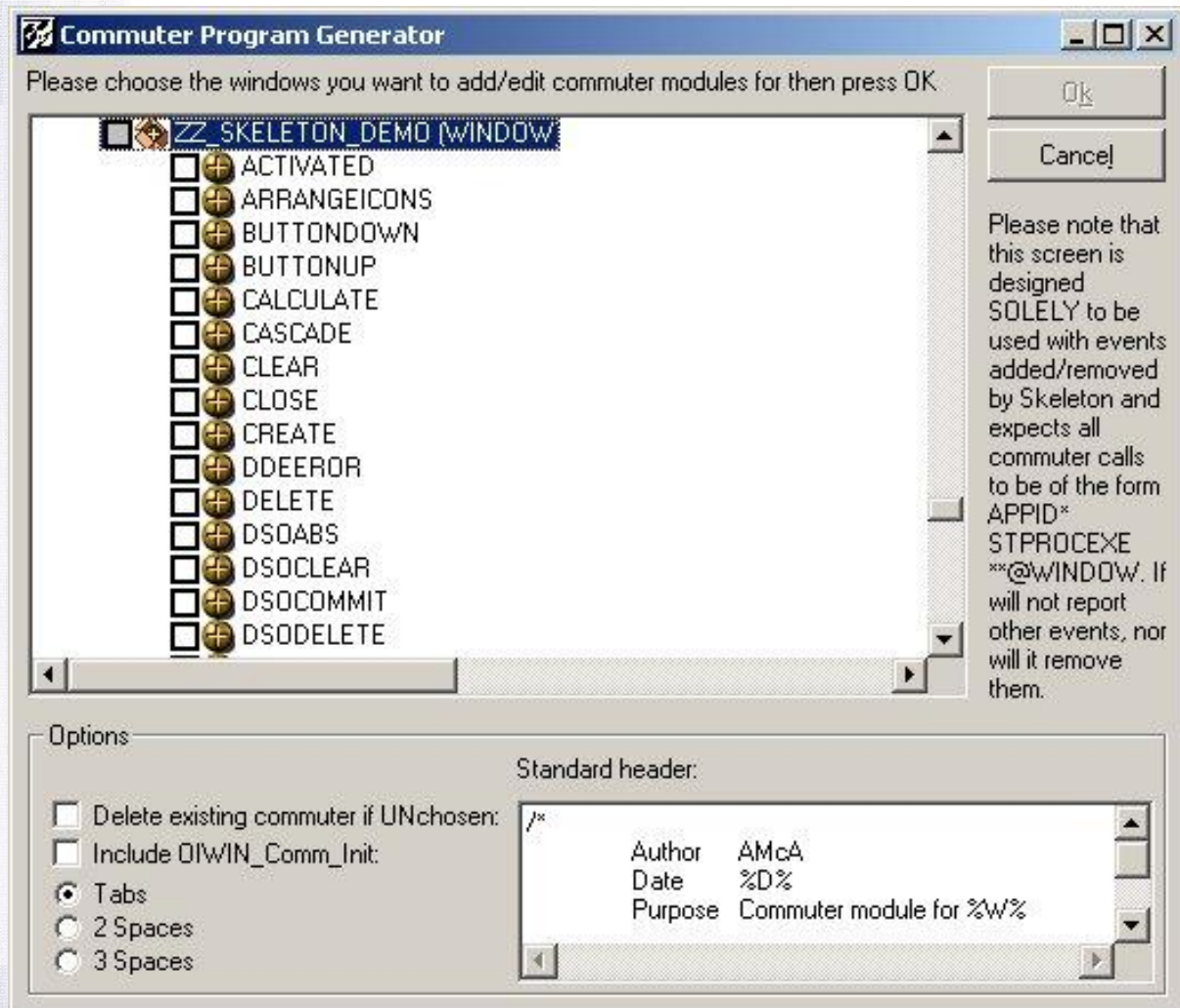
Firstly we're going to add Window events, so let's double click on ZZ_SKELETON_DEMO

SPREZZATURA



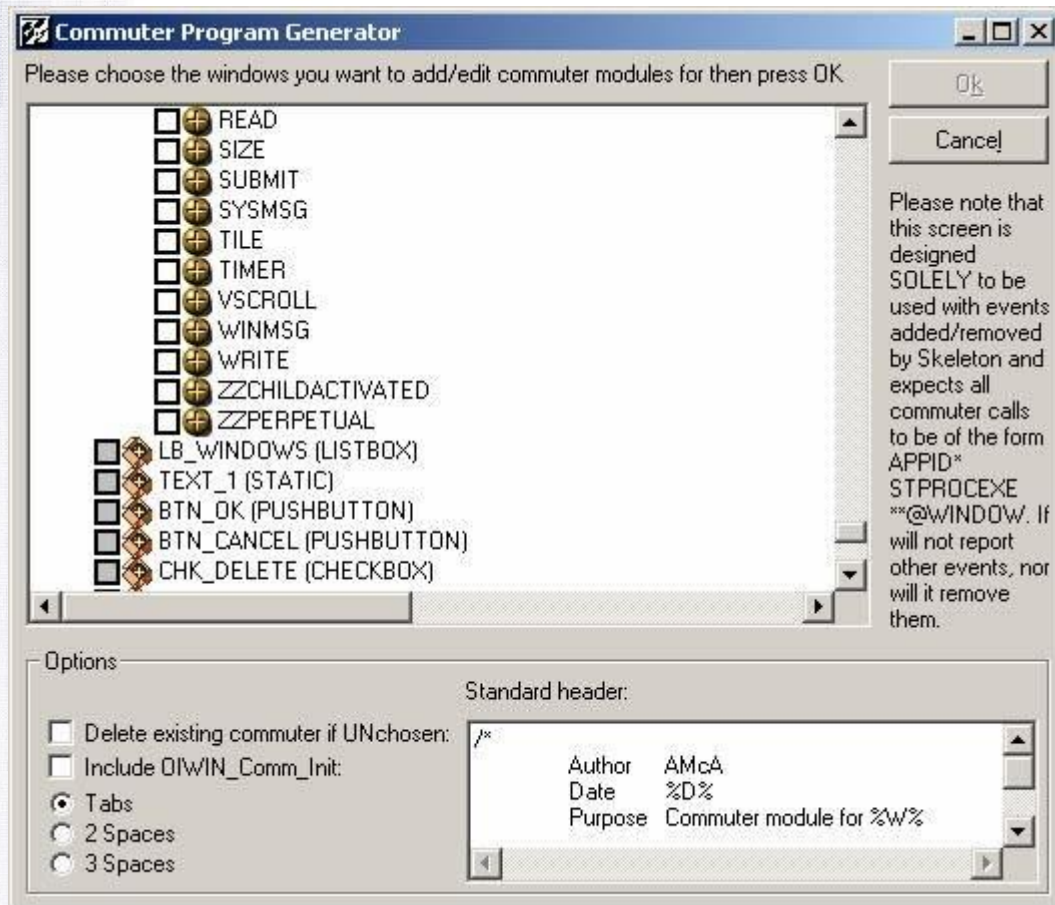
This expands all of the controls in the Window. Now we're going to double click the ZZ_SKELETON_DEMO (WINDOW) entry to see the events for the Window

SPREZZATURA



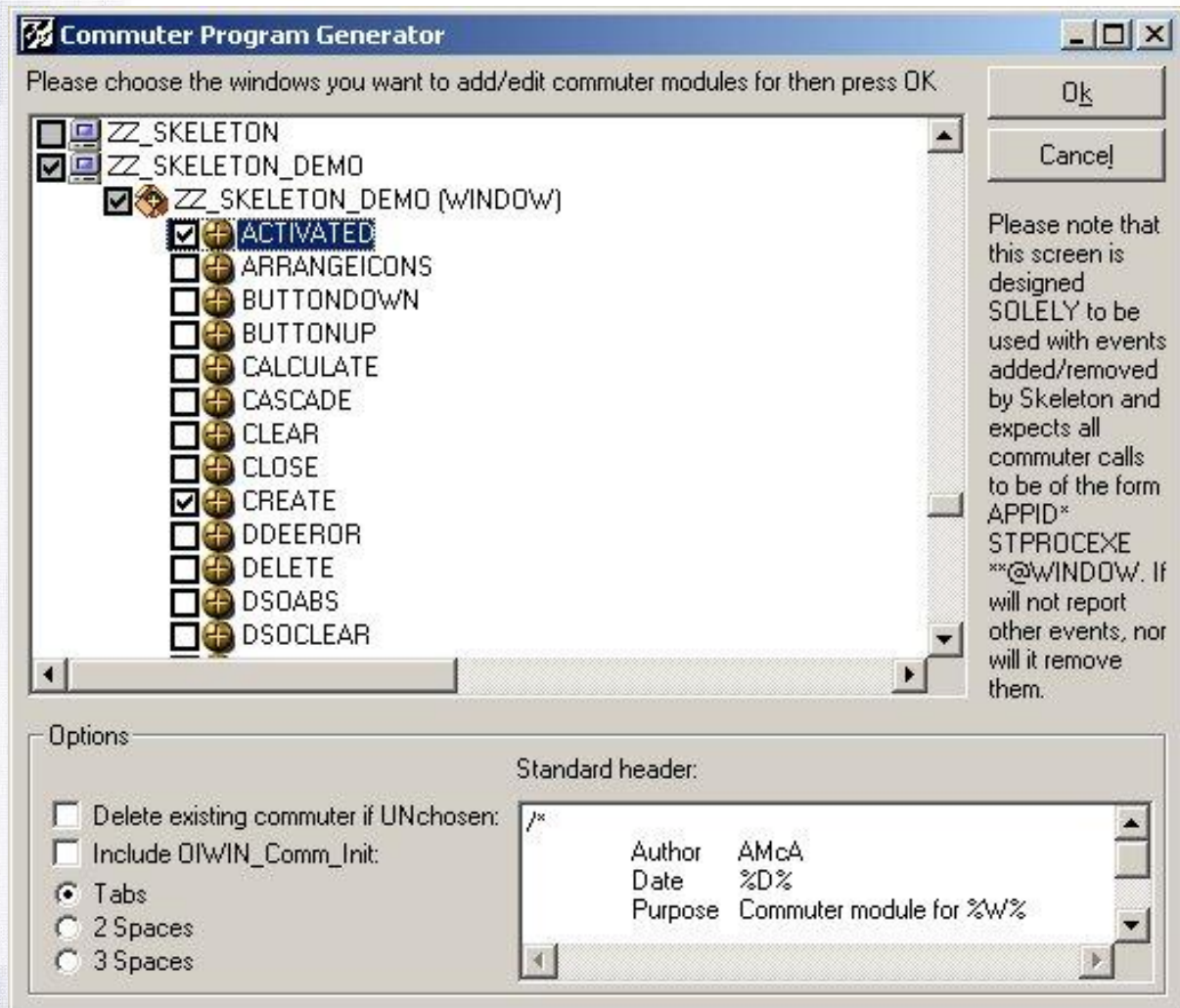
Scrolling down we'll see some "pseudo events" used internally at Sprezzatura for promoted events...

SPREZZATURA



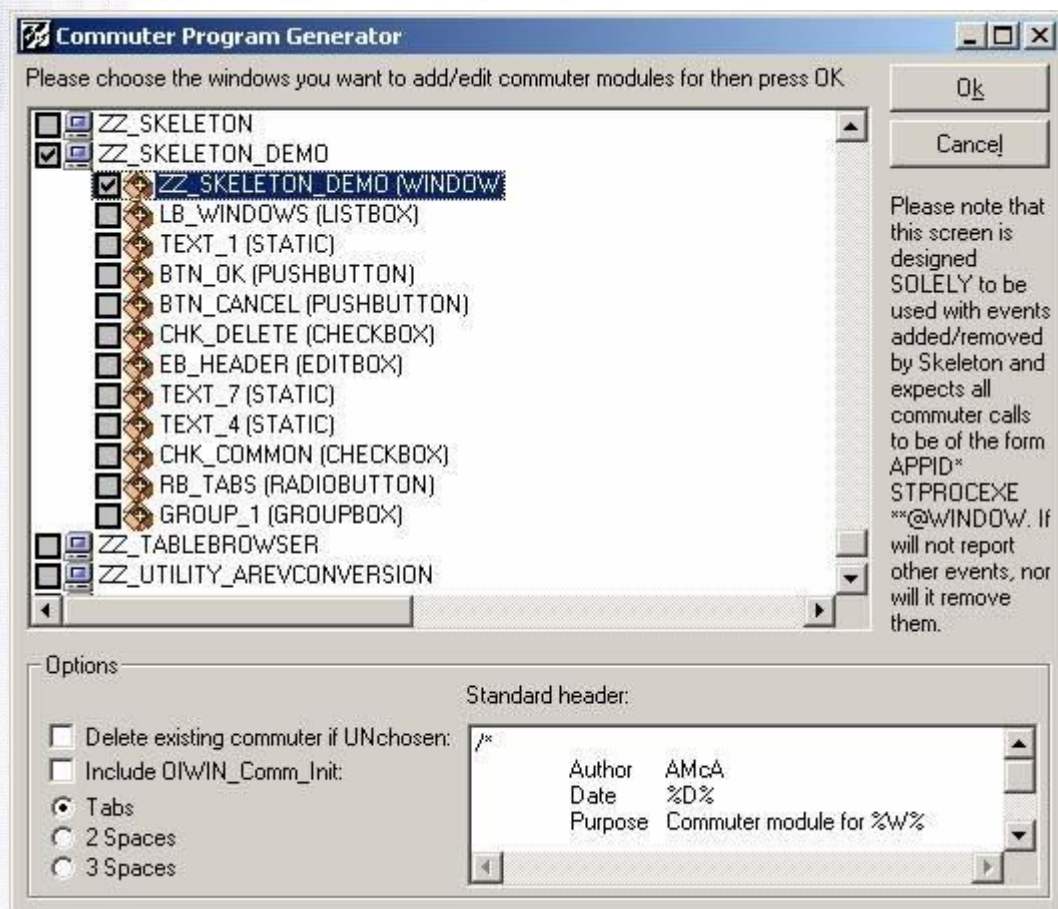
But for now we're just going to go with an ACTIVATED event and a CREATE event so double click these

SPREZZATURA



And collapse the ZZ_SKELETON_DEMO entry by double-clicking it

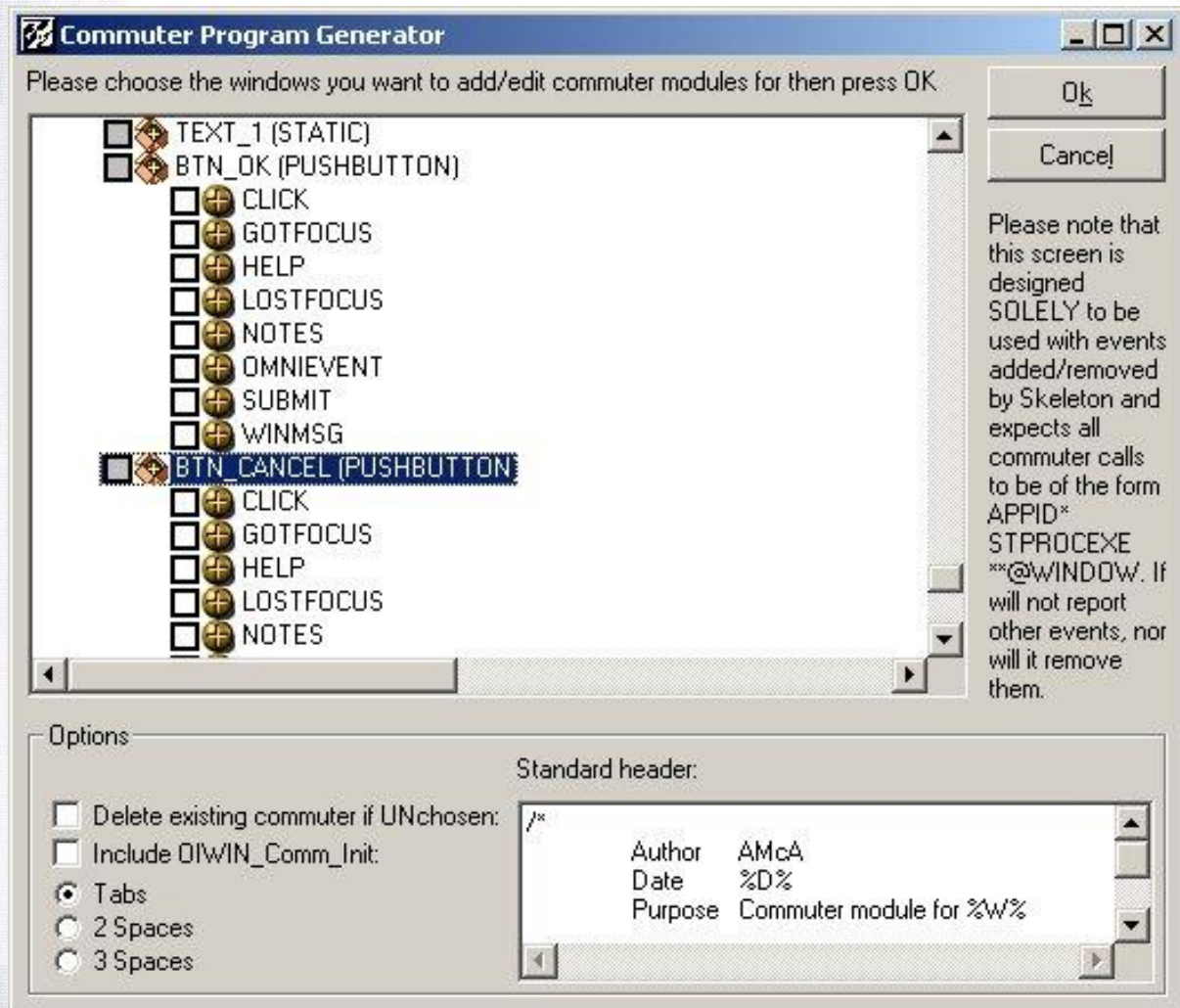
SPREZZATURA



Note the shaded check to show there are items checked in this subsection.

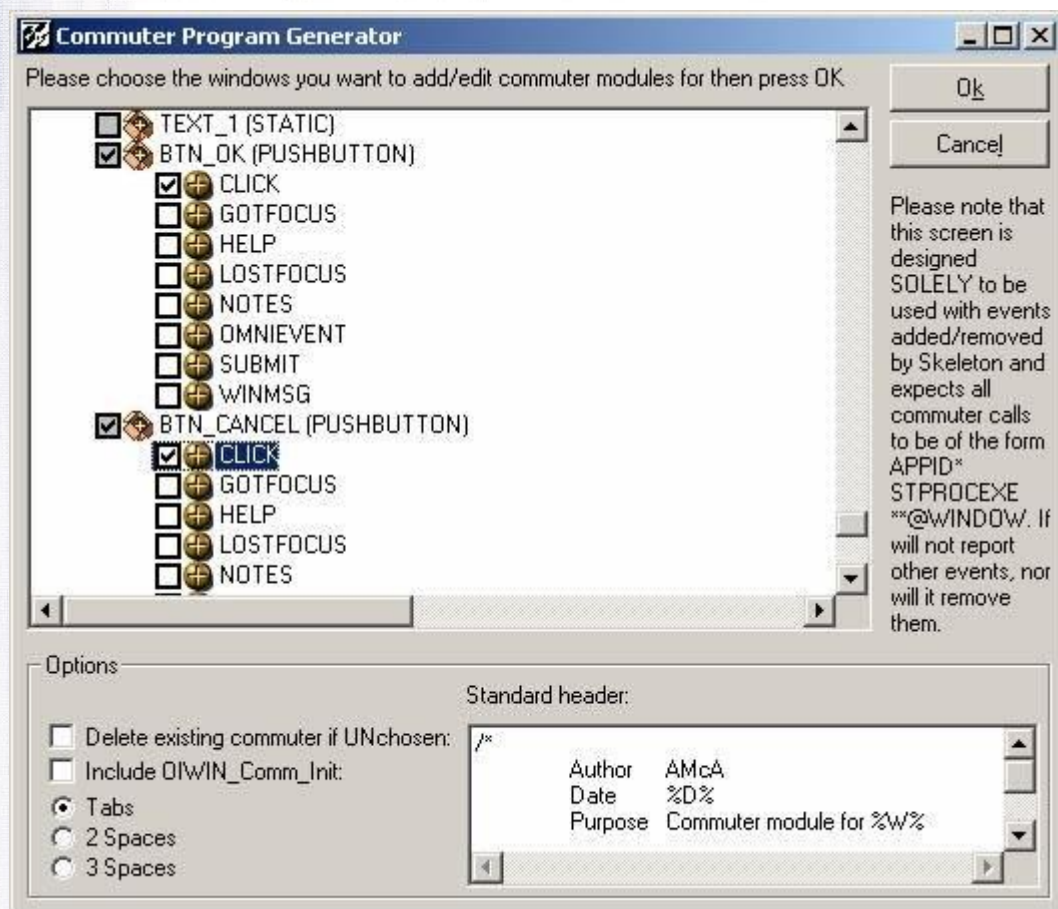
Now we're going to add a click event to the BTN_OK and the BTN_CANCEL so expand these

SPREZZATURA



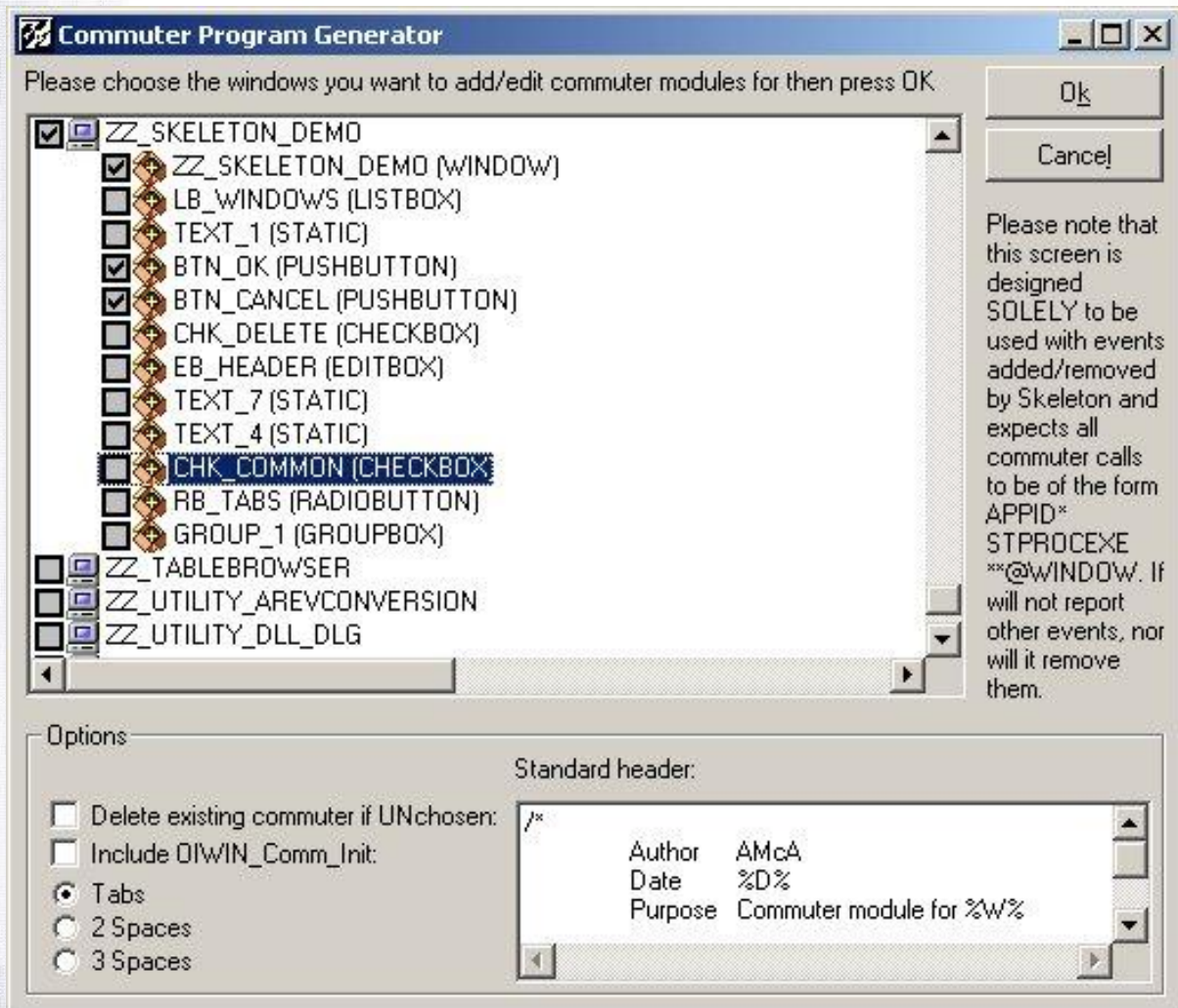
And double click the CLICK event

SPREZZATURA



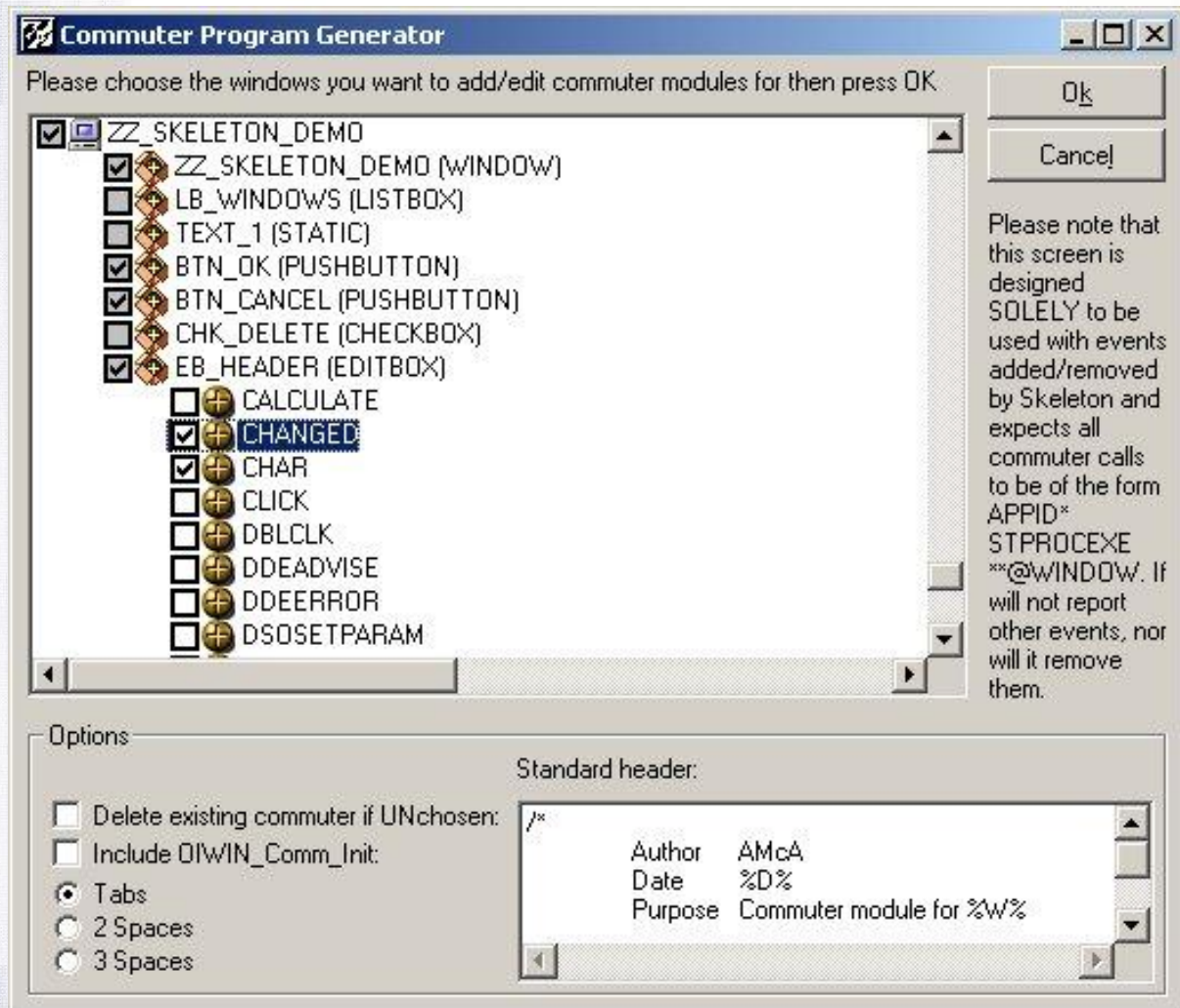
And collapse them again

SPREZZATURA



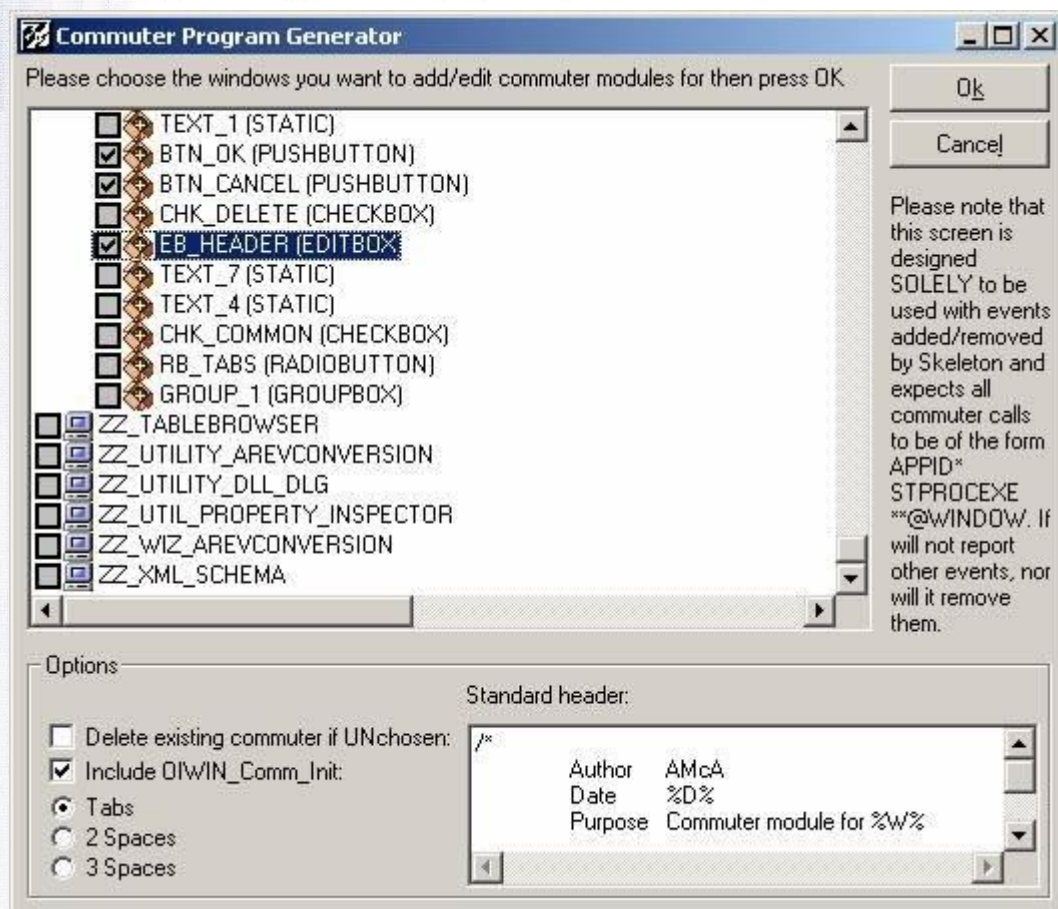
Finally we'll add a CHAR event and a CHANGED event to the EB_HEADER

SPREZZATURA



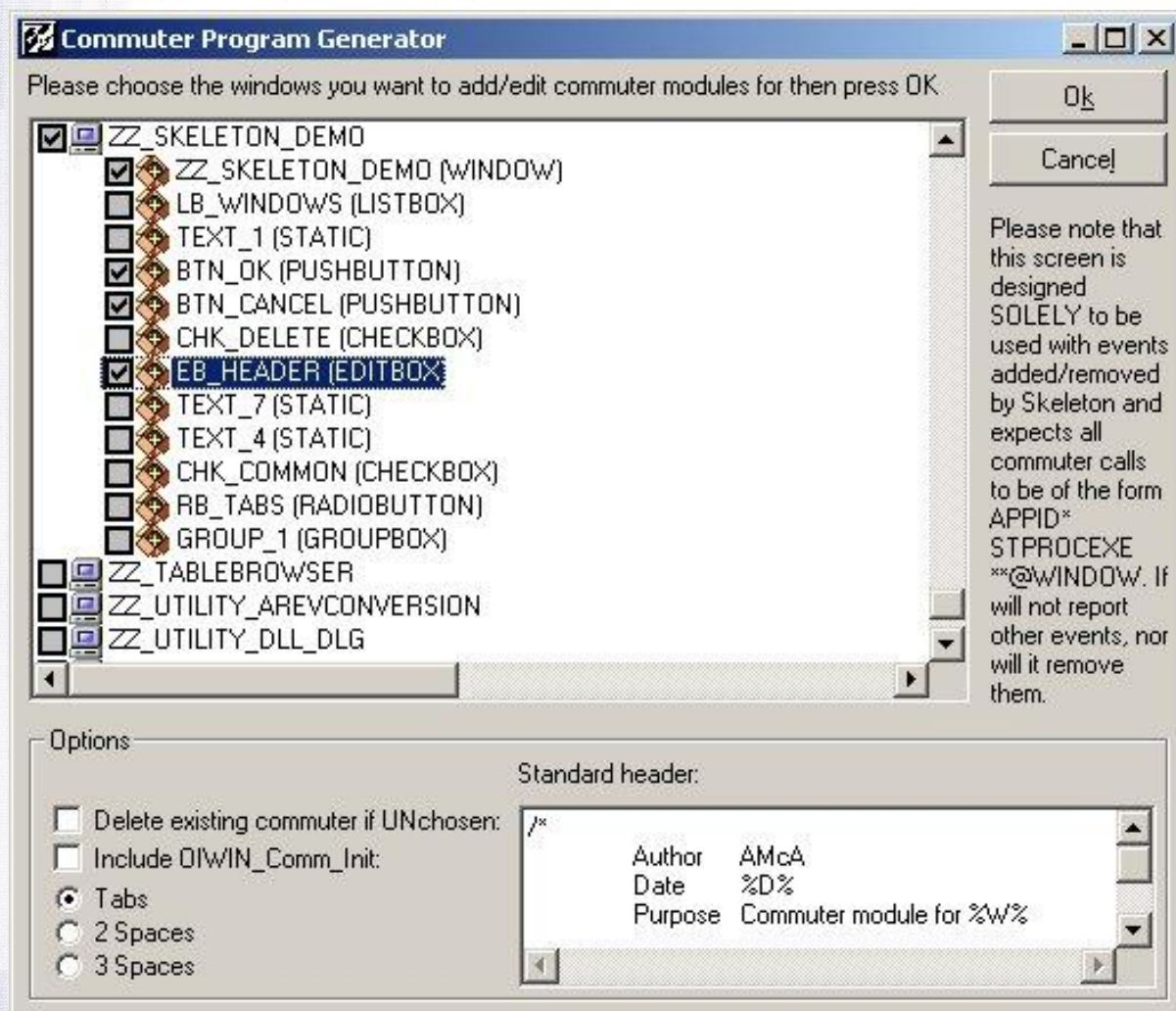
Finally we'll indicate that we want Window Common included

SPREZZATURA



And when we've finished we'll click the (now enabled) OK button.

SPREZZATURA

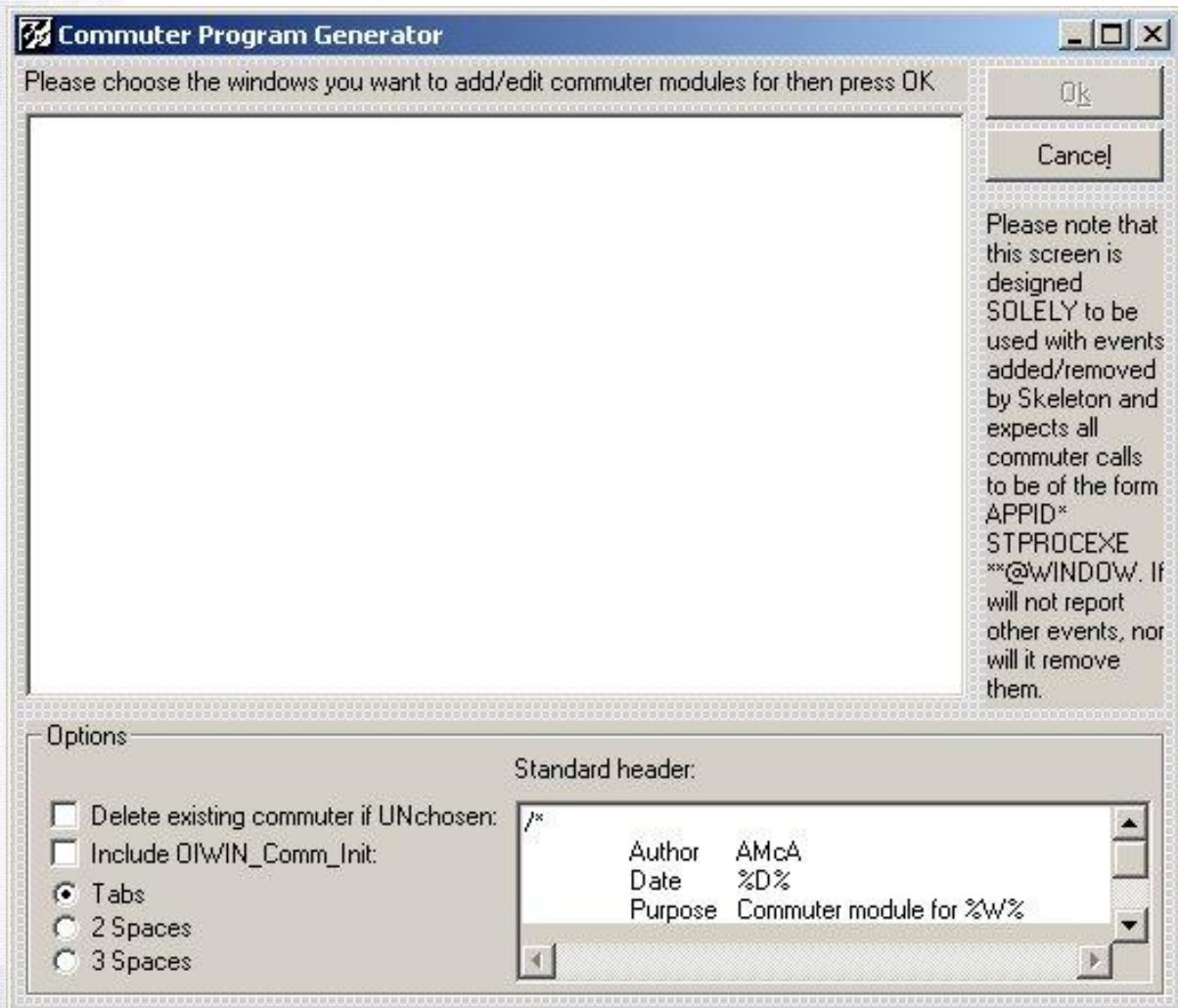


The system pauses whilst it generates the code and updates the window and then returns to the application manager.

Talking the Talk

If we now open the form in FORMDES by double clicking it

SPREZZATURA



We can see that the quick events have been added in – so for example let's double click the Window to look at the events on the window

SPREZZATURA

QuickEvent for ACTIVATED

Event: **ACTIVATED<q>**

QuickEvent Options:

- General
- Start a window
- Start MDI child window
- Execute a popup
- Execute a procedure**
- Display QuickHelp
- Display a message
- Index lookup
- Close window
- Read the row
- Write the row
- Clear the form
- Delete the row
- Internet

Send Message to:

☒ Entity ☐ Control/Window

SYSPROG*STPROCEXESKELETON_@W**

Message: **EXECUTE**

Parameters: **'@SELF','ACTIVATED'**

Return value in:

Control:

Property:

Buttons: Accept, Close, Scripts..., Help, Clear

QuickEvent for ACTIVATED

Event: **ACTIVATED<q>**

QuickEvent Options:

- ACTIVATED<q>
- ACTIVATED<q>
- ARRANGEICONS
- BUTTONDOWN
- BUTTONUP
- CALCULATE
- CASCADE
- CLEAR
- CLOSE
- CREATE<q>
- DDEERROR
- Index lookup
- Close window
- Read the row
- Write the row
- Clear the form
- Delete the row
- Internet

Send Message to:

☒ Entity ☐ Control/Window

SYSPROG*STPROCEXESKELETON_@W**

Message: **EXECUTE**

Parameters: **'@SELF','ACTIVATED'**

Return value in:

Control:

Property:

Buttons: Accept, Close, Scripts..., Help, Clear

You'll notice that ZZ_SKELETON has inserted a STPROCEXE name of SKELETON_@WINDOW. It has done this so as not to overwrite a routine with the same name as the window! If we now go to the

SPREZZATURA

system editor we can open the routine SKELETON_@WINDOW or more accurately in this case SKELETON_ZZ_SKELETON

```

Function Skeleton_Zz_skeleton(ctrlEntID,branch, param1, param2, param3, param4, param5, param6, param7, param8, param9, param10, param11, param12, param13, param14, param15, param16, param17, param18, param19, param20)
/*
  Author   AMcA
  Date    17 November 2004
  Purpose  Commuter module for ZZ_SKELETON
  Notes
*/

$Insert Logical
WinID = @Window
$Insert OIWin_Comm_Init

RetVal = True$
Declare Function Get_Property, Set_Property, Get_Repos_Entities, Utility, Msg, Generic_Event
Declare Subroutine Get_Property, Set_Property, Utility, Msg

Locate branch In "ACTIVATED,CHANGED,CHAR,CLICK,CREATE" Using ',' Setting BranchFound Then
  On BranchFound GoSub Activated,Changed,Char,Click,Create
End Else
  Call Msg(@Window, 'Invalid branch - ' & Branch & ' passed to SKELETON_ZZ_SKELETON')
End
Return RetVal

Activated:
  Begin Case
    Case CtrlEntId = @Window

```

The routine looks like this :-

```

Function Skeleton_Zz_skeleton(ctrlEntID,branch, param1, param2, param3,
param4, param5, param6, param7, param8, param9, param10, param11, param12,
param13, param14, param15, param16, param17, param18, param19, param20)

```

```

/*
  Author   AMcA
  Date    17 November 2004
  Purpose  Commuter module for ZZ_SKELETON
  Notes
*/

```


SPREZZATURA

```

$Insert Logical
WinID = @Window
$Insert OIWin_Comm_Init

RetVal = True$
Declare Function Get_Property, Set_Property, Get_Repos_Entities, Utility,
Msg, Generic_Events
Declare Subroutine Get_Property, Set_Property, Utility, Msg

    Locate branch In "ACTIVATED,CHANGED,CHAR,CLICK,CREATE" Using ',' Setting
BranchFound Then
    On BranchFound GoSub Activated,Changed,Char,Click,Create
End Else
    Call Msg(@Window, 'Invalid branch - ' : Branch : ' passed to
SKELETON_ZZ_SKELETON')
End
Return RetVal

Activated:

Begin Case
Case CtrlEntId = @Window

End Case

Return

Changed:

Begin Case
Case CtrlEntId = @Window : '.EB_HEADER'
GoSub ChangedEb_header

End Case

Return

Char:

Begin Case
Case CtrlEntId = @Window : '.EB_HEADER'
GoSub CharEb_header

End Case

Return

```


SPREZZATURA

Click:

Begin Case

Case CtrlEntId = @Window : '.BTN_CANCEL'

GoSub ClickBtn_cancel

Case CtrlEntId = @Window : '.BTN_OK'

GoSub ClickBtn_ok

End Case

Return

Create:

Begin Case

Case CtrlEntId = @Window

End Case

Return

ChangedEb_header:

Return

CharEb_header:

Return

ClickBtn_cancel:

Return

ClickBtn_ok:

Return

We can now modify our code to perform the functionality we require.

CAVEATS

Because the ZZ_SKELETON routine puts calls in as QuickEvents there are certain things that it will not trap efficiently. As a simple example, if a WRITE event is added the code will not be called until AFTER the WRITE has taken place. The ZZ_SKELETON code could be modified to insert a direct call into the

SPREZZATURA

Script for the WRITE but this has not been done in this version. Similarly with this version if code exists on file you will not be able to generate new commuter code for it.

Finally this code generates small amounts of Sprezzatura specific code. This can easily be removed from the generator but we have left it in to stimulate thought in adopters.

[Click here](#) to download the Skeleton.zip which contains an RDK install of the utility along with the product license and a readme. As usual with our FREE utilities the files are supplied as is and we cannot accept any liability for any loss through the use of the files.

The Universal Driver Hints and Tips

Bill Caisley Revelation Software EMEA Support

With the introduction of the new Universal Driver (UD) there are a few things you need to be very aware of.

OI7 ships with the UD client installed and it will default to this unless told to do otherwise when installing OI7. If you are using the UD this is not a problem but if you choose another driver at installation and then wish to upgrade to the UD at a later stage be careful. Currently, if you install the UD onto OI7 and install the client again it corrupts the LH3.DLL and the LH3CLNT.DLL and presents an ENG0800 error when attempting to start OI. Revelation is looking into correcting this in the near future. To fix this you will have to replace the two corrupt DLL's with the DLL's from a clean install of OI7, which is available on the latest "Revelation End of Year CD".

One of the exciting features of the UD is that it removes the 64KB restrictions of previous drivers and increases the frame size to a possible 4GB. To achieve this, the header section in the Linear Hash files has been changed to add extra addressing bytes. The UD is backward compatible and can understand the header information of older drivers but the converse is however not true. The older drivers could not have envisaged the new header changes.

What does this mean?

Files that are created using the UD will not be recognised by any of the older drivers. OI with an older driver will not "see" the new files. So any development that creates new LH tables using the UD will not deploy into a system utilising an older driver. This is especially true when using the RDK to create upgrades for existing clients. You could however change the driver in OI7 to All Networks 2.1, for example, and then create the new tables which will make them useable in any of your systems.

The REVPARAM file should now not create any more confusion; it has been made very simple. The UD looks for a REVPARAM file in the directory of the .LK file being addressed. If it is not found, the UD will search up the directory tree all the way to the Root until it knows that there is no REVPARAM file

SPREZZATURA

present or reacts to the REVPARAM it finds. So with the UD you now only need a single REVPARAM file, if you require it. The safest place to put it is in the OI directory where Oinsight.exe resides. However if you have multiple copies of OI running on the same server and they are in "parallel" directories, you could have a REVPARAM file in each OI directory or just one on the root drive.

Remember when dealing with the UD your best friend is the new updated release of the **Universal Driver Installation Manual**. Follow it carefully and you will stop almost all of the headaches.

Errors commonly associated with the Universal Driver

ENG0800	As described above.
FS231 & FS1019	These errors are related to there being a problem with the REVPARAM or .INI file/s or entries in the file/s. There is a check sheet available on http://www.revsoft.co.uk/fs231_ts_guide.htm for the FS231 error as it can be very elusive.

S/List 3.0 – Changes to License – Martyn Phillips

As you will no doubt be aware, S/List has recently undergone a major overhaul, and during this period many of the requests from our customer base have now been included in the latest versions. This project has resulted in many different versions of the product being emailed to different people, where their issues have been addressed in that version, on a case by case basis. While new functionality and fixes were easily addressed in the product, there remained one fundamental question that many of our S/List users had – "What about support and maintenance and how can I be sure I have the very latest version?"

Many of you will have recently downloaded and installed the latest version from our web site, seemingly free of charge. This was the final stage in a long pilot during which we were working on a new mechanism for making upgrades more easily and readily available to our users. I am pleased to say that the system worked perfectly, with licenses less than 12 months old being upgraded without any communication with us other than downloading the software itself from our web site. Those people with older versions of S/List 3.x were, as expected, prompted for an installation code and those people needed to contact our office to obtain the said code. As a good will gesture these have been supplied free of charge during this trial period.

So what does this mean for our existing and new users?

How you will be affected depends greatly on when you bought your original S/list product license:

**S/List bought and supplied as version 3.01.44 or later.**

You will not have to do anything. You have and will continue to benefit from free support and maintenance for the 12 months following supply of your original S/List installation files and your unique installation code. At the end of the 12 month period you will have the opportunity to renew your support and maintenance as detailed below.

S/List 3.x bought prior to version 3.01.44.

You will need to contact our office to obtain a new installation code to install the latest version of S/List. As a goodwill gesture, we are continuing to provide all S/List 3.x customers with FREE upgrades and maintenance until the end of the year (31st December 2004). Please be sure to download your FREE upgrade as soon as possible to avoid any disappointment.

S/List 2.x users.

All S/List 2.x licenses will need to be upgraded to version 3.x through the purchase of a 2.x to 3.x upgrade. This upgrade will also include the free support and maintenance for 12 months from the purchase of the upgrade. Please contact mp@sprezzatura.com for the cost of the upgrade and details of how to obtain your version 3.x upgrade files.

New licenses.

If you have not yet bought your S/List 3.x license and are planning to do so in the near future, you can purchase the full product, with support and maintenance for twelve months, at the current prices £99.00 (\$169.00 USD) for a single user license.

To address our customers' requests for a structured support and maintenance scheme, we will be creating a new S/list package which will be formally launched on 1st January 2005. This new package will replace the old 'product only' package that we have previously offered. Further details will appear on our web site very soon with the new pricing coming into effect on the launch date, however it will consist of the following:

All new S/list licenses will cost £125.00 GBP (\$225.00 USD) and the delivered package will consist of:

- a) the S/List Designer software (development license)
- b) documentation in electronic format
- c) 12 months email based support (support@sprezzatura.com)
- d) 12 months upgrades (available from www.sprezzatura.com)

SPREZZATURA

On the anniversary of the original purchase you will have the option to extend your current support and maintenance for further 12 months or remain with your then current version and purchase support as and when required on a 'pay-as-you-go' basis. The cost for the support and maintenance renewal will be £50.00 GBP (\$90.00 USD).

We believe that this new way of supplying S/list will ensure that you and your users will consistently have access to the very latest S/List features and innovations via our web site and access to our support and S/List development team for prompt response to your S/List technical questions.

p.s. The new pricing will not become valid until 1st January 2005. Please be sure to order any new S/List licenses that you or your users require before the 31st December 2004 to benefit from the new package at the old cost.

Further details about S/List, the user manual and ordering details can be found by [clicking here](#).

Volume MFSs – Aaron Kaplan – Managing Director Sprezz Inc

One of the least understood, yet more powerful components of all Revelation products is the Modified Filing System or MFS. This tool goes back to somewhere around Rev E and was then known as an AFS (Alternative Filing System).

An MFS can come in three basic types. The first type is a simple or standard MFS. This is what most people talk about when they refer to an MFS. In this incarnation the MFS is a shell around the base filing system (or BFS). The concept behind it is beautiful in its simplicity.

Here's the contents of the SYSTABLES record for the SYSREPOSLOCKS table

RTP57*REVBOOT

SYSREPOSLOCKS

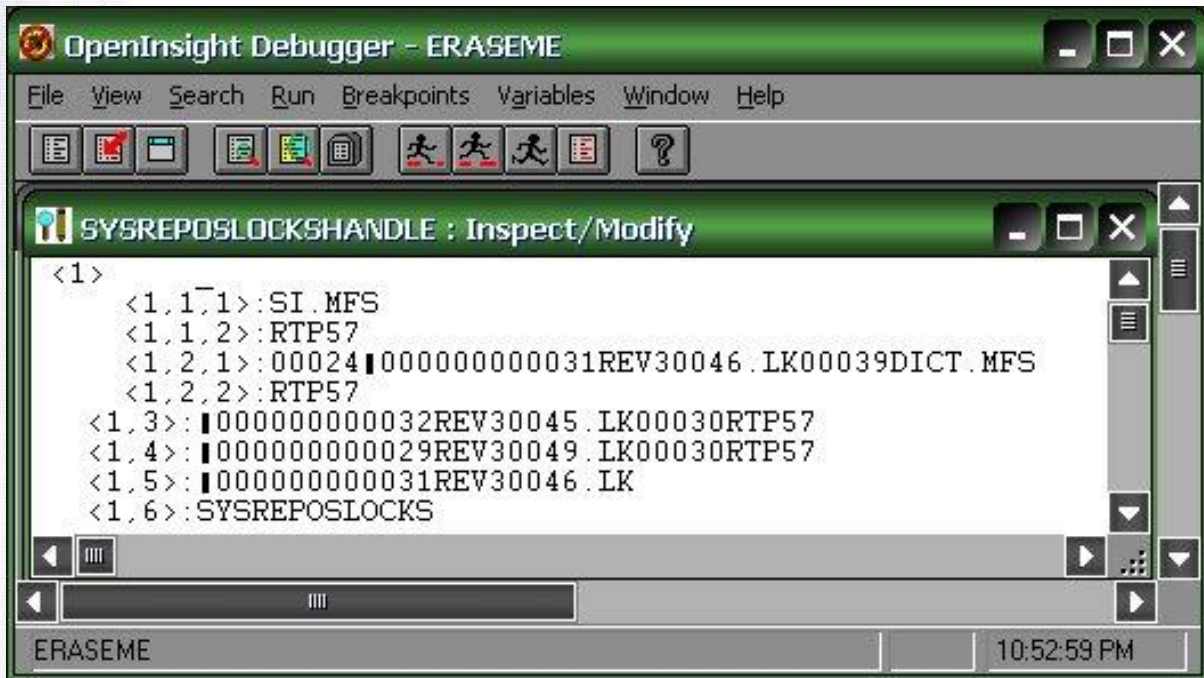
GLOBAL

SI.MFS²RTP57

According to the dictionary of SYSTABLES, field 4 is FS_LIST, which is short for Filing System List. This field contains the list of all the filing systems associated with this particular file.

When the system executes file based opcodes (READ, WRITE, OPEN, etc) the opcode interpreter works a little bit of magic. Here's a view of the SYSREPOSLOCKS table's handle after an OPEN statement

SPREZZATURA



Notice that the first @VM delimited field is the contents of field 4. This is the file system list, and it's how the system knows how to access the file. The file based opcodes pull this value out of the handle and then extract the first subvalued field. In our example this would be SI.MFS.

The system then generates a function call with the rest of the information in the statement. NOTE: This is the statement, as in the READ or the WRITE, and not just the rest of the handle.

This function is called in the format

```
@NextFS( Code, FSList, TableHandle, Name, FMC, Record, Status)
```

The MFS programming manual from Revelation will give you detailed information on what each variable means, since they can mean different things in different contexts. The first three variables are always consistent, and are the only ones important for our discussion here.

Code contains a number code informing the program what branch to execute.

FSList contains the remaining FSList as known by this program.

TableHandle contains the rest of the table handle (or where it's returned in the case of an OPEN).

Of these, FSList is the most important. The filing system logic of Revelation products is based on chaining of programs, and this chaining is based on FSList.

SPREZZATURA

When your MFS is called, the first thing it needs to do is to strip off the first filing system in the list. That program then becomes the next program in the chain. Your program does what it needs to with the data: archives it, compresses, encrypts, validates, adds tracking information, or whatever purpose you can come up with, and then calls the next program down the chain. As the next FS in the list is called, the preceding programs may have added or removed various parts of various sections of the file. Your MFS does not care what happens to the data above itself on the chain, nor for that matter, what really happens to it down below.

You should be able to see that this is a layered set of calls, giving each operation a pre and post call. You should also see how this allows you to modify the behaviour of the base system. This is why it's called a modified filing system.

That was a quick and brief explanation of the simplest form an MFS can take.

The next form is more complicated than that. In this form, we'll use the MFS concept to allow our MFS to act as a BFS. The programming behind this is very simple, but the concept and potential is what takes time to get used to.

What we do is set up our program to act as the base filing system for a volume, which places it on a par with RTP57, RTP52 and RTP59 (LH, LNK and ROS). Our program then receives all calls for access. The trick to making this work, is calling the appropriate file system yourself. You already know how to pass to the next FS list.

In this format, you simply call the actual base filing system you want:

```
RTP57( Code, FSList, TableHandle, Name, FMC, Record, Status)
```

This passes the code right down to RTP57, which does the READ or the WRITE or the OPEN or whatever other call is required.

There's not really much of a difference in the programming concept behind it. The difference is in the scope of the system and its effects. For those who think this is an idea that does not really have much use, I refer you to the inner workings of REPOS_BFS, which does exactly this. REPOS_BFS parses the record, manipulates the various repository index files, encrypts the records, then calls RTP57 to do all the file access. It's simple and powerful, powerful enough that most of you probably thought it was a completely new filing system.

Some people have taken this concept and used it when they wish to place an MFS on a large number of files. If there are hundreds of files that require auditing, for example, placing an AUDIT_MFS on each one then this can be a time consuming task. However, there is an easier way to handle this desired functionality, and that's through the third type of MFS, the volume MFS.

SPREZZATURA

A volume MFS is like a file MFS, but instead of shelling all the access to a file, it shells all the access to a volume. You write the volume MFS the same way you would write a file based MFS. The difference is in the installation. On a volume MFS, you need to add the information to the SYSPTRS table. This is a little used table in OpenInsight. Those of you with an Advanced Revelation background might remember the VOC table. Records in the VOC table did not correspond to any particular format, and could mean program definition records, batch TCL commands and volume pointers, just to name a few.

SYSPTRS in OpenInsight purpose is to hold all those old volume pointer definitions in ARev. Surprise, surprise, Revelation did not forget about that little piece of the system.

The format of SYSPTRS is quite simple:

- <0> Alternative_Volume_Name
- <1> VOLUME (always)
- <2> FS listing
- <3> Real location of the volume

For example, if you wished to have your F:\SHARED\SYSTEM\APPS\SPREZZWARE\UTILITIES volume stored as something easier, you could have a record like

- <0> SPREZZUTILS
- <1> VOLUME
- <2> RTP57
- <3> F:\SHARED\SYSTEM\APPS\SPREZZWARE\UTILITIES

Then, if you need to attach or manipulate that volume, you just treat it as the name SPREZZUTILS, which is a much easier to remember name.

Adding your MFS to this record is just a simple matter of adding to the FS list in field 2. Delimit the values with value marks, and it's all done.

- <0> SPREZZUTILS
- <1> VOLUME
- <2> SECURITY_MFS²RTP57
- <3> F:\SHARED\SYSTEM\APPS\SPREZZWARE\UTILITIES

When the SPREZZUTILS volume is attached, all files in that volume will have the SECURITY_MFS placed on them.

SPREZZATURA

There's one small gotcha with this though. Some of the system utilities do not like dealing with SYSPTR records with a volume MFS, so you may have troubles doing a LIST_VOLUMES on them. You should also not try to save this information to your DBT file, but attach the files during your login process.

This is a small price to pay for something so powerful, don't you agree?

OpenInsight XP – Carl Pates – Technical Director – Sprezzatura Group

When Windows XP was released it came with a slick new user interface built on top of an integral 'skinning' engine that implemented a new way of drawing windows controls. Dubbed "Visual Styles", this system allowed developers to define appearance of controls and forms from simple colours to textures and shapes. It was possible to manipulate each defined part of a control as well as each part of the non-client (frame and caption) area of a window.

Most of this was enabled via a new function library called uxTheme.dll, but there was also a major internal change to the way in which Windows implemented controls. Previously Windows supplied a series of 'standard' controls (Edit lines, list boxes, checkboxes etc.) direct from User32.DLL, and the so-called "Common Controls" (TreeViews, ListViews, Tabs and so on) from ComCtl32.dll.

With the advent of Windows XP Microsoft created a new version (Version 6) of ComCtl32.dll which handled all the processing for both types, but it also hooked into the services of uxTheme.dll so that all these controls would now adopt whatever visual style was chosen by the user. However, so that no applications were broken, the default behaviour in Windows XP was to use the old way: Visual Styles would not be used for an application unless it was explicitly set to do so.

Enabling Visual Styles in OI

Enabling Visual Styles in OI is actually quite simple, but we have to let Windows know that it's OK to do so, as OI itself currently does not implement them. We do this by creating an "Application Manifest" file, which is simply an XML document that describes an application and any components required to run it.

When an application creates forms and controls, Windows XP looks for a manifest file to see if it should use ComCtl32.dll version 6 to create them, or if it should use User32.DLL and ComCtl32.dll version 5 to draw the controls instead.

Normally these manifest files are compiled into resources and embedded directly into an application's executable file along with bitmaps, icons and suchlike. However, they may also be created as standalone files in the same directory as the application they apply to. All we need to do is adopt the appropriate naming convention which is simply "appname.exe.manifest".

SPREZZATURA

E.g. for OpenInsight we need to create a manifest file called "oinsight.exe.manifest" and place it in the same directory as "oinsight.exe".

The contents of the file itself are fairly straightforward:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">

  <assemblyIdentity
    version="1.0.0.0"
    processorArchitecture="X86"
    name="Revelation.OpenInsight.Sysprog"
    type="win32"
  />

  <description>Your application description here.</description>
  <dependency>
    <dependentAssembly>
      <assemblyIdentity
        type="win32"
        name="Microsoft.Windows.Common-Controls"
        version="6.0.0.0"
        processorArchitecture="X86"
        publicKeyToken="6595b64144ccf1df"
        language="*"
      />
    </dependentAssembly>
  </dependency>
</assembly>
```

So, just paste the XML above into Notepad and save it as "oinsight.exe.manifest" in your OpenInsight directory. The next time you launch OI you'll have Visual styles enabled!

Turning off visual style for a control

Of course, there are instances where you won't want a control to have visual style enabled. We regularly use a 4-pixel high group box to draw separator lines in a form, as you can see in the example below:

SPREZZATURA

Filter Selection

Select the type of filter you want to use, followed by the filter options and the destination for the results.

Filter Option: **SELECT statement**

Enter an R/List SELECT statement in the box below and click 'OK' to execute it. If you want help building your statement click the options button (...) at the side or press F2.

Statement:

Destination:

- ☐ Active Window
- ☒ Active List
- ☐ Saved List - Enter a list name here:

OK Cancel

Now take a look at it with XP Visual Styles enabled – it's almost completely invisible, except for the new-style rounded corners at each end:

SPREZZATURA

To stop this from happening we need to tell Windows to draw the group box in the normal way, and we can do this with the `SetWindowTheme` function exported from `uxTheme.dll`.

`SetWindowTheme` is defined in the Windows SDK as

```
HRESULT SetWindowTheme( HWND hwnd,
                        LPCWSTR pszSubAppName,
                        LPCWSTR pszSubIdList
                      );
```

Which means it takes 3 arguments:

hwnd	Handle to the window whose visual style information is to be changed.
pszSubAppName	Pointer to a string that contains the application name to use in place of the calling application's name.
pszSubIdList	Pointer to a string that contains a semicolon-separated list of class

SPREZZATURA

<p>identifier (CLSID) names to use in place of the actual list passed by the window's class.</p>
--

Well the last two seem pretty complex but we don't really need to worry about them in this instance. The important thing is that if we pass empty strings for them then Windows will turn off the control's visual styling.

So we need to prototype this function by creating a record called DLL_UXTHEME in the SYSPROCS table and add the following lines to it:

```
UXTHEME
INT STDCALL SetWindowTheme( HANDLE, LPWSTR, LPWSTR )
```

Next execute the following in the System Editor Exec line or the System Monitor:

```
RUN DECLARE_FCNS "DLL_UXTHEME"
```

Now to use in Basic+ we simply do this:

```
declare function SetWindowTheme, get_Property
hwnd = get_Property( @window : ".GRP_LINE", "HANDLE" )
call SetWindowTheme( hwnd, "", "" )
```

One other thing to note here: If you try this on a pre-Windows XP system (Like Windows 2000, Windows 98 etc) you'll hit the debugger. This is because uxTheme.dll does not exist on those systems, so you should check for its existence via the isLib() function before you try and use it.

E.g.

```
declare function SetWindowTheme, get_Property, isLib
if isLib( "UXTHEME.DLL" ) then
    hwnd = get_Property( @window : ".GRP_LINE", "HANDLE" )
    call SetWindowTheme( hwnd, "", "" )
end
```

Securing Menus via S/MS

The latest addition to the Sprezzatura utility fold is S/MS – Sprezzatura's add on menu security module for OpenInsight. This grew out of a custom project for two different clients. One who wanted to be able to attach a different menu to a window at create time and one who wished to be able to individually secure those menu items on a case by case basis. Thus was born S/MS.

SPREZZATURA

One of the few areas of lack of flexibility in the current version(s) of OpenInsight is in the treatment of Menus on MDI frames (and when it comes to it on any Window). Frequently the developer needs to customise the menu options that are available to a user based upon their security level and whilst simple modifications may be made using Set_Property ENABLE and VISIBLE this requires that place holders be in place for all potential menu choices which may not always be practical.

S/Menu Security (S/MS) provides a way around this. It allows you to decide upon window creation WHICH window to "copy" the menu from at creation time. Thus you could create a dummy window per "Security Group" with its own menu structure and assign a menu based upon the user's personal profile at runtime. In addition you can individually customise said menu by deciding on a menu option by menu option basis whether that option ought to be included for the current user. All of this is done automatically for you simply by making one call from your window create event.

All of this functionality is provided by a single subroutine called ZZ_Menu_Security.

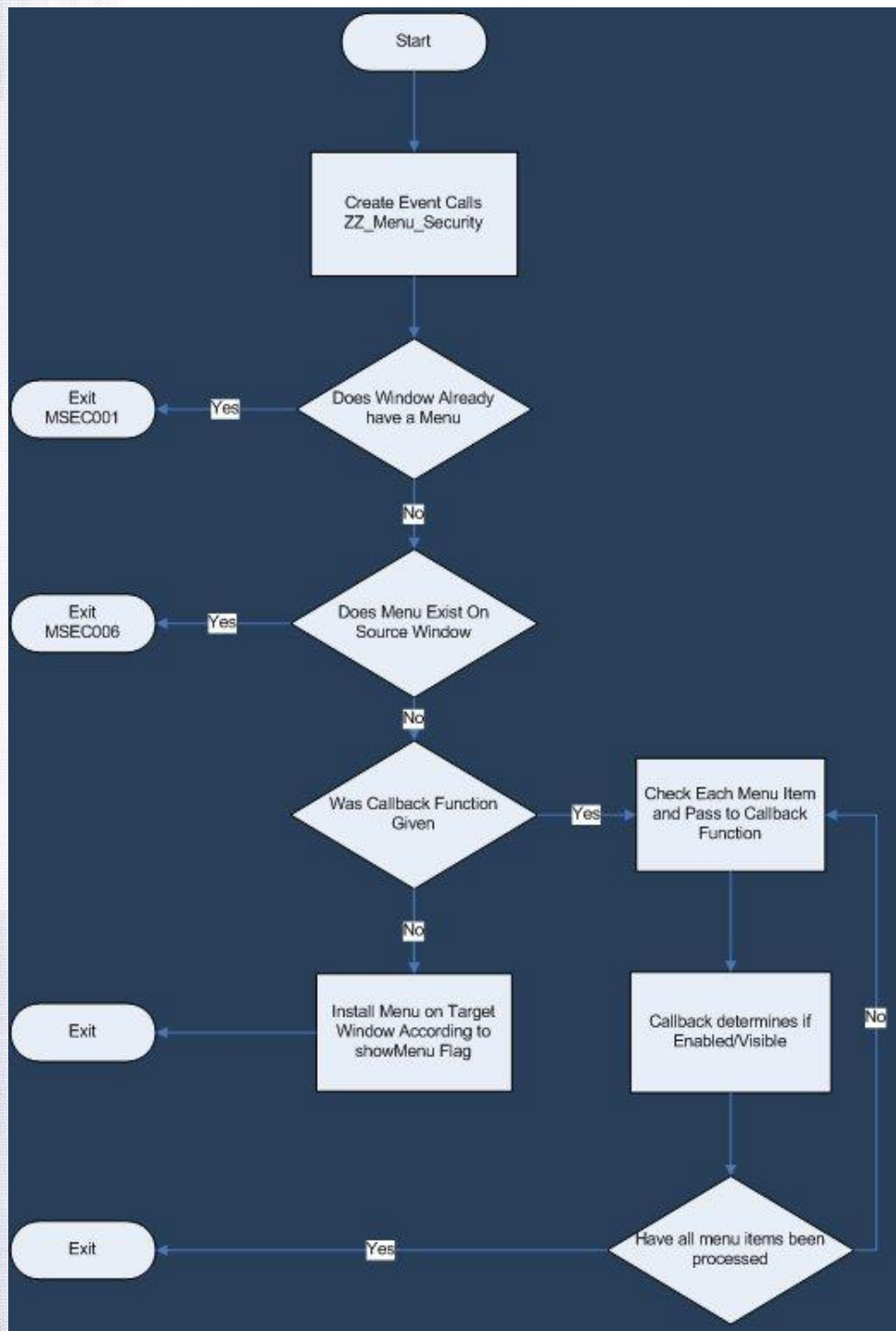
An important note about licensing

S/MS is licensed PER DEVELOPER copy of OpenInsight. Thus once you have legitimately acquired the software you may incorporate it into all of your delivered applications but you may not redistribute it to other developers or to different developer serial numbers.

It is perhaps easiest to describe the functionality of S/MS with a flow chart :-



SPREZZATURA



So as you can see the operation is incredibly simple, you simply place a call to the routine on the create event of the window (or on a promoted create event if appropriate) and take it from there.

S/MS is licensed on a "per developer copy of OpenInsight" basis and is priced at less than a day's consultancy. For further details please [contact us](#).

SPREZZATURA

Eric the Emu's AREV Trips

Recent discussions in the Revelation forum (well this has been delayed – Ed)

http://www.revelation.com/_85256DB80017688B.nsf/0/DAA27C2893F8210085256C02003A217A?Open from 26 July 2002, hail back to one of the emu's long-lost utilities.

There are some early versions of AREV and REVG which don't really provide a mechanism to reveal the last keystroke pressed. Once upon a time, possibly on an adrenaline-charged Monday, the enquiring emu was faced with several angry payroll officials who insisted they were saving data but that data was being lost by their AREV 1.12 payroll system. The emu set about logging the critical keystrokes of each user, looking for moments of sheer brilliance when the idiot users hit Escape instead of F9 in data entry windows.

The DOS keyboard buffer is circular, located at 0040:001E and there are two pointers in the BIOS in low memory 0040:001A & 0040:001C which point to the head and the tail of the buffer. The emu thought this was a fair place to look for evidence. In AREV you can use the following KEYTRAP function to determine the last key pressed. This works for all but the Shift / Ctrl and Alt keys.

The Escape key puts a value of `\011B\` (= 283 in hi-lo format) in the keyboard buffer. Check for a return value of 283 which will be the most recently added value in the buffer if the user escapes.

To create the routine \$KEYTRAP (which you should then catalog)

```
A = \020000000000000000000065253BA4000\  
A := \8EC226A11C003D1E007508268B0E3C00\  
A := \EB07908BF0268B4CFEBE00005B5A07B8\  
A := \0014BA00002EFF1E0A00CB\  
OPEN 'BP' TO BP.FILE THEN  
    WRITE A TO BP.FILE, '$KEYTRAP'  
END
```

```
* Here is the RBASIC to test it...
```

```
subroutine test.keytrap
declare function keytrap
call msg('This is a message',' ',' ',' ')
* now see what key the user pressed to exit the message
lastkey=keytrap()
if lastkey = 283 then
    call msg('you pressed escape to leave the message',' ',' ',' ')
end else
    call msg("you didn't escape from the message",',' ',' ',' ')
end
return
```


SPREZZATURA

Because it's so much fun (almost as much fun as being in Vegas), you can work out the other key scan codes for yourself. If you use a Microsoft ergonomic keyboard, with the keys split at funny angles, this activity can take up to seven weekends. You also tend to see everything as italics....eight becomes infinite, and so on....

As my fine old grandfather once said, as he slaved over the card reader in the Vax PDP-11 in his garage (used to catalogue his home-brewing beer inventory) "You can always tell the pioneers, they're the ones with the arrows in their backs..."

Celebrity Interview – Bill Caisley – Revelation EMEA Support

Who are you?

Bill Caisley

Who do you work for?

The wife and kids!

But seriously.... Revelation Software

What is your role in the RevSoft community?

Tech Support for Revelation UK. I am committed to gleaning as much knowledge as possible on the full Revelation product range.

How long have you been involved with RevSoft?

Since June 2003. With Multi Value systems since 1990.

Which products do you use?

Arev and OI the moment.

What do you most like about RevSoft products?

The simplicity and power. Also the way the products are evolving and keeping up with technology

What do you most dislike about RevSoft products?

What's to dislike?

SPREZZATURA

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

I guess I would follow the flock and opt for MS SQL. I can't wait for the fuzzy "Do what I'm thinking" release.

What are your favourite 3 books and why?

The Hobbit.

The Lord of the Rings - Looks like 4 books but really 2 stories. I can read them over and over; this is the way history should have been. The films brought the story to life for me but much better as a read.

Supernature – Lyle Watson's descriptive and informative explanation of the every-day in layman's language. It really appeals to my lateral thinking.

What are your favourite 3 CDs/Albums and why?

Pink Floyd - The Wall. They don't make them like this anymore.

The Beatles - white album. Defines a period in my life when I was young and foolish.

Chris Isaaks - Baja Sessions. Best "sippin" album for me.

What are your three favourite films and why?

The Shawshank Redemption - Right will prevail and overcome any adversary. Surprise ending. The good guy wins!

Die Hard - Action..action..action. The good guy wins big!

The Life of Brian – Monty Python's best work. It is the most intelligent comic farce I have ever seen. Consider the lilies... The good guy gets it in the end.

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

I was present at the release of Nelson Mandela and the democratisation of South Africa. That was big enough for me!

And finally Your motto/witty aphorism

Life is a gas... Let it out! (Me)

SPREZZATURA

Peripheral Trivia

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

- ☑ TV Stargate Atlantis
- ☑ Book: Joel On Software (Joel Spolsky)
- ☑ CD: The Streets – A Grand Don't Come For Free
- ☑ WEB: www.v-flyer.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

S/ENL - ©2010 Sprezzatura Ltd. All rights reserved. Portions copyright Microsoft Corporation Inc. Portions copyright Revelation Technologies, Inc. No portion of this journal (other than code segments) may be reproduced by any means, be it photocopied, digitised, transcribed, transmitted, reduced to any electronic medium or machine readable form, nor translated into any other language without the prior written consent of Sprezzatura Ltd or Sprezzatura, Inc. The moral rights of the authors have been asserted.

Disclaimer - Whilst every effort is made to ensure the accuracy of the information contained herein, neither Sprezzatura Ltd nor Sprezzatura Inc. can accept liability for the failure of anything documented herein to work nor for damage resulting from the application of methods/techniques learned herein.

TRADEMARK NOTICE

OpenInsight is a trademark of Revelation Technologies Inc. trading as Revelation Software. Microsoft, Windows™, and MS-DOS are registered trademarks of Microsoft Corporation. All other product names are trademarks or registered trademarks of their respective owners. Printed in the United Kingdom.

S/ENL Volume 4 Issue 2, 20th November 2004.

Please encourage your correspondents to send e-mail to admin@sprezzatura.com with SUBSCRIBE SENL in the subject line to get their own free subscription. Everyone is welcome! Tell your friends about S/ENL.