



# WWW.DDUG.ORG

## December newsletter



**Welcome** to the second [www.ddug.org](http://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.

Thanks to our members I've been able to get this newsletter published again. Only one week behind schedule. This newsletter is distributed as word document to members of the DDUG. It is also available as an Acrobat Document on our website.

The intention of approaching each month 2 or 3 members to prepare an article was again successful. The articles can be anything from a programmer's logbook, via listings, macro's packages, to Tip's & Tricks, 3<sup>rd</sup> 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 five articles:

1. *'Codes do we need them?'*  
by Peter Brooks
2. *'A 'Little' System Integration Job!'*  
by Peter Tawse
3. *'Debugging in a Multiuser Environment'*  
by Clive Richmond
4. *'XML you want it, even if you say you don't'* by Marco Kuipers
5. *'The Loneliness of the Long-distance Programmer'* By Neil Hamilton

I hope you all enjoy reading this newsletter, the track is build and the train is rolling. All we need is the fuel.

IT IS TIME TO THINK ABOUT THE...

## **CORROBOREE 2002**

The DDUG committee would like to inform you the meeting is promised to be the best one ever...

**When:**

Thursday 7 March 7pm – 10pm; Welcome & cocktail party  
Friday 8 March; 8am – 17pm Corroboree  
Saturday 9 March; 8am – 17pm Corroboree  
Sunday 10 March; 8am – 17pm Corroboree

**Where:**

All Seasons Premier Menzies Sydney  
A four-and-a-half star hotel in the CBD

**Costs:**

Early bird conference price A\$ 299,-- (till 31/01/2002).  
Normal price A\$ 349.

Only for members of the DDUG. Non members will have to join DDUG to attend the Corroboree. For new member attendees, the DDUG joining fee will be waived.

**Program:**

The program is still under construction, and will be until the last moment. Keep your eyes on our website though, because some very exciting announcements will be made there shortly.

**Accommodation:**

Members are encouraged to stay at the venue. We have a special deal for the rooms, to be discounted from A\$ 195 per night to A\$ 119 per night per room. This is accomplished via a booking office. The A\$ 119 per room is frozen till Friday 14 December. After this date, a normal discounted rate will be applicable.

Please contact direct:

Sheryl Hookham, Traveller's Services  
Tel: 02.9654.3420 Fax: 02.9654.3422  
Email: [itla@Optusnet.com.au](mailto:itla@Optusnet.com.au)

And mention the DDUG Corroboree rate.

**Partners:**

On request of members during last Corroboree, we will encourage members to bring their partner. We anticipate on 50% of the partners attending, and an alternative program will be organised for them.

**What to do now?**

- Put it in your agenda
- Talk with your partner
- Create a budget
- Reserve a seat ASAP
- Book the hotel room ASAP
- Keep an eye on [www.ddug.org](http://www.ddug.org)

**More info:**

During 12 December Webcast...

Your Corroboree Organisers,

**Chris Vawser & Marco Kuipers**



# WWW.DDUG.ORG

## December newsletter



### **1. Codes do we need them?**

*Peter Brooks*

All too often when using database software we are confronted with a code for this and a code for that. Why do we need codes? What is their purpose? Traditionally codes were always used to save space – space on the hard drive and real estate on the screen. These issues are no longer of importance.

Have you ever wondered when entering in a code and description for a parent file why the code is required? Most of us do it because that is what we do. I get really annoyed at, for example, searching through a Customer list only to find I cannot search by the customer name. Which programmer was too stingy to allocate an index on the customer name?

I avoid codes like the plague. If they are not necessary don't have them. Sure I here you say "there a plenty of times they are required", that may be true. Space to display the information and perhaps consistency with older programs and their users come to mind. To demonstrate my point I have contributed a full working VDF7-IDE compliant application.

This contribution I hope will inspire other developers to contribute a full working version of a software product. We need more source code examples out there to encourage new developers and share new ideas. This application is a Mini Project Management system. Read the Readme.txt file for more information.

Peter Brooks

[www.membershipadmin.com](http://www.membershipadmin.com)



## WWW.DDUG.ORG December newsletter



### **2. A 'Little' System Integration Job !**

*Peter Tawse*

Some 4 months ago a long-time Client requested that I switch my Call-Centre System that they were using so that – instead of being integrated with an old CBA Accounting system written in DF2.3 (and subsequently ‘promoted’ to DF3.1c) – it be integrated to use the latest release of Triumph’s Business System (from Perth, Western Australia), written in VDF7.

At the same time it was proposed to upgrade the existing ‘Call-Centre’ app. from Vdf6 to Vdf7 SP3.

Initial analysis confirmed that this could be done, but that considerable changes would be required to switch from a ‘Batch-Orientated’ system to one that is ‘Transaction-Date’ orientated.

Before starting the system-changes, the existing application was migrated from Vdf6 to Vdf7 (still using the old files), so that any problems associated with the Vdf version upgrade could be eliminated from the equation.

I was delighted to find that only very minimal changes were needed – although some custom Tab-Handling code that was working under Vdf6 ‘spat the dummy’ and refused to function under Vdf7. No big deal, we had an alternative means of navigation between Tabs using Function-keys F11 and F12 anyway, so decided to do without the Alt-# tab-switching. (The system contains 2 dbTabDialogViews positioned one above the other, and the standard DAW F6 “Switch” key doesn’t rotate/switch between them).

#### **What was involved, and the problems encountered!**

Once the Vdf7 app. upgrade was running, a number of datafiles had to be changed right throughout the existing ‘Call-Centre’ code, and of course the associated field-names also. A very big job, especially when I’m not familiar with an editor that provides ‘Global Changes’ capability – must learn Multi-Edit properly REAL SOON NOW !

In most cases the field-length’s and types were fine, but in respect of one file the use of Sture Anderson’s ‘IntToStrR’ function was required, as we use a numeric Delivery-Address Number and Triumph have set this up as an ASCII field. No big deal, problem solved, and lists now display nicely in the correct collating sequence. Thanks Sture – I’ve found those utilities of yours are most useful.



## WWW.DDUG.ORG December newsletter



All file-relationship's had to be checked, and in some cases the 'Relates' that were standard within the Triumph system had to be turned off or amended within my own DD's. This was a difficult area, as – whilst I did have ready and welcome guidance from the Triumph boys in relation to implementing their necessary 'Business Rules' - I could not obtain any example .DD's from them.

To cope with the extra functionality required by my 'Call-Centre System', additional fields were also added to the end of the Triumph Debtor's Master and Delivery-Address files – after adding an "End\_Of\_Triumph" marker field !

Dataflex is GREAT in this respect – just add the extra fields you need to the end of the existing datafiles. All the existing Triumph Vdf7 programs continued to function in the normal manner – no problems at all !

More testing / recompile / testing cycles ....

Now, add that 'Popup' list that they want – the one that's activated from a 'Starzen Shortcut Bar' - showing a **large** list of 'All Pending Jobs'. It allows unallocated jobs to be allocated (in real live time) to the Technician who's somewhere out in the field, and - having allocated it – send that Technician an SMS message containing all the 'Job-Specifications' to his mobile phone. Beats 'Pigeon-Post' hands-down !

More testing / recompile / testing cycles .... Hey – it's all working. Nearly done.

Now, add in my Security System 'pluggin' (only a little more editing to 'Use' some PKG's, and amend a few Object-Types), recompile and retest. Beautiful – everything's working fine. Wait a mo, the phone's ringing !

5 minutes later: Well that was the site in question, and – having got the latest 'Test' system from me last night – would like to know 'Is there any way I can add .....'

It's a never-ending-cycle, isn't it. And haven't we all been here before. Of course, the answer to their query (after a little head-scratching) was 'Yes, I can do that'.

It's an easy one this time, but isn't that what I'm paid for ?

### The results:

My Client now has a new Vdf7-based Call-Centre Job-Allocation and Tracking system – fully integrated with an 'off-the-shelf' Vdf7 Accounting system, and –  
IT'S ALL WORKING AS PLANNED AND REQUIRED.



WWW.DDUG.ORG  
December newsletter



I'm even getting unsolicited comments back from the Client's staff to say how nice and 'User-friendly' the new system is. Nice !

So, what started out as a 'little' job (somewhere around 120-150 hours work) kinda grew, and grew, and GREW. Not that the specifications changed much (well they did, but only in a minor evolutionary way), but major items kept being added to the list to provide extra facilities.

So, some 400 hours later I have a very happy Client, with a new system that works, and that their Staff find easy and intuitive to use, - and I'm well-paid as well.

Moral of the story: From little 'things', BIG things can grow.

On that note, I think I'm off to bed. Chaio !



### 3. Debugging in a Multiuser Environment

*Clive Richmond*

The art of writing code in Visual DataFlex has become much easier since the advent of the Integrated Development Environment. You can simply drag and drop controls, change properties, compile and test using a few clicks of a mouse. However sometimes things you 'think' should work when running your program just don't. Bugs!

Like ourselves many of you who have evolved from procedural DataFlex would diligently trace the source of the problem with a sprinkling of SHOWLN commands. And some of you may have even ventured further and used the DEBUG command.

Since version 5 of Visual DataFlex a debugger, vdfbug, has come as standard. It is extremely powerful and allows you to interrogate all sorts of information, including file buffers, global and local variables, properties, the function/procedure call stack and even trace the OOPs messages as they fire.

Having said that, using the debugger can be tedious. You must make sure that any precompiled packages and the program you are testing have been compiled with the **Add debug info** option set. Also its slow to run, especially if you are based on a network or the program is particularly large. Inevitably you're going to slip back and add a SHOWLN statement here and there. It's quick-n-easy and you'll leave the debugger to more challenging problems.

#### **The problem with SHOWLN.**

Unfortunately SHOWLN does present some obstacles when developing in a team. If someone is solving a problem in a package that I include, I don't really want to see their debug messages. Likewise they probably don't wish to see my debug messages either.

But worst still is forgetting to remove those statements before a production version of your application is shipped out the door, followed by the support call 'This output window. Is it new? How do I turn it off?'. Red-faced and embarrassed ? Yes!

#### **Criteria for our solution.**

To prevent the outcomes described above we require a command that addresses the following problems.



# WWW.DDUG.ORG

## December newsletter



1. It must be able to distinguish between developers.
  - 1 Its default behaviour is passive and developer must enable it to display their debug messages. Being passive means our production version wont display debug messages when left in the code.
  - 2 The command must be simple enough so developers will actually use it. Too complicated and we're back to where we started.

### **Implementation.**

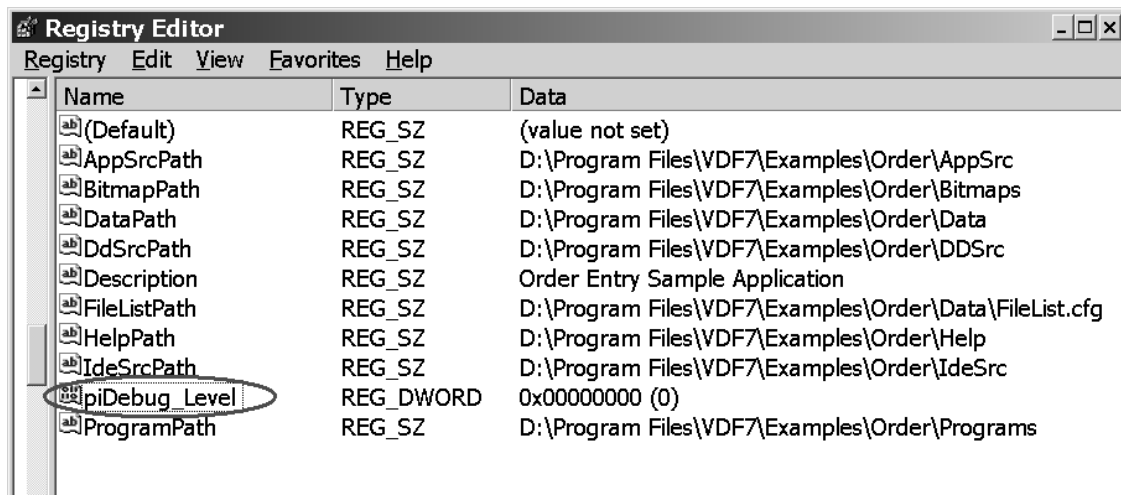
The implementation, as it turns out, is reasonably straight forward. We create a new set of block commands or macros, `DEBUG_BEGIN` and `DEBUG_END`, that represent the start and finish of where our debug statements can be placed. The `DEBUG_BEGIN` is provided with an optional argument to determine either the developer or a specific area of functionality e.g. reports.

```
DEBUG_BEGIN C_DEBUG_CLIVE  
    SHOWLN "nQty: " nQty ", nAmt: " nAmt  
DEBUG_END
```

To complete point 1 of our criteria we subclass the workspace and define a new property called `piDebug_Level`. This property is then used in the `DEBUG_BEGIN` macro to determine if the debug block should be processed. If the value is 0 the block is skipped while an ordinal number that matches the property is executed. So in an instance of our new workspace setting the property's value to `C_DEBUG_CLIVE` would only display those debug statements belonging to myself and no one else.

```
Object ProgramWorkspace is a MyWorkspace  
    Set WorkspaceName to CURRENT$WORKSPACE  
    Set ModuleName    to 'Order'  
    Set HelpName      to 'OrdEntry.hlp'  
    Set piDebug_Level to C_DEBUG_CLIVE  
End_Object
```

Although the default value for `piDebug_Level` is 0, if a level is set, as in the example above, then our production version would no longer be passive. To avoid this and fulfill the criteria of point 2, we define the property in our workspace as a `DWORD` value. It also means we no longer require the statement in the instance of our workspace.



The final criteria is easily catered for via the IDE's code completion function. Each developer can customise the CodeCompleter.kwd file and add the new command with their own argument. The file is located in the folder C:\Program Files\VDF7\Usr\Global\IdeSrc. When a developer wants to include a debug block all they need do is type *debu* and press Ctrl+Space. The IDE completes the command and marks the block between the parenthesis ready for editing.

```
Debug_Begin C_DEBUG_CLIVE
    Showln {param1 ... paramN}
Debug_End
```

## Conclusion.

The DEBUG\_BEGIN command offers a quick and simple way to include debug statements in your application for either a specific developer or area of functionality. It may be used to trace bugs or provide diagnostic information while on site. As debug statements are now formalised it is easy to profile your source and determine where your developers are having coding difficulties but also remove or comment out these commands using a stream editor such as sed.

Clive Richmond  
5<sup>th</sup> December 2001  
[ddug@triumph.com.au](mailto:ddug@triumph.com.au) <<mailto:ddug@triumph.com.au>>  
<<http://www.triumph.com.au>>





# WWW.DDUG.ORG

## December newsletter



### The Source.

#### Macro Command.

```
#COMMAND DEBUG_BEGIN
Register_Function piDebug_Level Returns Integer

#IF !0>0
  #IFSAME !1 C_DEBUG_JOE
  If (piDebug_Level(ghoWorkSpace) = C_DEBUG_ALL OR;
      piDebug_Level(ghoWorkSpace) = C_DEBUG_JOE) Begin
    #ELSE
      #IFSAME !1 C_DEBUG_CLIVE
      If (piDebug_Level(ghoWorkSpace) = C_DEBUG_ALL OR;
          piDebug_Level(ghoWorkSpace) = C_DEBUG_CLIVE) Begin
        #ELSE
          #IFSAME !1 C_DEBUG_FRED
          If (piDebug_Level(ghoWorkSpace) = C_DEBUG_ALL OR;
              piDebug_Level(ghoWorkSpace) = C_DEBUG_FRED) Begin
            #ELSE
              #IFSAME !1 C_DEBUG_REPORTS
              If (piDebug_Level(ghoWorkSpace) = C_DEBUG_ALL OR;
                  piDebug_Level(ghoWorkSpace) = C_DEBUG_REPORTS) Begin
                #ELSE
                  #IFSAME !1 C_DEBUG_HELP
                  If (piDebug_Level(ghoWorkSpace) = C_DEBUG_ALL OR;
                      piDebug_Level(ghoWorkSpace) = C_DEBUG_HELP) Begin
                    #ELSE
                      #ERROR 105 INVALID PARAMETER !1
                    #ENDIF
                  #ENDIF
                #ENDIF
              #ENDIF
            #ENDIF
          #ENDIF
        #ENDIF
      #ENDIF
    #ELSE
      If (piDebug_Level(ghoWorkSpace) = C_DEBUG_ALL) Begin
        #ENDIF
      #ENDIF
    #ENDIF
  #ENDIF
#ELSE
  If (piDebug_Level(ghoWorkSpace) = C_DEBUG_ALL) Begin
    #ENDIF
  #ENDIF
#ENDCOMMAND

#COMMAND DEBUG_END
End // DEBUG_BEGIN
#ENDCOMMAND
```

```
Enum_List
  Define C_DEBUG_NONE // 0 = No debug
  Define C_DEBUG_ALL // 1 = Any debug
  Define C_DEBUG_JOE // 2 = Joe's
  Define C_DEBUG_APC // 3 = Clive's
```



# WWW.DDUG.ORG

## December newsletter



```
Define C_DEBUG_FRED      // 4 = Fred's
Define C_DEBUG_REPORTS  // 5 = Debugging reports
Define C_DEBUG_HELP     // 6 = Help Interface
End_Enum_List
```

### Workspace subclass.

```
Class MyWorkspace is a Workspace

  // Construct_Object:
  // Declare public property to hold the debug level
  //
  Procedure Construct_Object
    Forward Send Construct_Object
    Property Integer piDebug_Level      Public 0
  End_Procedure //Construct_Object

  // DoAssignWorkspaceData:
  // Augmented to retrieve the debug level.
  //
  Procedure DoAssignWorkspaceData String sWorkSpaceTag
    Integer iDebugLevel

    Forward Send DoAssignWorkspaceData sWorkSpaceTag
    Move (WS_REGBASENAME+'\' +sWorkSpaceTag) To sWorkSpaceTag
    Get Profile_DWord sWorkSpaceTag "piDebug_Level" To iDebugLevel
    If (iDebugLevel <> REG_VALUE_NOT_EXIST) Set piDebug_Level To
iDebugLevel
  End_Procedure // DoAssignWorkspaceData

End_Class // MyWorkspace
```

### CodeComplete.kwd changes.

```
<Keyword=Debug_Begin/>
<Code=DEBUG_BEGIN C_DEBUG_CLIVE/n/tShowln {param1 ... paramN}
/nDEBUG_END/>
```



# WWW.DDUG.ORG

## December newsletter



### 4. XML you want it, even if you say you don't

*Marco Kuipers*

You can't read an IT magazine, paper or website anymore without seeing the frightening letter combination X.M.L. Frightening because if it is getting known by the customer, he might ask you what it is!!! Offcourse as a real developer, you've been too busy with fixing problems rather than have the time to look around...

I'm probably one of the thousands of people that tries to explain XML in a simple way using a limited amount of words.

*Marco statement 1: What CSV was 10 years ago is XML now.*

XML is the name of a document containing 'structured data'. In a CSV file, you could place a table, with field names. A mail-merge can read the data and spit out documents. Very helpful and very handy.

Better would it be if not one table but also child records could be placed in the 'CSV' file. Traditionally you solved this by adding extra columns, and forget about normalisation. This is acceptable and workable if not too complex. But now we have something better:

In XML you can define data the way you like. So I can make the following XML document:

```
<DDUG_Newsletters>
  <November_2001>
    <Publish_Date>1/11/2001</Publish_Date>
    <Articles>
      <Article ID=1>
        <Title>The Client who didn't know he needed SQL'</Title>
        <Author>Barry Munro</Author>
      </Article>
      <Article ID=2>
        <Title>Version Control System, What, When, Where</Title>
        <Author>Marco Kuipers</Author>
      </Article>
      <Article ID=3>
        <Title>Add spellcheck to your VDF in 10 minutes</Title>
        <Author>Marco Kuipers</Author>
      </Article>
    </Articles>
  </November_2001>
  <December_2001>
    <Publish_Date>10/12/2001</Publish_Date>
    <Articles>
      <Article ID=1>
```



# WWW.DDUG.ORG

## December newsletter



```
<Title>Codes do we need them?</Title>
<Author>Peter Brooks</Author>
</Article>
<Article ID=2>
  <Title>A 'Little' System Integration Job !</Title>
  <Author>Peter Tawse</Author>
</Article>
<Article ID=3>
  <Title>Debugging in a Multiuser Environment</Title>
  <Author>Clive Richmond</Author>
</Article>
<Article ID=4>
  <Title>XML you want it, even if you say you don't</Title>
  <Author>Marco Kuipers</Author>
</Article>
<Article ID=5>
  <Title>The Loneliness of the Long-distance Progr... </Title>
  <Author>Neil Hamilton</Author>
</Article>
</Articles>
</December_2001>
</DDUG_Newsletters>
```

When you study the XML document you understand why it is so much more powerful than CSV.

Above sample is easy to create using eg. Microsoft XML Notepad, but you can see, that this can be created just as easy with writeln statements in your program. You only have to be careful to live by your own standards.

I'm not saying, you shouldn't use a parser; I'm just saying it is simple enough to do yourself.

Now what do you do with this XML document? What about:

- Communicating to another system
- Outputting data for the web, eg telephone list (layout is in separate file).
- Etc...

So get involved, don't skip the articles, attend the DDUG December Meeting...

Marco Kuipers  
Marco.Kuipers@nci.com.au

This is short introduction, to help beginners to create a basic understanding of XML.



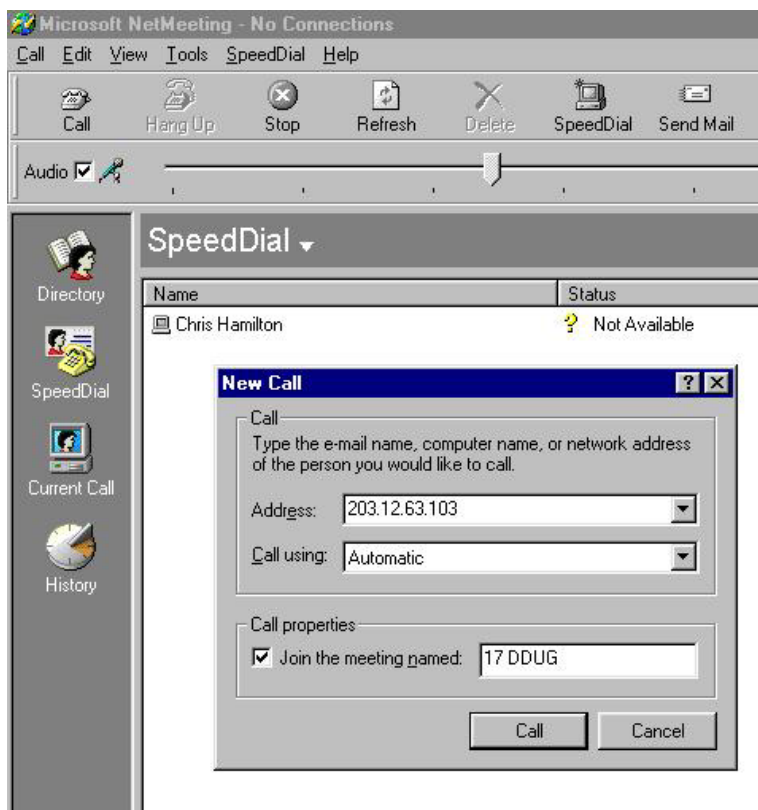
## 5. The Loneliness of the Long-distance Programmer

*Neil Hamilton*

DataFlexers not working in a “Software-shop” have a loneliness existence. It is not surprising that they are avid users of Newsgroups such as *news.dataaccess.com* and rush to be members of user groups like **DDUG**. The only problem is the members are very thinly spread on the ground and meetings rarely occur in their city or town. Air fare and accommodation costs prevent them popping down for a meeting or regional conference like the **DDUG Corroboree**. So they lose interest in the user group and the meetings stop happening.

One economic solution is to use the services of the phone-company and a friend who can arrange access to an Internet Conference Server. The trick is to find the best combination of their services.

DDUG is fortunate that through hard work and diligence it has gained access to an Internet Conference Server at 203.12.63.103.



This Internet Server is based on MS-Information Server 4.1 and supports **MS-NetMeeting 2.1x**.

This site restricts your operating system to Win95, Win98 and NT 4.1.

All the later versions of MS operating systems use **MS NetMeeting 3.2x** that means you cannot use them to use 203.12.63.103. Do not try an install the older version of NetMeeting onto your new computer – it will trash your computer. (Been there

done that).



## WWW.DDUG.ORG December newsletter



O.K. so what are we getting for our investment of zero dollars? (We have free access).

We can have up to 10 connections to the Server and up to 5 camera images can be seen simultaneously. The Conference Server also supports *Voice-over-IP*. Add a camera and microphone and you can see and talk to all other parties connected to the Conference Server.

But most importantly **any** computer in the Internet Conference can *share* their programs with all other computers attached to the Conference Server.

The only problem is the phone-company, the one providing the broad-band link to the Internet, is barring the *Voice-over-IP* for Cable Modem and ADSL

So we can see one another, share desktops, and communicate via the Chat Window, but only some of us can talk to one another. However, for those who can talk via *Voice-over-IP*, it is like a combination of short-wave radio and CB-Radio. At present this technology is less than satisfactory for an audience of people.

We have half of the solution! Now to find a way to share good quality voice communication.

Telstra, the dominant phone company in Australia, offers \$3.00 capped phone calls from 7:00pm to Midnight Sunday to Friday and from Noon to Mid-night on Saturday. This has greater quality than *Voice-over-IP* and you know the address (the phone number) of the other party. Thus a user group meeting in one city can share its proceedings with a meeting in another city for a total cost of \$6.00. One phone call for the voice and one phone call for sharing the computer desktop.

Telstra also has an Easy-Call “3-way talking” service in most large cities and towns. This service is available for phones not connected to a PABX. Normally this is used to ring a second party when you are in the middle of a phone call.

- Press [Recall] and wait a few seconds for a dial tone. The first party will hear music on hold from the phone company
- Dial the second party and establish the conversation.
- Press [Recall] [3] and the first party then joins you and the second party – giving a three-way conversation.

Now if we have more than one speaker-phone in each of the three locations we can have small groups of people hearing and see the same presentation. If the audience press the microphone-button on their speaker phone to “off” they will then remove any extraneous feed-back into the voice conference. To ask questions they merely press the microphone-button to “on”.



## WWW.DDUG.ORG December newsletter



A high resolution Video Projector (Beamer) gives the best image. Large TV has low resolutions and it is difficult to read any small writing.

O.K. we now have high quality voice and computer image for people in three separate locations, all for a total cost of long-distance phone call. Do this in the evening after 7:00pm local time or on Saturday afternoon and the total cost is \$6.00 for a 3 location meeting.

**You want more!** You want more than three venues with good quality and cheap voice communication. Well it can be done.

- Party Two, in your **3-way Talking** presses [Recall]. You now hear music from the phone company
- If Party Two **3-way Talking** facilities on their phone service can then dial party a fourth phone and establishes a conversation with them.
- Party Two then presses [Recall][3] and you now have a 4-way Conference.
- If Party One, the other half of your original call, has **3-way Talking**, they can also press [Recall] and dial a fifth phone number and establish a 5-way Conference.
- And so on, until you run out of people with **3-way Talking** service on their phones.

At this stage the total cost is \$15 for five venues if all initiations of calls have a local time after 7:00pm on Sunday to Friday or after mid-day on Saturday.

One or both final phones in the chain can be overseas / international calls. It is the last phone in the chain that is paying the International Call rates. Alternative the last part may be in a venue without **3-way Talking** e.g. a hired venue in the CBD.

The problem is when there are more than two phones without **3-way Talking**. Sorry, it cannot be done. You have to carefully limit yourself to this restriction.

The other problem is it takes time to establish all these links. Audiences get fidgety when the appointed time comes and nothing happens, it can take up to 15 minutes to create a 8-party conference using **3-way Talking** (your committee does this every month with their Board Meetings).

**But wait, you want even more!** You now have more than two locations with phones whom do not have **3-way Talking**. Let us say you want to involve New Zealand or some location in compatible Time zones. Well it can be done, at a cost much less than an airfare, but a bit more expensive than a long-distance phone call.



## WWW.DDUG.ORG December newsletter



Telstra has a **Conferlink** Service. People ring into a Sydney Gateway Phone Number, or dial a 1800 number to join a voice conference. Just like our **3-way Talking**, but it all happens instantly with any phone in the world, as long as you pay the whole phone bill!

If you do not have a **Conferlink** Account on with your phone company you can book a Conference. You phone Telstra and tell them the starting time, the number of parties and the length of the conference. They will send you the 1800 phone number, the conference number and the pin for your Conferlink Conference. Dial that number at the nominate time you have a high quality voice conference. International parties dial the Sydney Gateway Phone Number to join the Conference.

If you do this regularly you can establish a permanent **Conferlink account** with the **Telstra**. At they will give you a dedicated conference id and pin. As the owner of that **Conferlink account** you will have a special pin which means you can establish the **Conferlink** at any time from any phone.

Well, perhaps we do not need to be so lonely after all, as long as your computer has an operating system of Win95, Win98 or WinNT 4.1

Oh, in closing, please let us know if you can organise access to an Internet Conference Server that supports the latest version of NetMeeting. If you know of a cheaper set of solutions, please share your ideas.

NEIL HAMILTON