



WWW.DDUG.ORG

November newsletter



Welcome to the new

www.ddug.org newsletter. The DataFlex Developers' and Users' Group Inc. has most of its 35 members in Australia and New Zealand but welcomes all DataFlex users as members.

This newsletter is going to be published about once a month and distributed as word document to members of the DDUG. It is also available as an Acrobat Document on our website.

Each month 2 or 3 members are approached to prepare an article. This can be anything from a programmer's logbook, via listings, macro's packages, to Tip's & Tricks, 3rd party tools etc. Every writer for this newsletter is free in how & what to write as long as it is found interesting for its members by the newsletter editor.

This month we have three articles:

1. *'The Client who didn't know he needed SQL'* by Barry Munro
2. *'Version Control System, What, When, Where'* by Marco Kuipers
3. *'Add spellcheck to your VDF in 10 minutes'* by Marco Kuipers

I hope you all enjoy reading this newsletter, and it is a new start to a frequent and interesting information supply from you all to you all.

Marco Kuipers
Marco.Kuipers@nci.com.au

1. A WORD FROM THE PRESIDENT

November 10th sees the resumption of our WebConferences.

The choice of a Saturday means we can conveniently reach across the 5 time zones from Western Australia to New Zealand, that is the location for most of our members. A side consequence is we are also able to reach across the continental United States and Canada, it being 6:00pm Friday on their West Coast through to 9:00pm on the East Coast. In previous times DDUG, and its precursor VDUG, has had members from these countries.

The program has a 30-minute break between the two presentations to give attendees a chance to have their lunch, or something else to eat, depending upon their local time zone.

The committee, and Lucas Olde in particular, has spent a lot of time and energy in trying to use "free" voice over IP for our conferences. With the introduction of ADSL in the Australia telecom marketplace, this would seem to be the next step. So far our testing has found a "lag" being introduced by some devices somewhere in the WWW that prevents Voice Conference Links via broadband. So, in the meantime, we continue to use the "fee for service" of a Telstra Conference Call.

Make our day – contact you local Convenor and register you intention to attend the WebConference on November 10th.

Perth - Clive Richmond
(08) 9321 3378 clive@triumph.com.au
Adelaide- Merv Renton
(08) 8369 1344 merv@csolve.com.au
Melbourne - Chris Vawser
(03) 9741 7115 Syscob@AlphaLink.Com.Au
Sydney- Barry Munro
(02) 4272 5977 Mobile: 0403 443 688
Baz@peace-of-mind.com.au
Yeppoon - Lucas Olde
(07) 4939 3026 luke@cview.com.au

Neil Hamilton, 2001 President of DDUG



WWW.DDUG.ORG

November newsletter



1. The Client who didn't know he needed SQL.

“A developer's storey of a new gadget he found to play with and his client.”

Many notes are from actual communications, Invoice texts and diary entries.

The saga actually begins during September 2000.

Sept 16: On site in Newcastle, looking at yet another data corruption, client yells at developer in irritation, “. . . and can't you fix this once and for all . . .”.

Developer looks at system schema in his head, considers:

- 1) System has been first written by himself in 1988 and upgraded as the flavours of Dataflex have matured over all these years past.
- 2) System is currently in VDF 5.01 (b or c or ?) using WinQl as reporting medium.
- 3) Fifteen users (2 remote) use this during what they consider is a high pressure & productive day, the developer looks at their wide comfort zone and wonders at management, but shuts his mouth.
- 4) Data goes right back to '88, millions of records, with a few lost over time through unsuccessful re-indexes, hundreds of reports, both functional reports & management tools.
- 5) The reports crawl along, some take 2.5 hours to get to screen.
- 6) Developer sighs, puts hand in pocket & rubs back of neck, thinking of his other workloads. And says “ yeah, OK”

Sept 18:

Developer orders 16-port 100 Base-T switch, couple of print servers, pack of Intel Pro 100 cards, all the SD-Ram he can find, RAID 5 array cabinet, upgrade to SBS 4.5 with much altered ex-NT server and tape drive, Veritas BU Exec, Diskeeper 6.0 etc., and loses his weekend to upgrade. And fights with client over money, but finally gets agreement.

Sept 23:

Developer fights hard with old client boxes, replaces many old VGA cards etc., and fights even harder with client over money.

Developer installs SQL 7 as part of supplied software with SBS 4.5, fights with client over internet & fax connections, all the while with a half-formed idea of using DAC's new SQL C-Kit down the line.

During all this, client suddenly blurts out “what about that rental deal you do with others?” A rental deal is struck, an SLA is drawn up, and suddenly developer realises he owns the whole old heap he has been working on.

And the bill for it.

He sighs as he knows the next set of car tyres are quite a ways off.

But this is 'future' building, right?

Sept 25:

Developer now back in his cave regarding the software system he has to fix.



WWW.DDUG.ORG

November newsletter



He buys (finally) VDF 7 and starts the Keating gravel-scratching J-curve of learning and plugging of code scraps from where ever.

Oct 18:

The changes developer had made to speed up system have resulted in not much support activity from Newcastle, and he wonders if he may get away with what he has done, just to hardware and OS upgrades.

Also if he paid the last phone bill.

Too quiet, so he rings.

The GM says “Hey, glad you called, we want to change the whole invoice structure into a project base style, where we can bill monthly, take prepayments, when can you take instruction?”

Oct 20:

The developer hates travelling Fridays, but he dutifully arrives on site up in Newcastle. And gets lots more work to do. It’s what he does, OK?

Oct 21:

Back in the developer’s cave.

Developer nervously kicks the SQL Connectivity Kit into action.

He is surprised at ease of conversion. Only lost 30,000 records with bad dates. SQL server has a problem with bad dates. Not bad for 6.5 million records. Damned date masks! Nothing in Dataflex has prepared him for this, so he goes to Angus & Robinson to buy books on SQL, putting off a tussle with the new Crystal stuff for a while.

Nov 13:

Developer has new reading prescription filled, his wife and cat both bite him now. He goes to sleep reading text books, and is generally just not nice to be near. He also has other people’s work to do, these people shout at him a lot.

Nov 27:

Developer is at last able to write SQL scripts to repair bad date data, and sets these aside for the Big Conversion Day (well, weekend – you bet). He scraps with int files and the queer way indexes get out of order when he adds more fields with more indexes. He downloads and looks at the way various fix-it software works, like Stsynchroniser, and others. No, it looks like using the DbBuilder’s inbuilt stuff has to be the way. With lotsa scripts to alter data once it arrives in the SQL database. He starts on the changes the client wants.

Dec 23:

Developer has finished the project module, loads car with box & projector, heads off to Newcastle. Throws new stuff against wall for client and his GM to review. They argue into the evening, until they go off to get drunk somewhere in town. Developer happily never relies on memory, and has copious notes on hundreds of new required reports.

Jan 7:

Developer is holidaying near Newcastle, cursing his rotten luck, the client has found him, wants more changes, and won’t buy a beer. He takes more notes.

Jan 16:

Developer gets ill, doesn’t work again till March.



WWW.DDUG.ORG

November newsletter



March 18:

Developer surfaces and is tangled up with some user group doing something up North, is not much use to anybody yet.

He opens up Crystal Reports 8, and, reading the book(s), creates an ODBC gadget on his desktop. He is surprised and happy how the old WinQl reports convert over without too much hassle. He makes sub reports and plays with pushing requests down to the server until the wee hours for many nights. Finally his mate tells him of reports not showing correct data in a few reports done for in-house software. Investigation reveals problems with various installed ODBC dll's in various boxes. He chases around web sites and sees it as a major problem, just waiting to bite on deployment.

April 8:

Developer has discovered OLEDB. Joy. The reports can be stored away from data, just about anywhere the client box can get at. He pounds away at keyboard much, much more.

April 23:

Developer has at last hit on a deployment schema. He will install the VDF 7 app on each user machine, with just enough support files to get it running, under their Program Files folder. This keeps reg entries easy to maintain.

He will use Robocopy to synch each box as it logs on, as he is sure as the Lawd made pretty girls, there will be changes.

He will store just the one copy of all the reports on a shared resource on the server.

He does the same with Word dot templates needed by the VDF 7 app.

The pictures the client needs, the developer has discovered, can be fired up in a user's browser by a call from the VDF 7 app, these can be stored in just the one place too.

Pathing to these damned reports and images and templates is stored in the system file, using UNC conventions.

Now the VDF 7 application, on each user machine, when goosed into action, comes alive looking for the server, (it had better find it too!) using trusted connections.

The remote sites connecting to the server sort of pushed this thinking along, and allows abbreviated versions of the app to be deployed 'out there'.

Mixed mode network libraries are slow, so he removes all but TCP/IP connections, leaving the ODBC port at the standard 1433.

The client, you see, has discovered Excel late in life, and clearly intends to use it instead of his expensive developer for quite a few reports.

When he regards how Word mail merges, he becomes ecstatic and buys the developer more beer.

May 18:

Developer is as ready as he ever will be to do this conversion.

The other clients waiting for him to work on their stuff provide good encouragement. The developer is frail and bruises easily, he sighs heavily as he packs the kitchen sink in his car and drives off to Newcastle.

Same day – in the evening on site:

One of the SCSI HDD's has failed over to the hot spare, it's rebuild still in progress.

Developer hadn't noticed it in the fuss of setting up his box(s) and spare junk in a nearby



WWW.DDUG.ORG November newsletter



office. Nothing to do but press on, he hooks up UTP cable to the switch he installed so long ago. And starts conversion.

May 20:

The conversion is finished, everything runs, a knee in the groin gets some recalcitrant client boxes going properly. Happily, the newer screens don't differ too much from the outgoing system, the dying swans of the office have subsided and even remember the developer at lunchtimes.

May 21:

Everything stops. One of the brighter staff has found the Sircam virus, she just won't rest till she tells everybody about, by email.

Developer finds a local computer place, buys Norton AV for the server, more for the users, installs AV on all of them, even the remote sites. The client only comes with developer for one of these remote sites, he has found ***something*** the staff have done wrong and is shouting at everybody.

May 22:

Developer is worried. Sometimes the app locks up, won't proceed till a few users back out of theirs, and then rejoin.

He lets DAC NL know, and is rewarded with words like, un-reproducible, this is working here OK, and so forth.

He haunts MS SQL newsgroups till wee hours, looking at transaction locking things, and noting replies telling him, he is programming poorly.

No surprises to the developer, he knew that all along.

May 23:

Ben in NL relents, sends developer new MS SQL dll.

He applies it. It works.

It also causes the creation of funny cache files with a cch extension.

These, he is told, enable the app to get loaded and going, faster.

It does too.

Locking problem goes away.

New SCSI HDD arrives by courier, push it into slot, way it goes all by itself.

Developer can't find the client this evening, so drinks Jamesons, alone.

Ponders futility of all human endeavour.

May 24:

Reports. More reports.

They are coming out of the developer's ears.

He puffs away, sure he is learning in the best way possible: getting paid for it.

CR 8 against an SQL backend is surely the quickest reporting there ever was.

This is a huge database, many tables, with many files. Crystal just handles it.

May 25:

This leg of this job is finished.

The developer packs up, heads further North, up near Taree.

He is not on this job too long, when Newcastle rings in a flap.

Two users cannot join the system.



WWW.DDUG.ORG
November newsletter



He packs again, drives down through holiday traffic to Newcastle.
And finds not one boss about anywhere, they have all left for parts unknown, it seems.
The problem went away when the users entered the correct passwords.
He writes up a bill for this, and heads back up to near Taree.

And Taree is another storey

Barry Munro
Baz@peace-of-mind.com.au
Piece-Of-Mind Software
<http://www.peace-of-mind.com.au/>
Wollongong NSW



2. Version Control System, What, When, Where

Every programmer knows there is something called version control (or revision control). Lots of programmers think they can have control over version without buying and using a VCS (Version Control System). For every programmer that starts working with VCS (because the employer or principal requires to) after some dark clouds the sun shines beautifully.

Misunderstandings about VCS:

1. It takes so much more time to do, especially for small fixes
2. It's difficult and cost lost of time to get used to
3. The way how I do version control myself is just as good
4. VCS is expensive

The basics:

A system knows about your precious files. It stores the delta's (differences) between the versions together with the human explanation of the version change. The system requires you to tell the system you are going to work on a file (Check-out) and give it back when finished (Check-in). The attribute Read-Only is used to let the user know while working in the IDE or Multi-Edit (or any other program) that the file is not yet Checked-out.

VCS Projects

A project in VCS is often a system in real life. A project contains files 'controlled' by the VCS and other files (work files). To place a file in VCS, you choose 'Add'; often you can add multiple files in one go by using a filter (eg *.SRC). A description of the file (if you do one) or files (if you do by folder or by filter) can be entered.

After adding the files to the VCS, the attribute is set to Read_Only. This is done for you to help reminding you the file is in VCS.

If you need to modify the file, you select it in the VCS and select Check-Out. Some VCS's ask for a check-out description, you can enter the job number or customer description of the problem here. All VCS's will undo the read_only attribute and mark the file in VCS as Locked by <username>.

After making your changes (compiled and programmer-tested) you check-in the program with the description of the changes.

Advantages:

1. You can always see what lines are modified between the versions
2. Version numbers are maintained automatically
3. When one programmer is modifying a component, another one can't work on the same version (you can by creating a branch but that later).



WWW.DDUG.ORG

November newsletter



Keywords

A cool thing is keywords. These 'variables' in your source are maintained each time you check-in or check-out your source. You can have a keyword \$Version\$ that is displayed on the helpscreen. You don't have to maintain this anymore, it's done for you. The same as revision history. You can insert a keyword \$History\$ in your source, and the full version history is inserted at this point. Of course there are many keywords and some VCS's have more than others.

Get

Sometimes you just want fresh a copy of a file without checking-out. This could be because you have a full copy of the source code in your C: drive for compiling, and one day you want all the new versions of components your colleagues have checked-in.

Undo-Checkout

This one does not need much words. Just unlocks the file and places the last checked-in source back in the working directory.

Lock

This is one of the actions done by Check-out. The check-out does a lock and a write of the most recent file in your working directory. The lock leaves the file alone, but locks the file as-if it is checked-out.

Label.

This one is very interesting. When modifying source, all the components have each a version number. But together they might also have a version number. Eg. the order entry view is version 1.21 and the customer entry view is 1.14. Together they make 'MyOrderingProgram v1.00'. This can be done by labelling all the components as 'Version 1.00'. Later, the order entry view is modified to 1.25 and the customer entry view stays the same. Version 1.01 is released, so the components get labelled 'Version 1.01'. At a later time, if changes on 1.00 are needed, you can Get or Check-out all modules labelled as 'Version 1.00'. The VCS knows the version numbers of the components and will recreate a set exactly the same as when you labelled the source. This is very powerful for releases.

Different version control systems:

Quma VCS: <http://www.qumasoft.com>

FreeVCS: <http://www.freevcs.de>

Visual SourceSafe: <http://msdn.microsoft.com/ssafe>



3. Add spellcheck to your VDF in 10 minutes

The power of ActiveX really shows when things like spell check are needed in your application. I was puzzled myself when I found, implemented, tested and worked with a spell check control in my VDF application within 10 minutes. Please look at the following steps:

1. FIND THE CONTROL

Ofcourse there are many places, but my favourite is www.ComponentSource.com Here I found a SpellText control made by Chado. Their own site is www.spelleditor.com

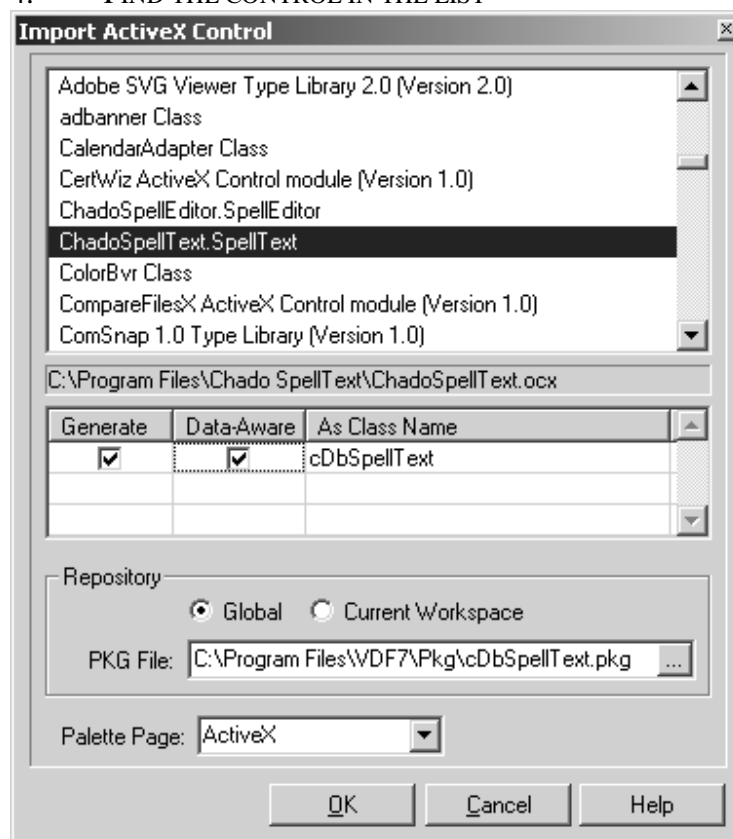
2. DOWNLOAD THE CONTROL

Just click on the link, install and all done

3. IMPORT THE ACTIVE X CONTROL IN VDF

Start the IDE, choose, Tools, Import ActiveX control BEFORE opening your program.

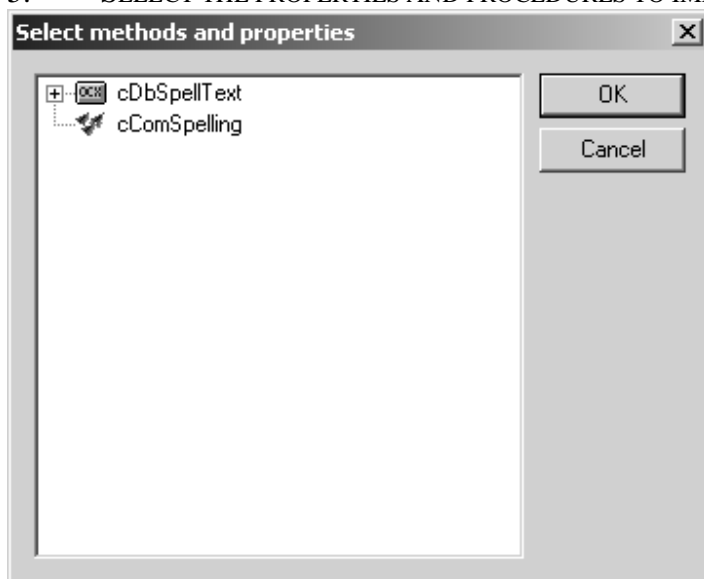
4. FIND THE CONTROL IN THE LIST



Choose in this case the ChadoSpellText.SpellText. Tick the Data-Aware checkbox so we can use this for entering our data. Then press OK.



5. SELECT THE PROPERTIES AND PROCEDURES TO IMPORT



In a separate dialog the properties, procedures and functions are shown that will be generated when pressing OK. You could decide not to support all features, so disable some functions etc. I would not recommend this unless you really know what you are doing. Just press OK.

A couple of progress bars will show the ActiveX gets read and the package gets generated

6. MODIFY THE CLASS TO MAKE THIS CONTROL DATA-AWARE.

This part could be tricky because of unfamiliar data types but in this case it couldn't be any simpler. Basically VDF can't know the following things automatically:

- How to set the value of the control
- How to get the value of the control
- What the event name is send after every change.

Therefore a couple of code changes must be made.

The full generated code is displayed with a lot of thing not many programmers haven't seen before but no panic, just got to the 'TODO:' bit via a find.

That should bring you to this code:

```
// This event is called when the DDOs need to update a control's
// value. You must add the code that passes this value to the OCX value.
Procedure OnRefreshDbOcxValue String sVal
    OLEVariant vValue
    // {ToDo: Set value to proper variant type}
    PutOLEVariantValue sVal as {OLE_VT_????} to vValue
    // {ToDo: Add code to set the OCX value}
    Set {set-ocx-value} to vValue
End_Procedure
```



WWW.DDUG.ORG

November newsletter



```
// This event is called when the OCX needs to notify the DEO that its
// value has changed. The name of this event will be different for each
// control. Look for some kind of "post-update event" message.
// to update the OCX value.
// {ToDo: Rename this procedure}
Procedure {post-update-ocx-event}
    OLEvariant vValue
    String sValue
    // {ToDo: Add code to get the OCX value}
    Get {set-ocx-value} to vValue
    GetOLEVariantValue of vValue to sValue
    Send UpdateDbOCXValue sValue
End_Procedure
```

Just modify this to:

```
// This event is called when the DDOs need to update a control's
// value. You must add the code that passes this value to the OCX value.
Procedure OnRefreshDbOcxValue String sVal
    Set ComText to sVal
End_Procedure

// This event is called when the OCX needs to notify the DEO that its
// value has changed. The name of this event will be different for each
// control. Look for some kind of "post-update event" message.
// to update the OCX value.
// {ToDo: Rename this procedure}
Procedure OnComChange
    String sValue
    Get ComValue to sValue
    Send UpdateDbOCXValue sValue
End_Procedure
```

Save the package and the heavy work is done.

7. ADD THE SPELLCHECK CONTROL TO YOUR APPLICATION.

Open your program and open your view.

Delete your current textbox, and place the new one that should be on the ActiveX tab on your toolbox.

Set the Database properties to your field, and set the label and label location.

Save the view

8. TELL THE DATA DICTIONARY YOU WANT A DIRECT ACCESS TO THE TEXT FIELD

Direct moves of string values from and to a DataBase textfield are only supported when defined in the data dictionary so. Add the following line to the DD object of the textfield:

```
Send DefineAllExtendedFields
```

9. COMPILE AND TEST

You should now have the joy of being confronted with your spelling errors...



WWW.DDUG.ORG

November newsletter



Description

Overdue policies and branches.

The policy report for the declaring Overdues, must be constrained to only show policies with the POLICY.OD_DEC_FLAG set to 'Y'.

On the overdue report entry page, preview page and xpg report, the Branch selection / display must be suppressed if the POLICY.BRANCH_O_DUES is set to 'N'.

10. BE HAPPY AND REGISTER THE CONTROL.

The control pricing is a bit odd:

- Life time free upgrades and support
- Unlimited developers licence

A one-application costs US\$ 135

An enterprise edition costs US\$ 315