Clarion Magazine

Clarion News

- » Clarion.NET Cool Buttons Video
- » Clarion 7 Beta Out To CSP Members
- » Oak Park Solutions Server Plans On Sale
- » AmazingGUI Glassy Buttons Project Preview
- » Clarion Desktop 4.0
- » CPCS Office Schedule
- » Icetips Magic Locks
- » UKCUG New URL
- » EasyOpenOffice 1.05
- » ClarionFAQ Comments Disabled
- » AmazingGUI Site Fully Operative
- » Clarion Business Assets For Sale
- » Clarion 6 Vista Installation Fix Version 1.10
- » SetupBuilder 6.5 Build 1953

[More news]

- » Clarion 101: Designing A Database
- » Clarion 101: Understanding Keys and Indexes
- » Clarion 101: Describing your Data
- » Clarion 101: Choosing a Database
- » Clarion 101: Understanding the IDE, Part 2

[More Clarion 101]

Latest Free Content

- » The New (Old) Server
- » C7 Goes Out To CSP Subscribers On Sept 4

Latest Subscriber Content

The New (Old) Server

With not much time left before the server switchover our ISP swapped the RAID array into a new box, and it looks like ClarionMag is back on a stable platform. Here's hoping it stays that way.
Dave Harms, Publisher

Posted Tuesday, September 04, 2007

C7 Goes Out To CSP Subscribers On Sept 4

SoftVelocity has announced it will release the current C7 alpha to Core Subscription Program (CSP) subscribers on Tuesday, Sept 4, 2007. From Bob Z's blog entry it appears this release will include the Dictionary Editor and the new Dictionary Diagram tool, with the AppGen to follow at a later date. As alpha participants have found, it's relatively easy to create C7 versions of existing C6 apps even without the AppGen, although maintenance has to be done in C6. Source code apps can already be maintained entirely in C7. Runtime library enhancements in C7 include Unicode and ClearType support, tabbed MDI windows, XP styles, new sheet/tab styles, and encryption support in the Topspeed driver. The alpha includes C6 template updates to conditionally support C7 features when the code is compiled with the C7 compiler. Bob has also posted blog entries showing details of the Dictionary Editor and the Data Diagram tool.

Posted Tuesday, August 28, 2007

ClarionMag Summer Break

The Clarion Magazine will be closed until Tuesday, September 4. I will be checking email periodically but not daily. As usual all credit card purchases are processed automatically; PayPal purchases require one manual step and I'll clear all PayPal orders as quickly as I can.
Dave Harms, Publisher

Posted Monday, August 20, 2007

Latest Alpha Build

SV has released another build to the alpha test group - see the ClarionMag blog entry.

Posted Friday, August 17, 2007

PostgreSQL Revisited

When ClarionMag first covered PostgreSQL five years ago, the native Windows version of this popular and robust SQL server was in beta. In gold release since 2005, the Windows version is going strong and is an increasingly popular choice for Clarion developers.

Posted Wednesday, August 15, 2007

- » ClarionMag Summer Break
- » Latest Alpha Build
- » Summer Double Issue - July/August
- » Tips Vol 4 Now In Print
- » Source Code Library 2007.06.30 Available
- » First Look: The C7 Dictionary Editor
- » Bob Z's Blog Post On The New Designers, With Full Size Images
- » Source Code Library 2007.05.31 Available

[More free articles]

Clarion Sites

- » South African Clarion User Group
- » NetTalk Central
- » Ingasoftplus
- » RADFusion
- » PINVOKE.NET
- » SealSoft Company
- » Lodestar Software (CWaddons)
- » The Clarion Connection
- » BoxSoft (a.k.a. Mike Hanson)
- » CapeSoft

[More sites]

Clarion Blogs

- » ClarionX
- » AtlanWare
- » Johan van Zyl
- » RADBlog
- » Mark Sarson
- » Pimp My Clarion

Coping With Vista - Creating Directory Shares

Jane Fleming wraps up her discussion of Vista's Public directory and network sharing with a detailed look at creating a user-friendly network share.

Posted Tuesday, August 07, 2007

Coping With Vista - It's Nice To Share

Jane Fleming continues her series on Microsoft's latest OS with a look at Vista-specific CSIDLs and enabling network sharing.

Posted Friday, August 03, 2007

Summer Double Issue - July/August

As in past years, we're taking some summer holidays and combining July and August into a single issue. We have a number of articles lined up for August and will post updates on the C7 alpha as new releases become available. The monthly PDF, the source zip and the source code library update will be available at the beginning of September. Office closure dates will be posted.

Posted Wednesday, August 01, 2007

Tips Vol 4 Now In Print

Clarion Tips & Techniques Volume 4 is in print, and all pre-purchases have shipped.

Posted Wednesday, August 01, 2007

Showing ABC Errors

The ABC library automated much of the error handling which previously had to be hand coded. But ABC also made it more difficult to get at error codes, and mixing ABC and "legacy" code has its own problems. Steve Parker unpacks ABC's error handling and shows how to discover and display ABC errors.

Posted Friday, July 20, 2007

New Alpha 2 Build Improves DCT UI

A new alpha build has been released with significant improvements to the dictionary editor user interface. Dave Harms reports.

Posted Friday, July 13, 2007

Profiling Your Applications

Your application is working, but is it working too hard? Once the code is doing what it should do, it's time to make sure it's not wasting your customer's time. Philip Prohm shows how to use Clarion's built-in profiling hooks to find bottlenecks in running code.

Posted Wednesday, July 11, 2007

Source Code Library 2007.06.30 Available

The Clarion Magazine Source Code Library has been updated to include the June source. Source code subscribers can download the January-June 2007 update from the My ClarionMag page. If you're on Vista please run Lindersoft's Clarion detection patch first.

Posted Friday, July 06, 2007

First Look: The C7 Dictionary Editor

SoftVelocity has released the first C7 alpha two build, including a preview version of the Dictionary Editor. Dave

- » DevDawn
- » John Griffiths
- » Gary James
- » Clarion Magazine

[More blogs]

Harms reports. (Free article: registration required)

Posted Tuesday, July 03, 2007

[Last 10 articles] [Last 25 articles] [All content]

Source Code

The ClarionMag Source Code Library

Clarion Magazine is more than just a great place to learn about Clarion development techniques, it's also home to a massive collection of Clarion source code. Clarion subscribers already know this, but now we've made it easier for subscribers and non-subscribers alike to find the code they need.

The Clarion Magazine Source Library is a single point download of all article source code, complete with an article cross-reference.

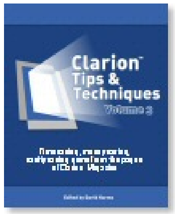
More info • Subscribe now

Printed Books & E-Books

E-Books

E-books are another great way to get the information you want from Clarion Magazine. Your time is valuable; with our e-books, you spend less time hunting down the information you need. We're constantly collecting the best Clarion Magazine articles by top developers into themed PDFs, so you'll always have a ready reference for your favorite Clarion development topics.

Printed Books



As handy as the Clarion Magazine web site is, sometimes you just want to read articles in print. We've collected some of the best ClarionMag articles into the following print books:

- » Clarion 6 Tips & Techniques Volume 3 - ISBN: 0-9689553-9-8
- » Clarion 6 Tips & Techniques Volume 1 - ISBN: 0-9689553-8-X
- » Clarion 5.x Tips and Techniques, Volume 1 - ISBN: 0-9689553-5-5
- » Clarion 5.x Tips and Techniques, Volume 2 - ISBN: 0-9689553-6-3
- » Clarion Databases & SQL - ISBN: 0-9689553-3-9

We also publish Russ Eggen's widely-acclaimed Programming Objects in Clarion, an introduction to OOP and ABC.

From The Publisher

About Clarion Magazine

Clarion Magazine is your premier source for news about, and in-depth articles on Clarion software development. We publish articles by many of the leading developers in the Clarion community, covering subjects from everyday programming tasks to specialized techniques you won't learn anywhere else. Whether you're just getting started with Clarion, or are a seasoned veteran, Clarion Magazine has the information *you* need.

Subscriptions

While we do publish some free content, most Clarion Magazine articles are for subscribers only. Your subscription not only gets you premium content in the form of new articles, it also includes all the back issues. Our search engine lets you do simple or complex searches on both articles and news items. Subscribers can also post questions and comments directly to articles.

Satisfaction Guaranteed

For just pennies per day you can have this wealth of Clarion development information at your fingertips. Your Clarion magazine subscription will more than pay for itself - you have my personal guarantee.

Dave Harms

ISSN

Clarion Magazine's ISSN

Clarion Magazine's International Standard Serial Number (ISSN) is 1718-9942.

About ISSN

The ISSN is the standardized international code which allows the identification of any serial publication, including electronic serials, independently of its country of publication, of its language or alphabet, of its frequency, medium, etc.

Clarion Magazine

Clarion News

Search the news archive

Clarion.NET Cool Buttons Video

Bob Z has posted a video showing a new glass style button for Clarion.NET.

Posted Thursday, September 06, 2007

Clarion 7 Beta Out To CSP Members

Clarion Core Subscription Plan subscribers are now receiving emails with download instructions for the Clarion 7 beta.

Posted Thursday, September 06, 2007

Oak Park Solutions Server Plans On Sale

All of Oak Park's server plans are now on sale for the month of September. Virtual Dedicated Servers are on sale for as little as $29.99 per month with no long-term commitments. These servers are great for Nettalk Webserver applications or pretty much whatever you want to run on them, including setting up your own small hosting company for your clients. Dedicated servers are also available.

Posted Thursday, September 06, 2007

AmazingGUI Glassy Buttons Project Preview

A preview of AmazingGUI's Glassy Buttons Project is available for download.

Posted Thursday, September 06, 2007

Clarion Desktop 4.0

Clarion Desktip 4.0 is ready for release. But here's the deal. There are currently 99 Pro Edition users, and 411 Free Edition users. When there are 100 Pro Edition users Garry will release version 4.0...

Posted Thursday, September 06, 2007

CPCS Office Schedule

Larry Teames will be out of office from 9/05-9/07/2007, and will not have email access. He will handle any outstanding issues as quickly as possible upon his return.

Posted Thursday, September 06, 2007

Icetips Magic Locks

Icetips Magic Locks consist of a set of Classes and templates that enhance the View Only forms to better indicate read-only controls. Magic Locks also locks checkboxes, radio buttons, drop down controls, etc. which can not be set as read-only. This reduces confusion, as users can not make visual changes to the data. Magic Locks can also allow updates of certain fields in View mode, which is not possible with regular View mode. This allows users to update certain fields even if the rest of the form is completely locked. Settings for each control can be overridden in the procedure template. Background color, font color and font style for locked controls can be set with a global template and overridden in a procedure template. Demo available.

Posted Thursday, September 06, 2007

UKCUG New URL

The UKCUG is now at a new URL. Any useful links relevant to Clarion can be listed on the website and third party suppliers can now also have links listed on the website as well for free. The next meeting is expected to be sometime in November 2007, venue to be decided. The UKCUG's first webinar (web seminar) is also under consideration; those who live overseas or can't get to the venue could still take part and even ask questions. That's only if enough people are interested; there will be a charge to participate in the webinar.

Posted Thursday, September 06, 2007

EasyOpenOffice 1.05

EasyOpenOffice 1.05 is now available. Numerous bug fixes, class and template changes. EasyOpenOffice is a set of classes and templates allowing you to exchange data between Clarion applications and OpenOffice Calc and Writer.

Posted Thursday, September 06, 2007

ClarionFAQ Comments Disabled

Jeff Slarve has temporarily removed the "Comments" and "Ask a Question" options from ClarionFAQ.com because they were being heavily abused by spammers. If anyone has questions or comments, please email them to Jeff directly.

Posted Friday, August 17, 2007

AmazingGUI Site Fully Operative

During the last days there have been problems with the amazingGUI web site server that were impeding demos and versions downloading. This problem is solved now and the web site is completely operative.

Posted Friday, August 17, 2007

Clarion Business Assets For Sale

Laurence Cope is handling the sale of Simon Burrows' Clarion-related business assets, including List & Label, VariView, Layout Manager, Mailmerge, Wordspell, ResortPro, Live Update and Solace Free Templates. Also for sale is Data Ferret/Prospector, a screen scraper that imports search engine results into a contact management database. For further information contact Laurence Cope on +44 (0)870 0101781 or email laurence at p-s-c.co.uk.

Posted Friday, August 17, 2007

Clarion 6 Vista Installation Fix Version 1.10

Version 1.10 of Lindersoft's "Clarion 6 Vista Installation Fix" application is now available. Due to file virtualization Clarion third party vendor installations cannot autodetect the installed Clarion version. This program makes the appropriate WIN.

INI modifications. This is a freeware application and you can redistribute it to your customers. You can use the fix on Windows Vista and Windows Server 2008 ("Longhorn"). Because Clarion 6 cannot be run on 64-bit versions of Windows, the application detects x64 now and displays a warning. Just follow the wizard to apply the fix and all 3rd party installers should be able to auto-detect your Clarion 6 environment on Vista and Windows Server 2008.

Posted Friday, August 17, 2007

SetupBuilder 6.5 Build 1953

SetupBuilder 6.5 Build 1953 is primarily a maintenance release that fixes reported issues and adds several other refinements. Lindersoft strongly recommends customers upgrade to the latest version of SetupBuilder 6.5 as soon as possible to maintain the highest level of support, performance and reliability.

Posted Friday, August 17, 2007

ClarioNET 6.3

A new 6.3 update release of ClarioNET is now available. It corrects a bug related to the report previewer, and adds compatibility with the latest CPCS release.

Posted Friday, August 17, 2007

CoolButtons 1.04 beta is now available.

CoolButtons 1.04 beta changes include: Added support for button resizing; Fixed bug where "Special buttons" did not work with multi DLL apps; Revamped images for colour consistency and size; Fixed bug where re-colouring produced white blotches in some images; Fixed bug where the button background was sometimes corrupted.

Posted Friday, August 17, 2007

NetTalk WebServer Coverage

Stu Andrews has a blog post on integrating third party tools into NetTalk WebServer.

Posted Tuesday, August 07, 2007

InBack Now At Huenuleufu

Jeff Slarve (JS Software), the original developer of InBack, has transferred the sales and development of InBack to Huenuleufu Development, along with a sizable list of user suggestions and feature requests. So far Huenuleufu has already adapted the program to the latest compiler technology and fixed some bugs, while making sure that it has a modern look and works better in Windows Vista. One new feature is the ability to recognize a command line that points directly to each backup set (i.e. "/1" launch the project number 1, and so on). This has the immediate benefit to allow you to program a direct link to the backup system in Windows' Programmed Tasks, to launch the backup set that you desire, even when you are not present. There are other enhancements as well, and many more planned. InBack will be available in two levels: The current standard one, and a new "professional" one that will have most of the new enhancements such as enhanced filtering, FTP backup, and later zip 64 support (for unlimited file sizes). The current InBack pro version adds date filtering to a given mask. If you are an original user, you have an opportunity to upgrade to the new InBack 1.6, and InBack Pro, for US$19: purchase at https://www.clarionshop.com/secure/checkout.cfm?pid=1042&q=1

Posted Tuesday, August 07, 2007

Tagkeys Maintenance Update

A maintenance update (version 1.3) is available from Tagkeys. This update affects all current users of Tagkeys. Changes include: Changed the default Tagkey from a semicolon (;) to a period (.) to provide default compatibility with keyboards other than QWERTY (this will affect all version 1.2 users - if you previously downloaded Tagkeys and wish to keep using the semicolon as your tag character, you must select it from the Setup page); Included functionality to automatically close Tagkeys when Windows closes; Added three new macros for registered users (PC User, PC Name, Preview Only); Added 10 date macros for registered users that lets you insert a date in either MM-DD-YYYY or DD-MM-YYYY format and use any of five separators (slash/dash/period/comma/space); Added "Extended Marcos" button for registered users that allows one-click access to the additional 24 macros; Fixed bug when exporting Tagsets and the root directory of any drive was selected; The installation program has changed and no longer includes Clarion specific tagsets preloaded in the database (all of the previous Tagsets are still available and can be freely downloaded from www.tagkeys.com/downloads); New SQL Basics tagset (various SQL scripts [mostly selects] that show the SQL basic SQL format for selecting, updating, and modifying tables and such).

Posted Tuesday, August 07, 2007

Icetips Extends Subscription Sale

Icetips Creative Inc. has extended its 33% introduction discount for Gold and Silver subscriptions through August 10, 2007.

Posted Tuesday, August 07, 2007

Trial Pay Icons

Dave Beggs is offering his icons using a trial pay option. Regular purchase is also available.

Posted Tuesday, August 07, 2007

Free Quickcrypt Wrapper

Jeff Slarve has posted a free class wrapper for the Quickcrypt encryption library on ClarionFAQ.

Posted Tuesday, August 07, 2007

Ruler Class

Andrew Popoff has made his Ruler Class free to use.

Posted Tuesday, August 07, 2007

Chicago User Group Meeting Aug 8

The next Chicago Clarion User Group meeting will be held Wednesday August 8th, 2007 at 6:30 p.m. in Suite 270 of Nebo Systems offices. The address is 1 South 450 Summit Ave. Suite 270, Oakbrook Terrace, IL 60181. The sign at the front of the office complex says Summit Oaks. There will be a roundtable discussion which will center on "Clarion, the Internet and Information retrieval", moderated by Jim Palmer.

Posted Tuesday, August 07, 2007

Clarion Connection 10 Years Old

The Clarion Connection is now 10 years old, having first appeared in July 1997. Congrats to Tom Ruby!

Posted Tuesday, August 07, 2007

MS SQL Explorer Becomes MS SQL Field Box

The ex-SealSoft tool MS SQL Explorer has changed its name to MS SQL Field Box. MS SQL Field Box is a small tool which allows you to view MS SQL Server database structure and paste table or field names in your favorite editor using mouse or keyboard hotkeys.

Posted Tuesday, August 07, 2007

Scheduler Class Released

Due to time constraints Roel Abspoel has released his schedule class as is. Demo and some docs are included. This is a beta release but has been tested in production environments.

Posted Tuesday, August 07, 2007

BG SoftFactory Back Online

Bernard Grosperrin's BG SoftFactory web site is back online.

Posted Monday, August 06, 2007

Looking For Clarion-Based Products

An existing Clarion-based software company in the U.S. is seeking to expand and add to its line of software products. Products must have an existing customer base with reasonable annual revenue from sales and support.

Posted Monday, July 30, 2007

Icetips Calendar System

The new Icetips calendar system lets you see what Icetips is working on.

Posted Monday, July 30, 2007

Clarion Magazine

PostgreSQL Revisited

by Dave Harms

Published 2007-08-15

It’s almost five years since I wrote an introduction to PostgreSQL for Clarion Magazine. In that article I provided some background to PostgreSQL’s academic upbringing, and allowed that while PostgreSQL was increasingly popular, it had some distance to go to catch up to its main rival, MySQL.

A fair bit has changed in five years, although some things are the same. MySQL still has massive market share, and easily beats PostgreSQL in a Google Fight. PostgreSQL is arguably technically superior, while MySQL is more widely supported and, at least for simpler operations, faster. There are, however, a few developments that warrant a new look at PostgreSQL from a Clarion developer’s perspective.

Five years ago the main open source release of PostgreSQL was just edging onto the Windows platform. You could get a beta release of that product (several commercial Windows versions were available somewhat earlier), but if you didn’t want the free beta and you didn't want to pay big bucks for a commerical version and you weren’t running some flavor of Linux/Unix you had to settle for PostgreSQL running under Cygwin, a *nix environment for Windows. I tried Cygwin, and it was ugly. Very ugly.

Five years on and the Windows version of PostgreSQL is well-established; version 8.0, released in late 2005, was the first gold release to include a native Windows version. Reportedly there have been over a million downloads of the 8.1 release, with two thirds of these being the Windows version. By comparison, MySQL AB reports about ten million MySQL installations worldwide.

PostgreSQL has long been a robust, transaction-oriented database; MySQL gained those capabilities with the InnoDB add-on file system. InnoDB, however, was purchased by Oracle in 2005. Oracle and MySQL are competitors, which makes MySQL's reliance on InnoDB somewhat problematic. Oracle has said it will continue to support MySQL with InnoDB, but this is not an ideal situation. Last year Solid Information Technology announced solidDB for MySQL, effectively a replacement for InnoDB, and MySQL AB is working on Falcon, which is also intended as a replacement for InnoDB. Falcon is definitely a work in progress, and solidDB is still in beta. Right now InnoDB remains the engine of choice for transactional SQL, if you can live with InnoDB’s uncertain future.

Years ago, before InnoDB, MySQL was a fast, flat-file database with an SQL interface. If you needed performance more than data integrity, if you wanted fast reads more than reliable writes, if you wanted to run on platforms from Solaris to Windows, you went with MySQL. If you could sacrifice performance for data integrity, and you didn’t need a native Windows database server, if you needed the ability to extend the database with custom types, you looked hard at PostgreSQL.

With the releases of PostgreSQL 8 and MySQL 5 the gap between the two products has narrowed considerably (again, keeping in mind that much of MySQL’s improvement in reliability is dependent on InnoDB). MySQL now has triggers, stored procedures, and subselects; PostgreSQL’s performance, particularly under very heavy load, can match or better MySQL.

Technically the two databases are closer than ever, and both are open source products, but they differ considerably in their licensing models. MySQL is released under multiple licenses; if you are releasing a GPL (Gnu Public License) program, you can use MySQL under the GPL license. If you are releasing a non-GPL program, however, you fall under the commercial license. You can contact MySQL sales for a quote or buy a license online starting at $595 per year per server for MySQL Enterprise Basic.

PostgreSQL, on the other hand, is free to use, and is released under the BSD license, as follows:

> PostgreSQL Database Management System
>
> (formerly known as Postgres, then as Postgres95)
>
> Portions Copyright (c) 1996-2005, The PostgreSQL Global Development Group
>
> Portions Copyright (c) 1994, The Regents of the University of California
>
> Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

> IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
>
> THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Well that’s pretty dang clear. No fees, and no recourse against the creators!

PostgreSQL is gaining mind share among Clarion developers, as near as I can tell. Five years ago it was rare to see a PostgreSQL question or comment posted in the newsgroups, no doubt because of the lack of a gold-release, open source Windows version; now PostgreSQL posts are relatively common. And in a recent ClarionMag poll PostgreSQL came in a strong second behind MySQL and just ahead of Firebird.

Installing PostgreSQL

I recently upgraded my desktop machine to Vista, and that’s where I elected to install PostgreSQL. Whatever your Windows OS of choice (assuming it's W2K or later) you’ll need to download the Win32 binaries. Go to the PostgreSQL downloads page and click on the download mirrors link. Choose a mirror and your browser will show you the FTP directories. I clicked on binary | v8.2.4 | win32 and downloaded postgresql-8.2.4-1.zip, but by the time you read this article a newer version of PostgreSQL may be available.

Unzipping to a temporary directory yielded the following files:

```
20/04/2007 12:06 PM 27,619,840 postgresql-8.2-int.msi
20/04/2007 12:06 PM 137,216 postgresql-8.2.msi
20/04/2007 09:24 AM 1,008 README.TXT
04/12/2006 11:02 AM 1,016 upgrade.bat
```

> The README.TXT file is brief and basically advises you to read the installation instructions at and the FAQ. The batch file is for those upgrading from a previous version of PostgreSQL, and the smaller of the two MSI install files is the one you run (it calls the larger file).

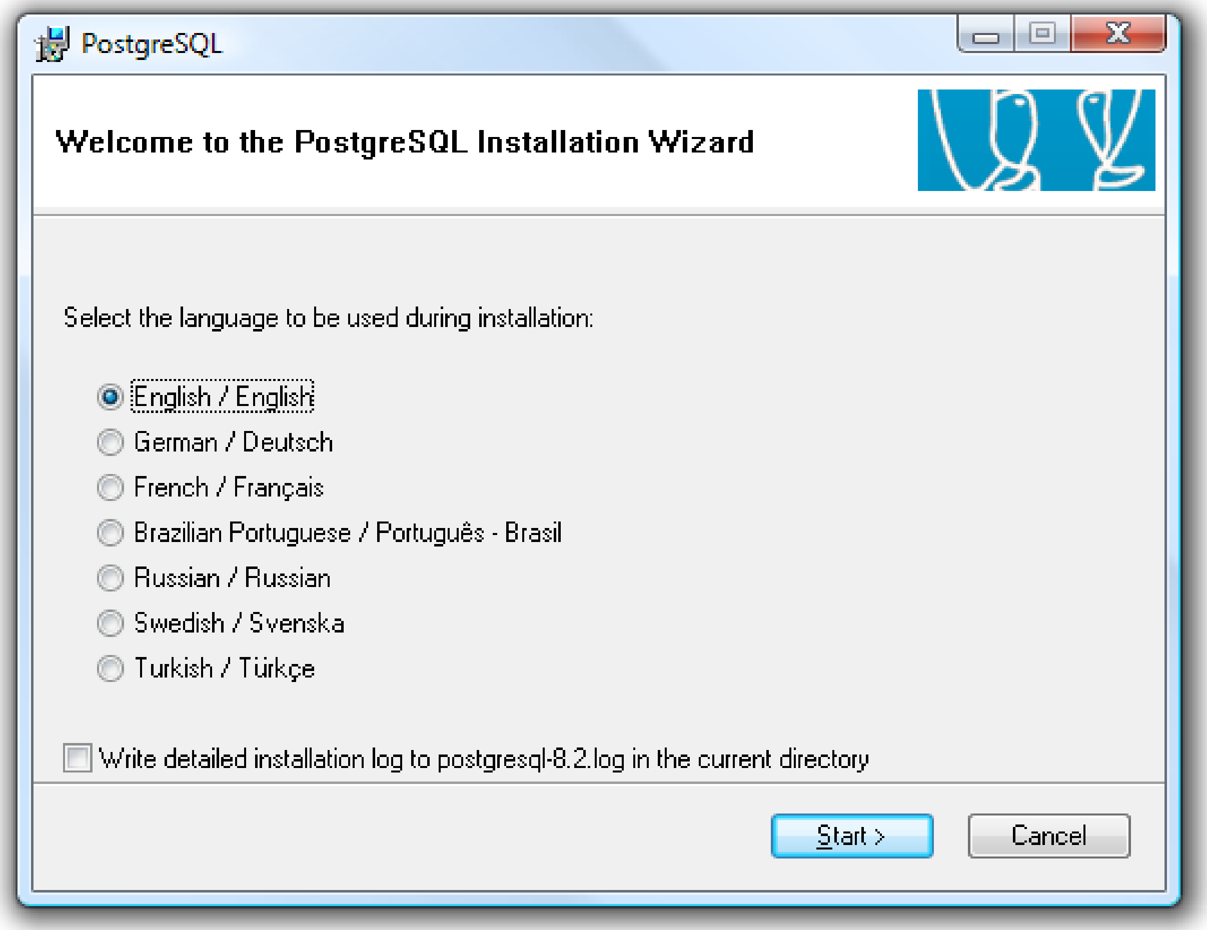
If you're installing on W2K or XP then all you need to do is execute postgresql-8.2.msi to begin installation. If you're on Vista and you have UAC turned on, things get a bit more complicated. Running the install as a normal user will, according to what I've read, result in an install failure because the installation program needs sufficient rights to create a new Windows user named postgres, under which name the server will run.

Following Jane Fleming's advice I set up a command prompt shortcut with administrative rights. I executed postgresql-8.2.msi from the command prompt and the install worked fine, because programs run from an elevated console window inherit those rights. Note that right-clicking and choosing Run as administrator doesn't seem to be an option with .MSI files.

Keep in mind that PostgreSQL requires NTFS. PostgreSQL will run on NT4 but in that case you'll have to compile it from source and you won't be able to use tablespaces. If you enjoy pain you might want to deploy on NT4.

If Unicode/UTF-8 matters to you, you should go with version 8.1 or later; PostgreSQL 8.0 has incomplete support for Unicode. Specifically you can store data in Unicode with that release but you will have problems with sort orders and with UPPER, LOWER, and ORDER BY. If you think Unicode is a non-issue because it isn't supported by Clarion 6, keep in mind that Clarion 7 *does* have Unicode support.

Figure 1 shows the first window of the PostgreSQL Installation Wizard.



PostgreSQL

Welcome to the PostgreSQL Installation Wizard

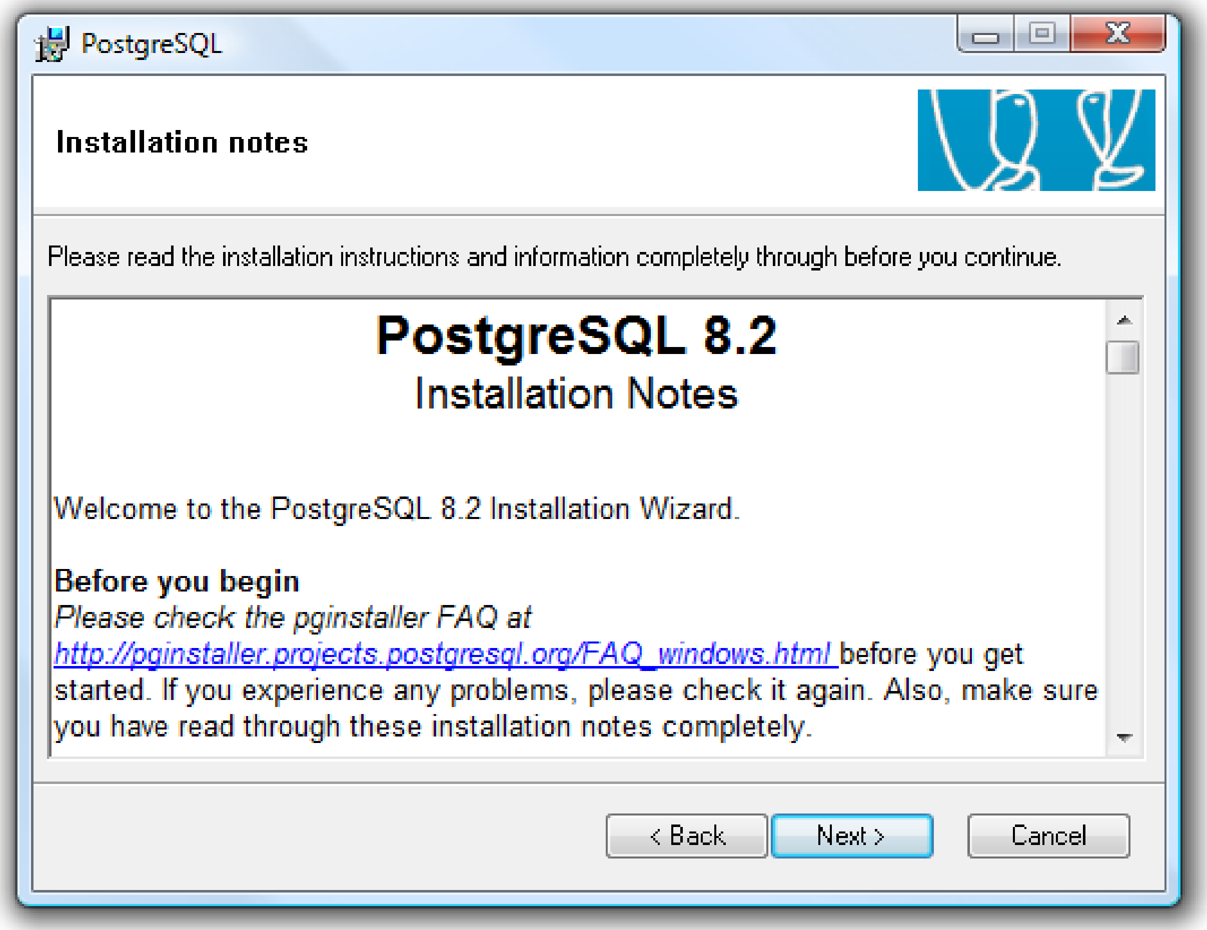
Select the language to be used during installation:

● English / English
○ German / Deutsch
○ French / Français
○ Brazilian Portuguese / Português - Brasil
○ Russian / Russian
○ Swedish / Svenska
○ Turkish / Türkçe

□ Write detailed installation log to postgresql-8.2.log in the current directory

Figure 1. Installing PostgreSQL

If you choose to create a detailed installation log you'll find it contains a detailed list of the installer's actions, including registry edits and configuration file settings.



Please read the installation instructions and information completely through before you continue.

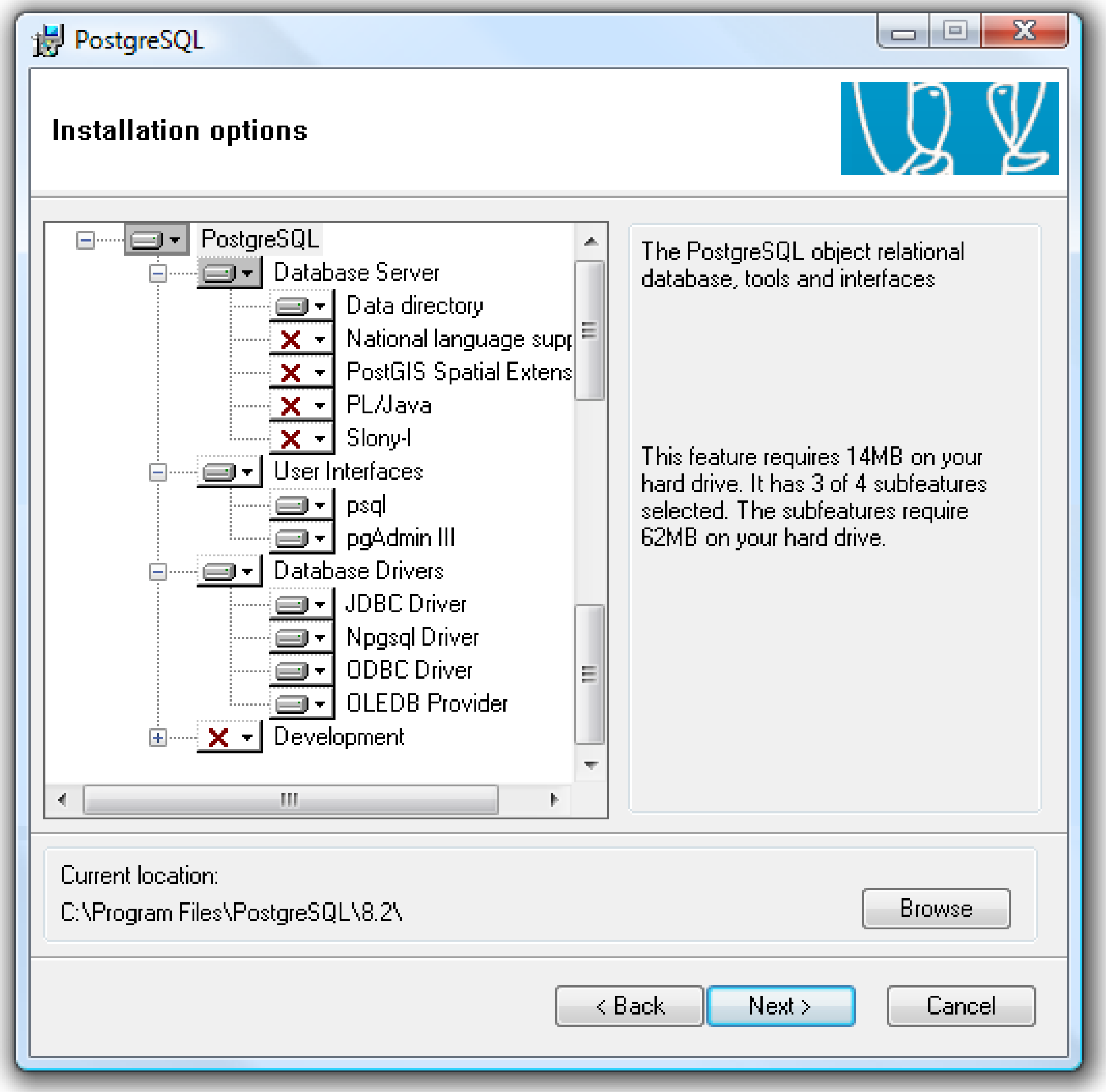
PostgreSQL 8.2
Installation Notes

Welcome to the PostgreSQL 8.2 Installation Wizard.

Before you begin
Please check the pginstaller FAQ at http://pginstaller.projects.postgresql.org/FAQ_windows.html before you get started. If you experience any problems, please check it again. Also, make sure you have read through these installation notes completely.

Figure 2. Installation notes

The installation notes window (Figure 2) reminds you to read the FAQ and the installation notes before proceeding.



Installation options

- PostgreSQL
  - Database Server
    - Data directory
    - ☒ National language supp
    - ☒ PostGIS Spatial Extens
    - ☒ PL/Java
    - ☒ Slony-I
  - User Interfaces
    - psql
    - pgAdmin III
  - Database Drivers
    - JDBC Driver
    - Npgsql Driver
    - ODBC Driver
    - OLEDB Provider
  - ☒ Development

The PostgreSQL object relational database, tools and interfaces

This feature requires 14MB on your hard drive. It has 3 of 4 subfeatures selected. The subfeatures require 62MB on your hard drive.

Current location:
C:\Program Files\PostgreSQL\8.2\

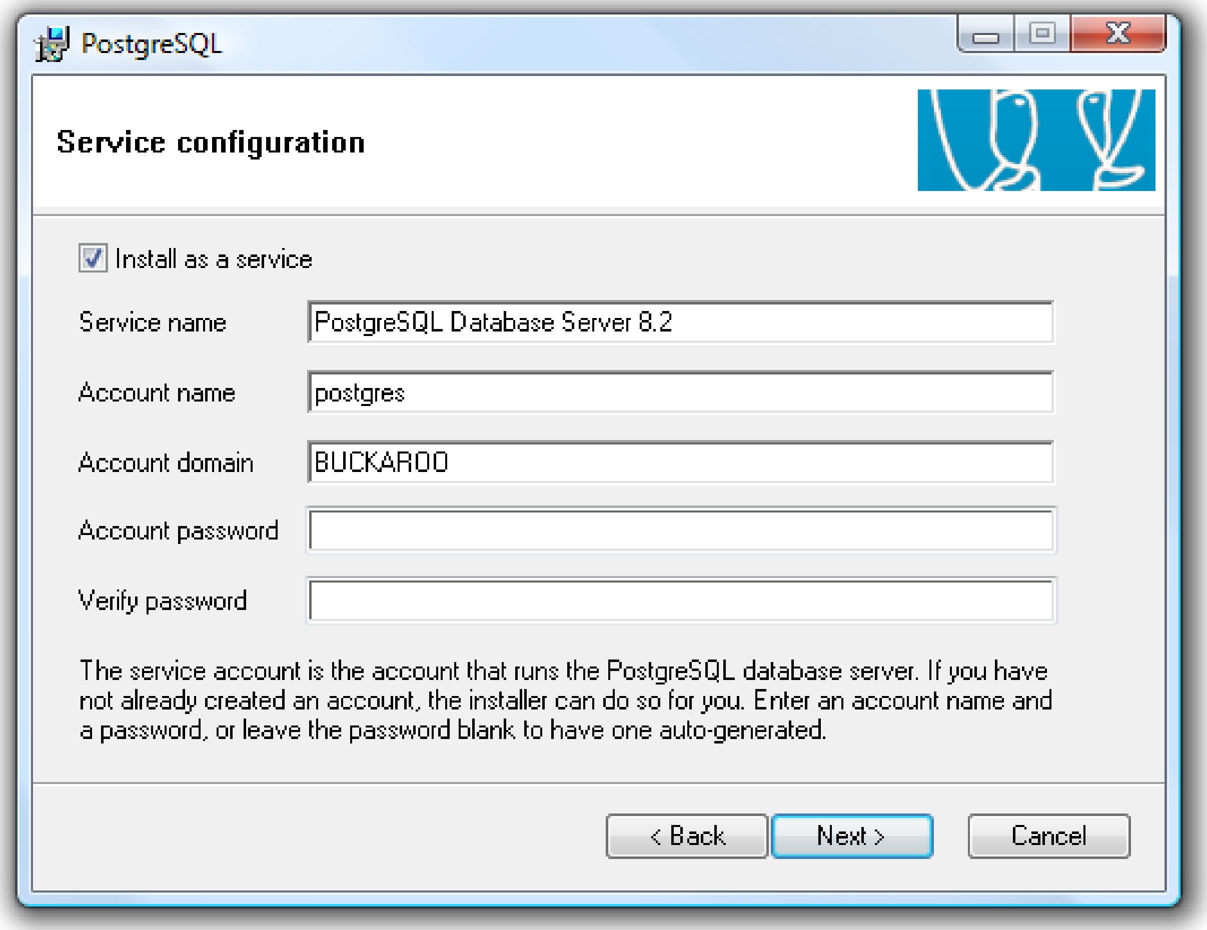
Figure 3. Installation options

I've doctored Figure 3 just a little so it shows all of the installation options at once. These include:

- Data directory - this will create a tablespace and the default databases

- National language support - see status and error messages in the language of your choice. Also adds language support for psql.
- PostGIS – one of the interesting things about PostgreSQL is the way its data types can be extended, and have been extended for various purposes. PostGIS is a geographic information system extension that adds support for geographic objects such as POINT, LINESTRING, POLYGON, MULTIPOINT, etc. This could be quite useful if you write software that involves any kind of geographic information. A geocoder for TIGER/line data is available.
- PL/Java - lets you create stored procedures in Java (you must have a Java runtime environment installed)
- Slony-I - this add-on provides cascading replication, i.e. one master to multiple slaves. It is designed for use at data centers and backup sites.
- psql – the command line interpreter
- pgAdmin III – a graphical management console.
- JDBC – JDBC driver (like ODBC but for Java).
- Npgsql – the .NET driver (a.k.a. data provider).
- ODBC driver – this is the one you'll use with Clarion (Windows version, that is).
- OLEDB driver – also known as ADO.
- Development tools – since PostgreSQL is open source, it’s easy to get the source and build PostgreSQL to your own tastes. But you probably won’t want to bother.

If you later decide you want one of these options you can always re-run the PostgreSQL installer. At least you should be able to. I didn't install PostGIS the first time around, but decided I wanted it later. So I ran the installer again, and it reported that another installer was running and it couldn't complete. That seemed odd, so I tried running the installer (again, from an elevated command prompt) with the same result. I rebooted, tried again, still got the error. I had two buttons to click: Cancel and Retry. In a fit of frustration I clicked Retry about a dozen times, and what do you know, the install went through. I haven't tested PostGIS yet, however.



Service configuration

☑ Install as a service

Service name: PostgreSQL Database Server 8.2

Account name: postgres

Account domain: BUCKAROO

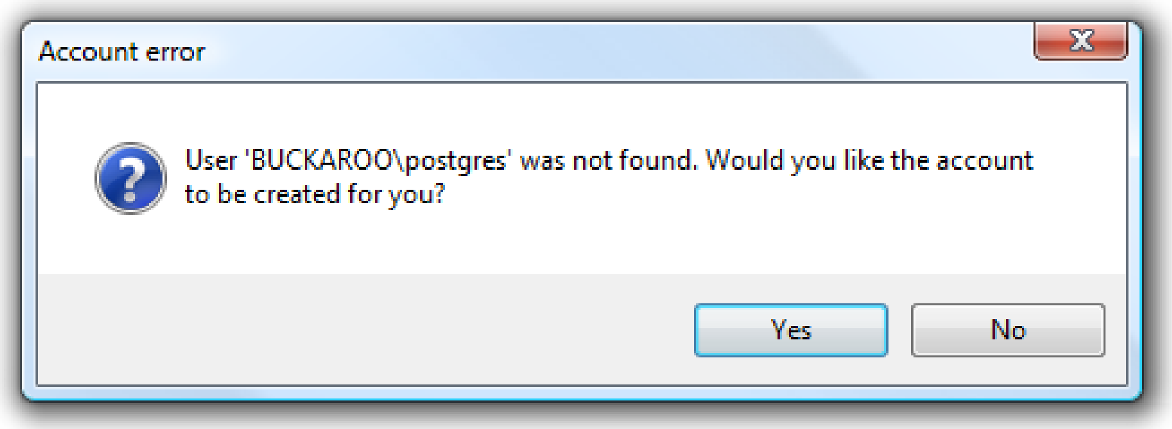
Account password:

Verify password:

The service account is the account that runs the PostgreSQL database server. If you have not already created an account, the installer can do so for you. Enter an account name and a password, or leave the password blank to have one auto-generated.

Figure 4. Setting up the service

Normally you'll want to install PostgreSQL as a service; if you don't, then you'll need to start and stop PostgreSQL manually. To run as a service PostgreSQL has to run under a user, and the install program will create this user for you (which is one of the reasons the installer needs admin rights). All you need to do is choose a password for this user.

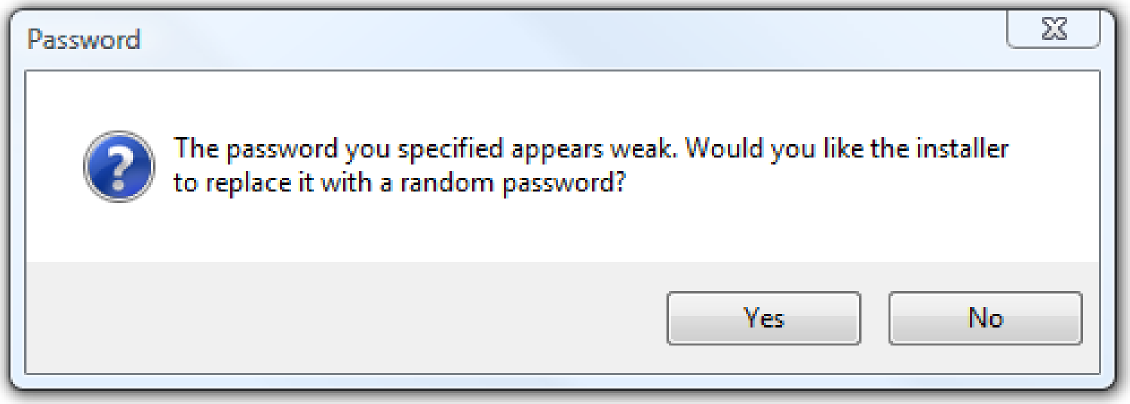


Account error

User 'BUCKAROO\postgres' was not found. Would you like the account to be created for you?

Figure 5. Creating the postgres user

If your password seems too guessable the installer will advise you and offer to generate a more secure password (Figure 6).

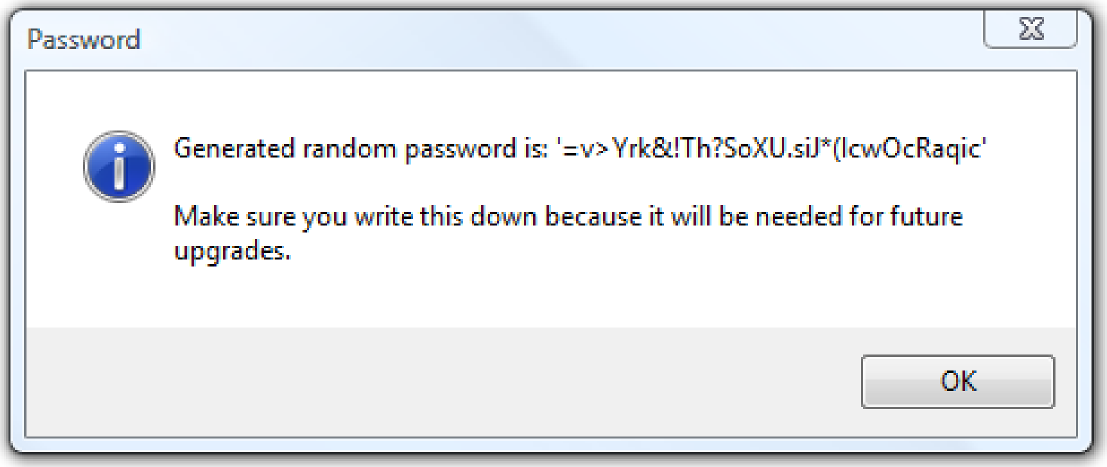


Password

The password you specified appears weak. Would you like the installer to replace it with a random password?

Figure 6. Not secure enough for ya, buddy?

Be careful about taking up the installer on its offer of a better password. Figure 7 shows the result.



Generated random password is: '=v>Yrk&!Th?SoXU.siJ*(lcwOcRaqic'

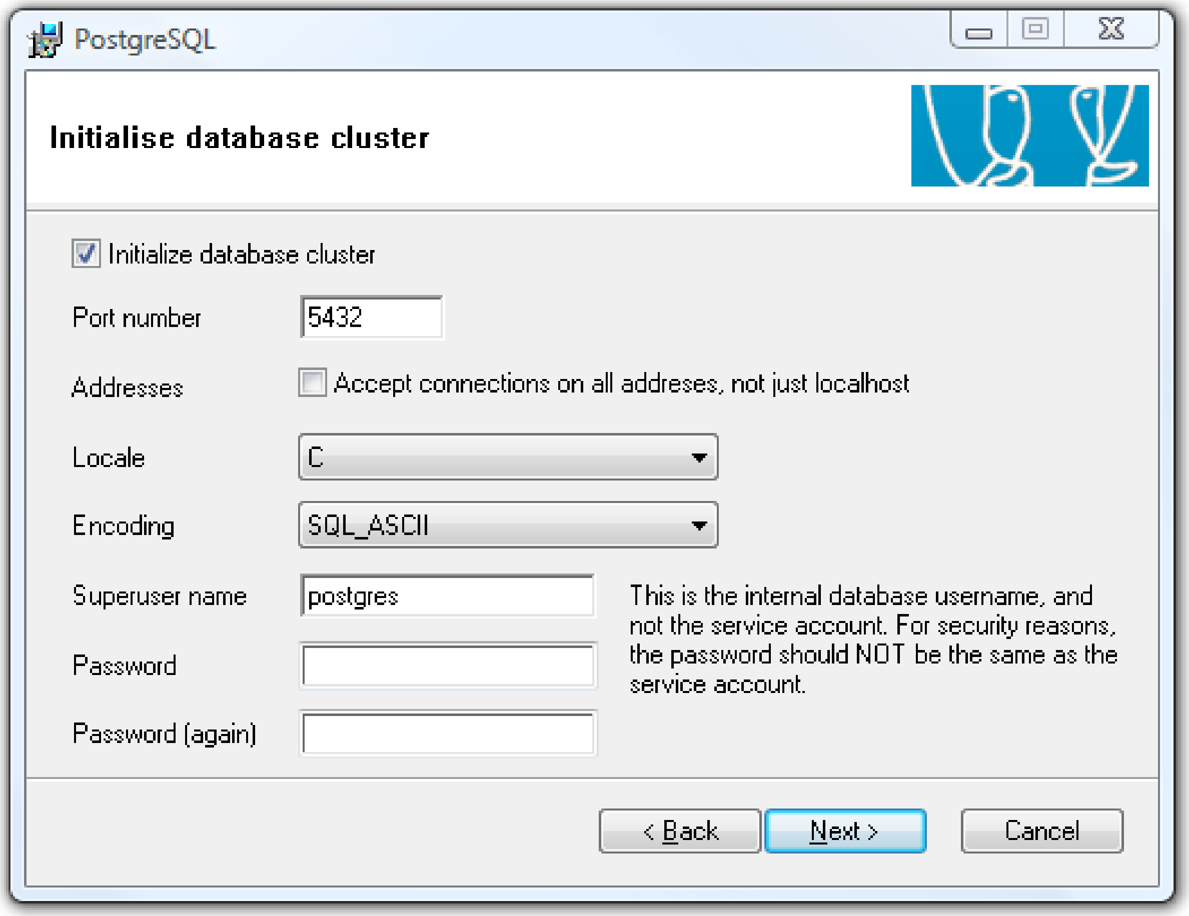
Make sure you write this down because it will be needed for future upgrades.

Figure 7. That's not a password. *This* is a password.

Yikes! I think I'll stick with my own password, thanks.

The next step is initializing a database cluster, or tablespace. Every database has to be in a cluster, but a cluster can contain more than one database. All databases in a cluster share the same locale settings and the same data encoding (see Figure 8).

Locale and encoding are two quite different settings. Locale determines the sort order, and if you're using non-English data you'll want an appropriate locale setting so your sorts sort correctly. You may also need to choose an encoding that supports your locale (and vice versa). But just because you create an initial database cluster with a particular encoding and locale and later realize you need something different doesn't mean you're hosed; all you need to do is create a new database cluster with the desired settings. Each database cluster also has a unique port number; that way you can have multiple instances of a database with the same name, but in different clusters. This is handy for test databases; all you need to do to access another database is change the port number, and if you're using ODBC you'll do that in the ODBC data source setup. Nothing whatsoever needs to change in your application.



Initialise database cluster

☑ Initialize database cluster

Port number: 5432

Addresses: ☐ Accept connections on all addreses, not just localhost

Locale: C

Encoding: SQL_ASCII

Superuser name: postgres

Password:

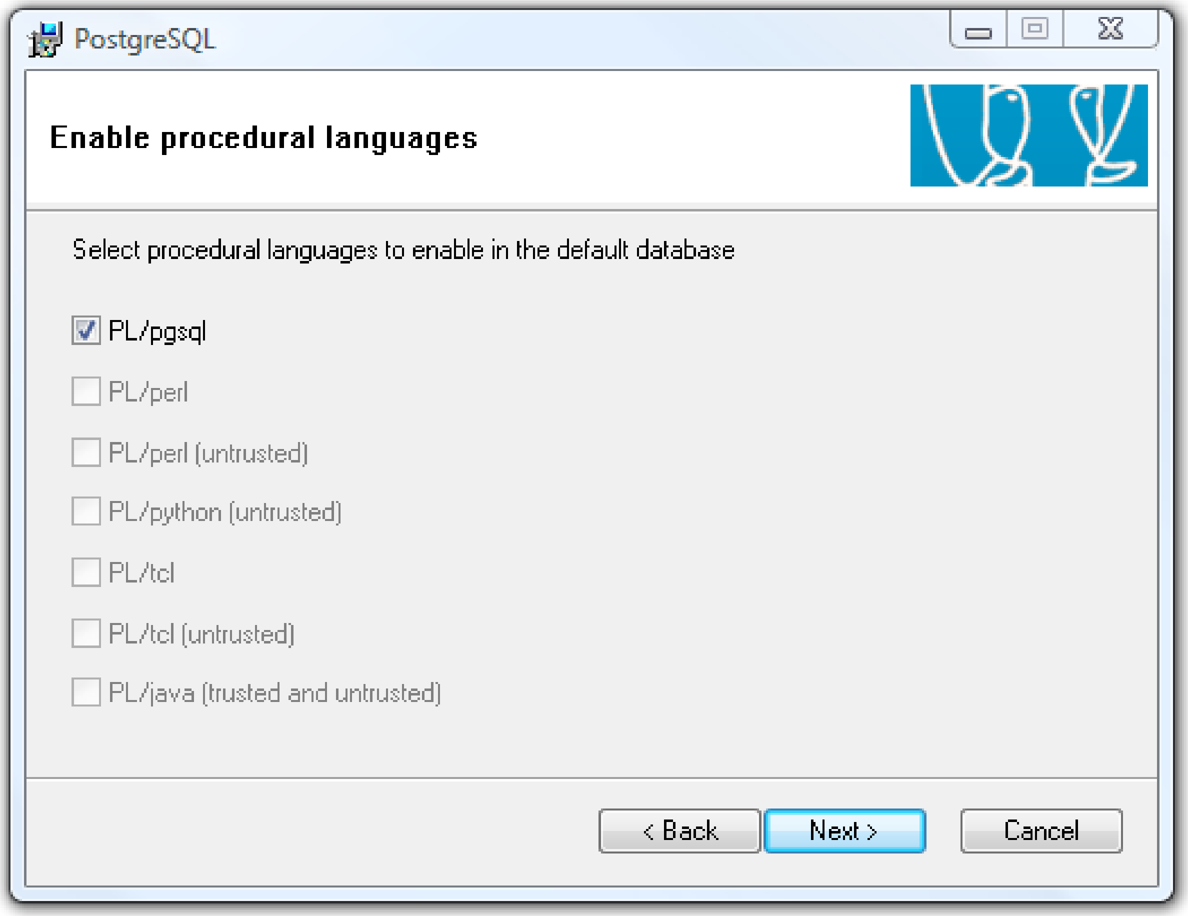
Password (again):

This is the internal database username, and not the service account. For security reasons, the password should NOT be the same as the service account.

Figure 8. Creating a database cluster

Take special note of this second requirement for a password to go with the superuser, which defaults to postgres just like the service user name. This is *not* the same user; the previous postgres user is a Windows user, while this is a Postgresql database user. Don't get the two passwords confused, as I did the first time around.

PostgreSQL supports a number of locales and encodings; I created my initial database cluster using the English (Canadian) locale and Unicode encoding.



Enable procedural languages

Select procedural languages to enable in the default database

☑ PL/pgsql

□ PL/perl

□ PL/perl (untrusted)

□ PL/python (untrusted)

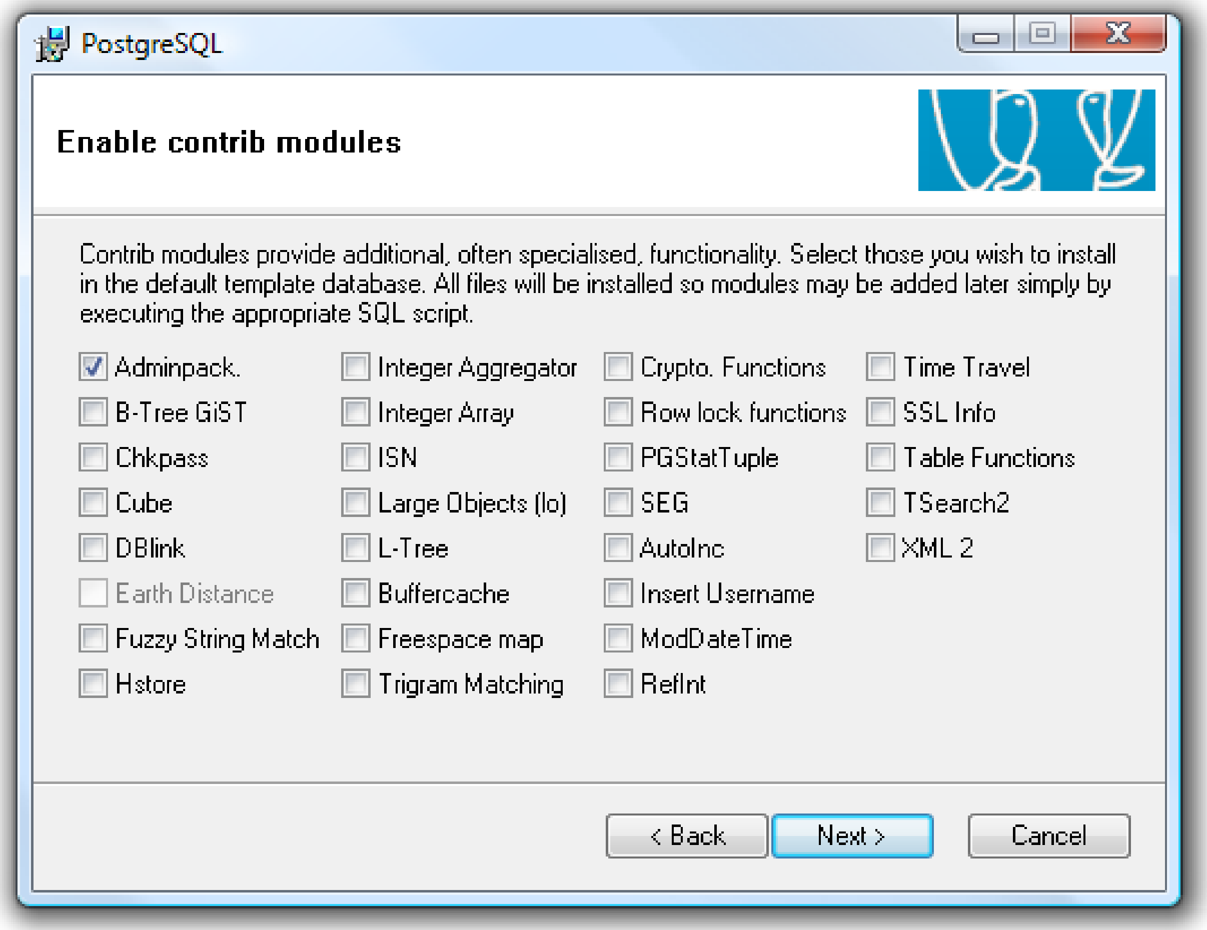
□ PL/tcl

□ PL/tcl (untrusted)

□ PL/java (trusted and untrusted)

Figure 9. Procedural languages

Earlier in the install you had the option of installing support for Java as a language for stored procedures; the window in Figure 9 lets you also add support for stored procedures written in Perl, Python and Tcl, as well as PostgreSQL's own pgsql language.



Enable contrib modules

Contrib modules provide additional, often specialised, functionality. Select those you wish to install in the default template database. All files will be installed so modules may be added later simply by executing the appropriate SQL script.

☑ Adminpack.
☐ B-Tree GiST
☐ Chkpass
☐ Cube
☐ DBlink
☐ Earth Distance
☐ Fuzzy String Match
☐ Hstore
☐ Integer Aggregator
☐ Integer Array
☐ ISN
☐ Large Objects (lo)
☐ L-Tree
☐ Buffercache
☐ Freespace map
☐ Trigram Matching
☐ Crypto. Functions
☐ Row lock functions
☐ PGStatTuple
☐ SEG
☐ AutoInc
☐ Insert Username
☐ ModDateTime
☐ RefInt
☐ Time Travel
☐ SSL Info
☐ Table Functions
☐ TSearch2
☐ XML 2

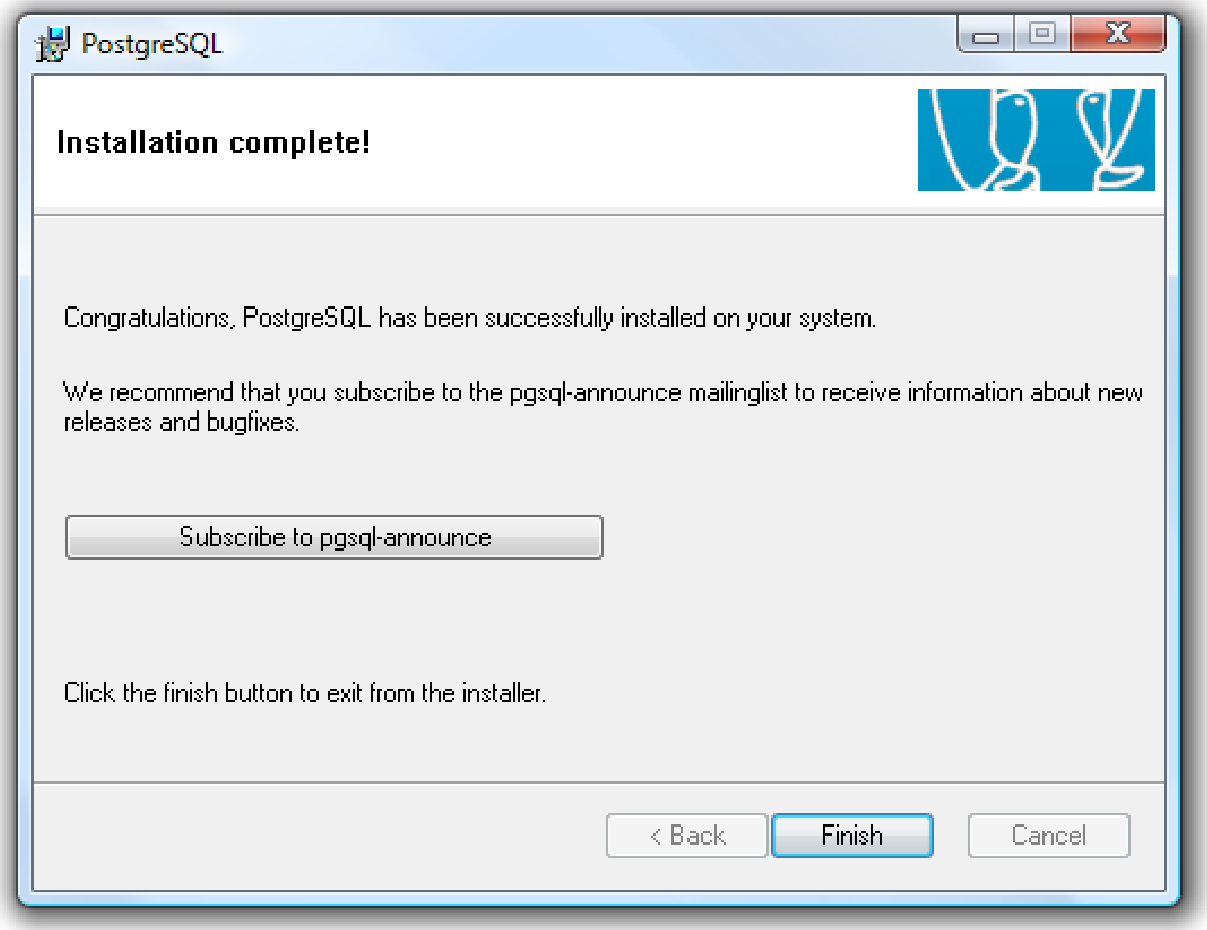
Figure 10. Contrib modules

There are a great many user-contributed modules for PostgreSQL. You can find the SQL scripts under PostgreSQL's share\contrib subdirectory, and the readme files in doc\contrib. As noted in Figure 10, you can install these later by executing the appropriate SQL script. The modules shown in Figure 10 include:

- B-Tree GiST - from the docs: "This is a B-Tree implementation using GiST that supports the int2, int4, 2 int8, float4, float8 timestamp with/without time zone, time 3 with/without time zone, date, interval, oid, money, macaddr, char, 4 varchar/text, bytea, numeric, bit, varbit and inet/cidr types."
- Chkpass - encrypted password data type
- Cube - cube data type

- DBlink - allows cross-database queries using function calls
- Earth Distance - calculate distances from lat/long
- Fuzzy String Match - soundex and other fuzzy matches
- Hstore - storing key/value pairs
- Integer Aggregator - a summary table for speeding up certain joins
- Integer Array - "an implementation of RD-tree data structure using GiST interface of PostgreSQL"
- ISN - book data type validation including EAN13, UPC, ISBN, ISSN etc.
- Large Objects - workaround for problem with deleting BLOBs using JDBC and ODBC
- L-Tree - tree structures
- buffercache - examing the buffer cache
- Freespace map - examine the free space map
- Trigram Matching - testing for similarity of text based on trigram matching
- Crypto. Functions - PGP and OpenSSL encryption
- Row lock functions - show row locking information
- PGStatTuple - helps determine if vacuum is necessary
- SEG - high forehead stuff re the geometry of measurements
- AutoInc - auto-incrementing code, no detailed info found
- Insert Username - inserts the current user's name
- ModDateTime - modify a timestamp
- RefInt - could be referential integrity triggers, in which case the code is obsolete as of PostgreSQL 7. Have a look at the C source.
- Time Travel - in PostgreSQL time travel usually refers to being able to historical queries by marking records as deleted, rather than actually deleting them, but I'm not sure that this is the intent of the code.
- SSL Info - SSL functions available if PostgreSQL is configured/built with SSL support
- Table Functions - various functions including random numbers, crosstabs
- TSearch2 - full text search
- XML 2 - XPath and XSLT functionality

Once you're done with the installation you'll see the final window which gives you a convenient way to get on the pgsql-announce mailing list. I've been on that list for some time and it's relatively quiet, averaging one or two emails a month. I recommend you sign up; if nothing else you'll be on top of any security-related issues that may arise. You'll also receive notices of product updates and upcoming events, like OSCON, O'Reilly's annual open source convention. Clarion Magazine covered OSCON last year and PostgreSQL was clearly a crowd favorite.



Installation complete!

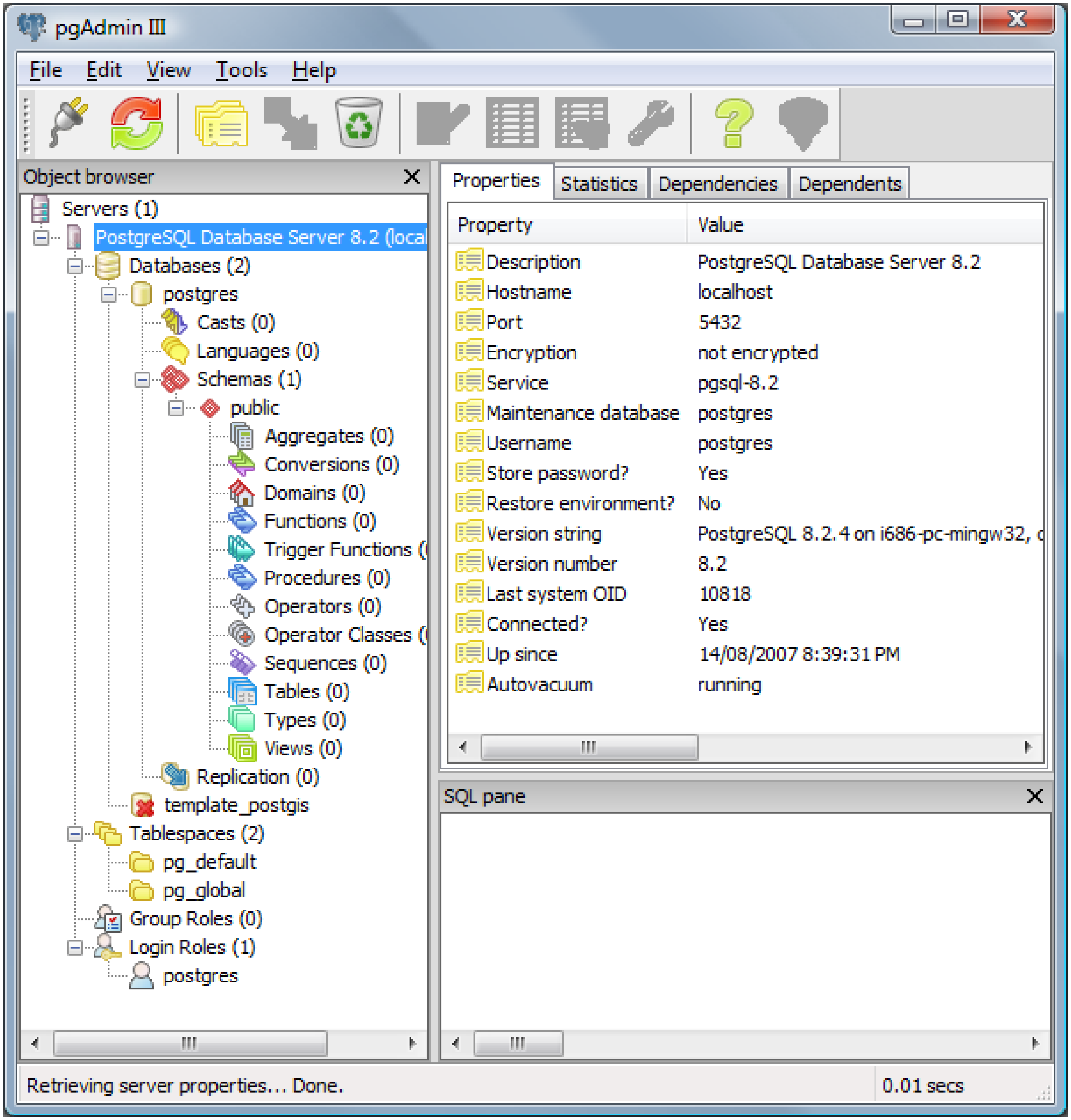
Congratulations, PostgreSQL has been successfully installed on your system.

We recommend that you subscribe to the pgsql-announce mailinglist to receive information about new releases and bugfixes.

Click the finish button to exit from the installer.

Figure 11. Installation is complete

Now that you have PostgreSQL installed, you're ready to being creating and using databases. I'll go into that in more detail in the next installment, but for now you might want to explore a little with PGAdmin III (remember the *second* password you supplied - you'll need that to connect to the server as the superuser).



Servers (1)
PostgreSQL Database Server 8.2 (local
Databases (2)
postgres
Casts (0)
Languages (0)
Schemas (1)
public
Aggregates (0)
Conversions (0)
Domains (0)
Functions (0)
Trigger Functions (
Procedures (0)
Operators (0)
Operator Classes (
Sequences (0)
Tables (0)
Types (0)
Views (0)
Replication (0)
template_postgis
Tablespaces (2)
pg_default
pg_global
Group Roles (0)
Login Roles (1)
postgres

Property	Value
Description	PostgreSQL Database Server 8.2
Hostname	localhost
Port	5432
Encryption	not encrypted
Service	pgsql-8.2
Maintenance database	postgres
Username	postgres
Store password?	Yes
Restore environment?	No
Version string	PostgreSQL 8.2.4 on i686-pc-mingw32, c
Version number	8.2
Last system OID	10818
Connected?	Yes
Up since	14/08/2007 8:39:31 PM
Autovacuum	running

Retrieving server properties... Done. 0.01 secs

Figure 12. pgAdmin III

You may also want to have a look at Roberto Artigas's Dct2SQL utility templates. If you currently use TPS files and you're accustomed to letting your Clarion apps create the database, you'll need to change your approach. In the SQL world you're better off creating tables with SQL scripts and migrating any changes back to the dictionary. If you have a dictionary already then Dct2SQL will give you a head start on creating those scripts.

David Harms is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995), and has written or co-written several Java books. David is a member of the American Society of Journalists and Authors (ASJA).

Reader Comments

Add a comment

Clarion Magazine

Clarion Tips & Techniques Volume 4

At over 650 pages, Clarion Tips & Techniques Volume 4 is another blockbuster brimming with terrific articles from Clarion Magazine. **ISBN: 978-0-9784034-0-9**

Buy now in the Clarion Magazine store.

Topics covered in this book include:

- Browses & Forms
    - Internationalization standards
    - Dynamic listbox formatting
    - Edit-In-Place
    - Hot fields
    - Recursive inserts and updates
    - Beautifying Clarion apps
    - Automatically closing windows

Source code

ClarionTipsTechniquesVolume4Source.zip (8.62 MB)

Errata

Corrections to the book will be posted here as necessary. If you have found an error please report it.

Clarion Magazine

First Look: The C7 Dictionary Editor

by Dave Harms

Published 2007-07-03

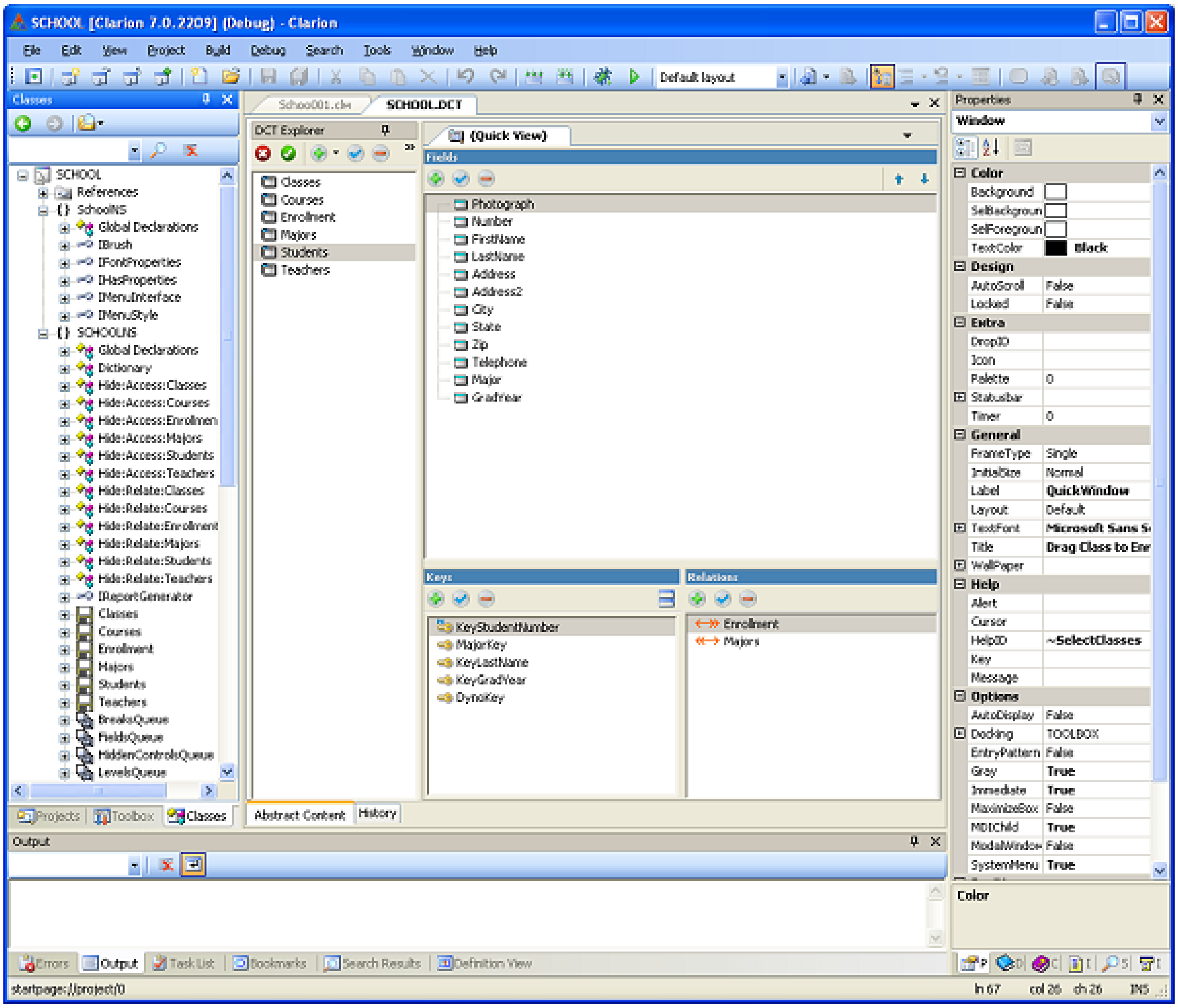
Late last week SoftVelocity released the first C7 alpha phase two build containing a preview version of the Dictionary Editor. Recent indications were the next alpha build would contain both the dictionary editor and the new visual data designer. The data designer has been held back until the next build to resolve a few nagging issues and improve usability with large dictionaries.

The data dictionary release in this latest build isn't a fully functional version in that you can't save your changes. That, according to Z, is to restrict testing to the conversion to the new DCT format and to get some feedback on usability. The next build is expected next week. And since this build is to gather feedback on the UI, don't be surprised to see some changes by the time it's in beta.

There are a number of new features in this second alpha release, including a massive number of bug fixes (many reported by alpha testers, but many others found and fixed by SV). In this article, however, I'll focus strictly on the new Dictionary Editor.

Flattening the IDE

One of the early pronouncements Bob Zaunere made on Clarion 7 was that the IDE would be flattened; that is, you wouldn't have to drill down many levels of windows to set particular values. More to the point, given the IDE support for multiple versions of Clarion, this new flattening would apply to the old template chains as well as to any new templates. I've always been curious as to how SV would implement that flattening, and the Dictionary Editor, I suspect, points the way.



Color	
Background	□
SelBackgroun	□
SelForegroun	□
TextColor	■ Black
Design	
AutoScroll	False
Locked	False
Extra	
DropID	
Icon	
Palette	0
Statusbar	
Timer	0
General	
FrameType	Single
InitialSize	Normal
Label	QuickWindow
Layout	Default
TextFont	Microsoft Sans S
Title	Drag Class to Ent
WallPaper	
Help	
Alert	
Cursor	
HelpID	~SelectClasses
Key	
Message	
Options	
AutoDisplay	False
Docking	TOOLBOX
EntryPattern	False
Gray	True
Immediate	True
MaximizeBox	False
MDIChild	True
ModalWindow	False
SystemMenu	True

Color

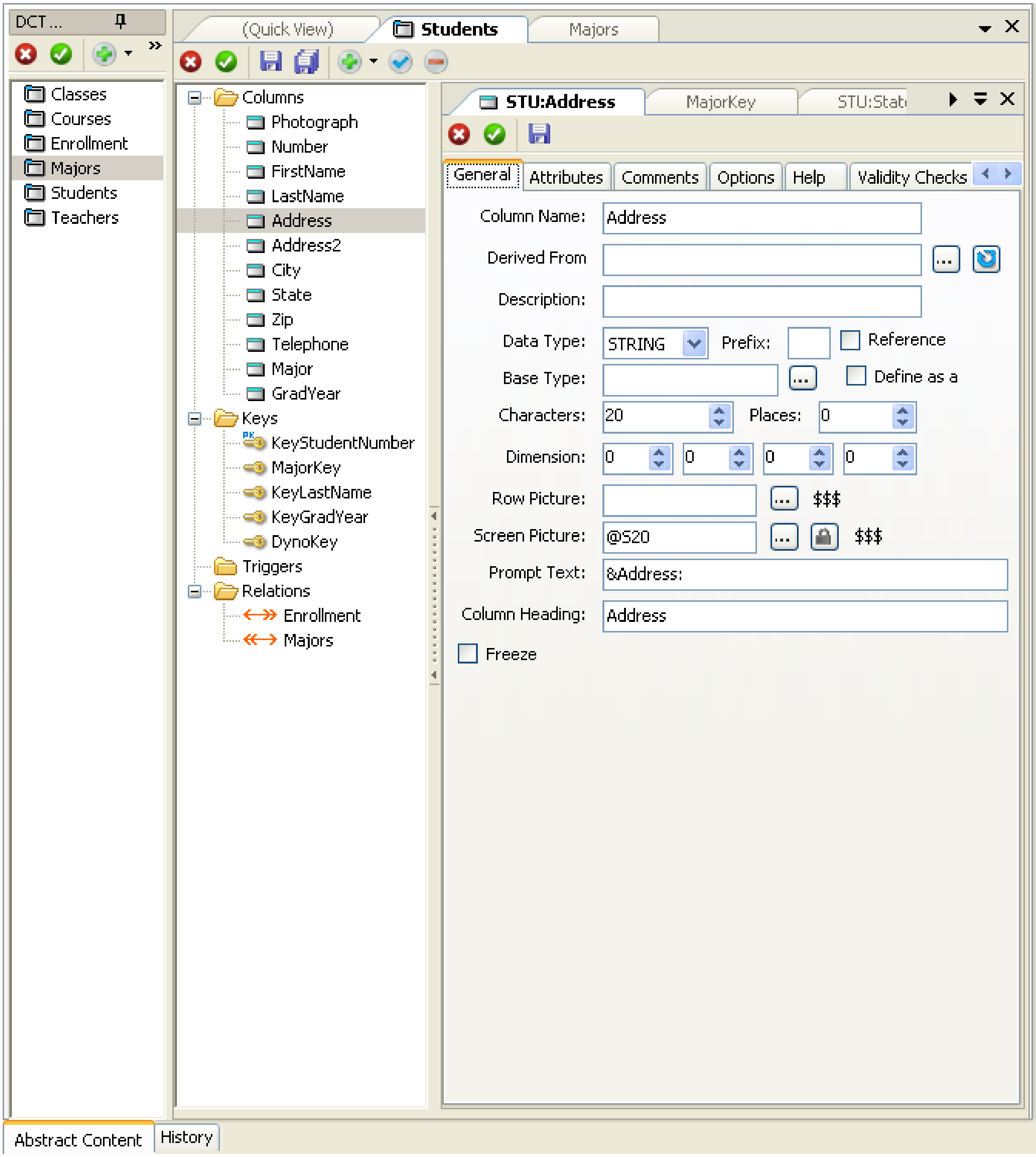
Figure 1. The IDE with a dictionary loaded (view full size image)

Figure 1 shows the IDE with a dictionary open. The first thing to note is that the dictionary editor occupies the same space in the IDE as the source editor or the window/report designer. There's been a suggestion that the dictionary table list should appear in the Projects toolbox, and I think that has a certain logic. The other issue is that the property toolbox, as seen on the right hand side of the IDE, is non-functional for the dictionary editor. As a result, you may want to switch to "Plain layout" when editing the dictionary, as this will hide both the project and property toolboxes and devote the entire IDE surface to the dictionary editor.

As I'll show below, a separate property toolbox does come into play in the dictionary editor, so you may wonder why you can't just keep the regular property toolbox up and use that. And perhaps someday that will be the case. But doing so (and using the project toolbox for the tables list) raises complex questions of design, implementation, and usability. For one thing, the dictionary editor prompts in C7 look familiar. Switching everything to a property toolbox-based approach would require a bigger learning curve (and there's already much that's different about the new IDE). For another, plumbing in a connection to the standard property toolbox will most likely take more time and introduce the possibility of bugs. And forcing the use of that property toolbox may otherwise constrict the use of screen real estate for other tasks where the properties are not needed.

At this point I don't think SV's approach presents any major problems, and it has the benefit of familiarity for existing Clarion users. And it does go a long way toward flattening the IDE.

Figure 2 shows just the DctEd, squashed down to roughly its minimum usable width. I've loaded several fields and keys form the Students and Majors tables, and have highlighted the STU:Address field.



Column Name: Address

Derived From:

Description:

Data Type: STRING Prefix: □ Reference

Base Type: □ Define as a

Characters: 20 Places: 0

Dimension: 0 0 0 0

Row Picture: $$$

Screen Picture: @S20 $$$

Prompt Text: &Address:

Column Heading: Address

□ Freeze

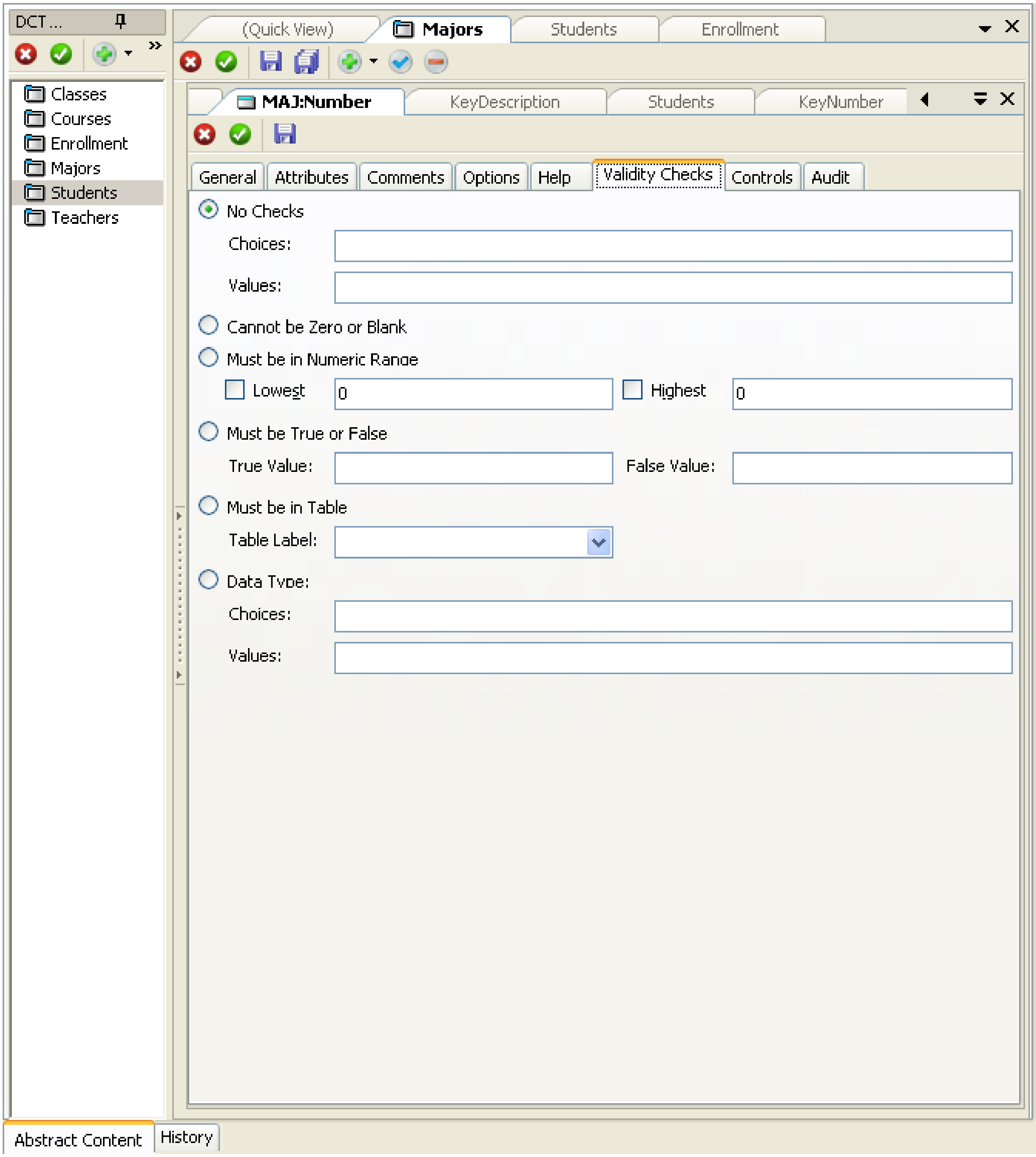
Figure 2. A dictionary with several fields and one key selected

There are a couple of things to note. The first is that thanks to the tabbed interface you can quickly switch between tables and fields (and go back to the summary via the Quick View tab). The IDE remembers the field level tab (as you'd expect) so if you were working on the General tab in one field and switched to Validity checks on another field (perhaps in another table), when you come back to the first field the General tab is still displayed.

You can also drag both the table and field tabs and change the order to suit yourself.

At present you have to double-click to load fields but this is being worked on, along with making the DctEd more keyboard friendly.

You may have noticed the odd-looking vertical control in Figure 2, between the second and third panes. Click on that control and the center panel disappears (see Figure 3).



● No Checks

Choices:

Values:

○ Cannot be Zero or Blank

○ Must be in Numeric Range

□ Lowest 0 □ Highest 0

○ Must be True or False

True Value: False Value:

○ Must be in Table

Table Label:

○ Data Type:

Choices:

Values:

Figure 3. The DctEd with the center panel collapsed

When you hide the center panel the right hand panel expands, and the controls on any sheets expand as necessary. If you've already selected the fields you want this is a nifty way to gain some real estate. Double-clicking (at present) on any table automatically unhides the center panel, as does clicking again on the control.

Figure 4 shows some of the new toolbar buttons. The red X is close, the green check is close and save. The disk icons are Save and Save All.

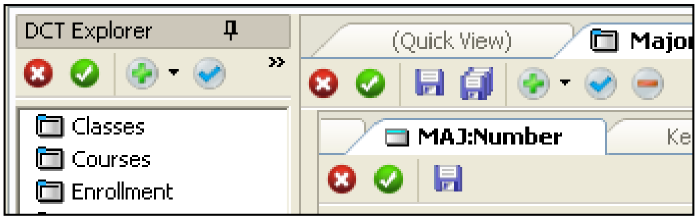


Figure 4. Toolbar buttons

In the DCT Explorer toolbar the green plus sign means Add, and if you click on the little triangle you get the menu shown in Figure 5.

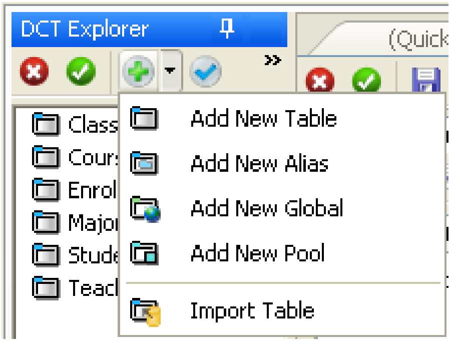


Figure 5. The Add menu in the DCT Explorer

Figure 6 shows the menu that appears when you click on the >> button.

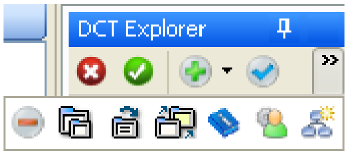


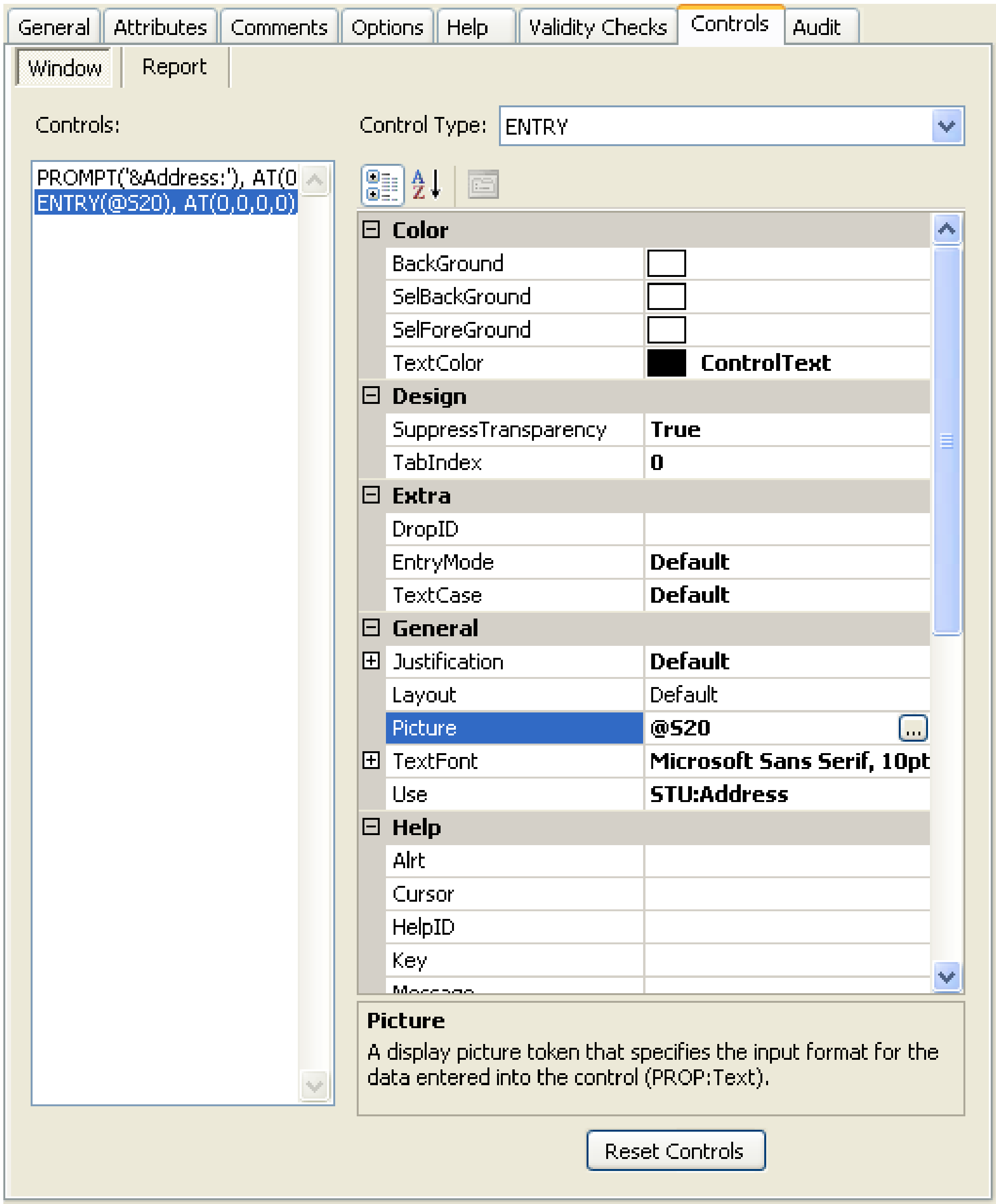
Figure 6. The expanded menu

The menu items in Figure 6 are, in order: Delete, Edit Table, Browse Table, Convert Table, Dictionary Properties, Users, and New Diagram. I can't tell you a whole lot about these yet since Edit Table is the only one that seems to work (although you can browse databases with this release, via the Tools menu.

The table menu (see Figure 4) also has a green plus button and a menu with these Add options: File, Key, Relation, Trigger, and Comment. The blue checkmark means Change the the red minus means Delete.

The field dialogs

I haven't taken a very close look at the field dialogs but for the most part these look the same as their C6 counterparts. One exception is the controls dialog, shown in Figure 7, which uses its own property toolbox (not the one normally displayed in the IDE).



Controls:

PROMPT('&Address:'), AT(0
ENTRY(@S20), AT(0,0,0,0)

Control Type: ENTRY

Color	
BackGround	□
SelBackGround	□
SelForeGround	□
TextColor	■ **ControlText**
Design	
SuppressTransparency	**True**
TabIndex	**0**
Extra	
DropID	
EntryMode	**Default**
TextCase	**Default**
General	
Justification	**Default**
Layout	Default
Picture	**@S20**
TextFont	**Microsoft Sans Serif, 10pt**
Use	**STU:Address**
Help	
Alrt	
Cursor	
HelpID	
Key	

Picture
A display picture token that specifies the input format for the data entered into the control (PROP:Text).

Figure 7. The field controls tab

In Figure 7 I've selected the Picture field, and the little ellipsis button appears. Click on this and you'll see a picture dialog such as is now in C6. As with C6 you'll have to close that dialog to continue with other work, but unlike the current version of Clarion, that kind of restriction isn't the norm in C7.

Conclusions

The C7 Dictionary Editor retains enough of C6's layout to look familiar, while significantly flattening the IDE. The extensive use of tabs makes it easy to navigate between tables, fields, keys etc., and the ability to drag tabs is a nice addition. The tab usage is also suggestive of the IDE flattening we can expect in the new AppGen.

David Harms is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995), and has written or co-written several Java books. David is a member of the American Society of Journalists and Authors (ASJA).

Reader Comments

Posted on Monday, July 16, 2007 by Bob Campbell

Hi Dave,

One thing I've always wanted is to have both an entry and a display version of a field, similar to the Column Heading and Prompt Text versions we have now.

In the Window Formatter, I want the ability to choose and/or switch a field from being an entry field to a display string, and vice versa.

There are several attributes that I have different between entry fields and display strings. One is the indent: I like to indent character entry fields, LEFT(2), but indent string displays with LEFT only.

Do you think we could have this in C7? It would save a lot of "touch-up" time.

(Is there a better/correct place to post this?)

Posted on Tuesday, July 17, 2007 by Dave Harms

Bob,

I've submitted a feature request on your behalf.

Dave

Add a comment

Clarion Magazine

Bob Z's Blog Post On Designers, With Full Size Images

Published 2007-06-15

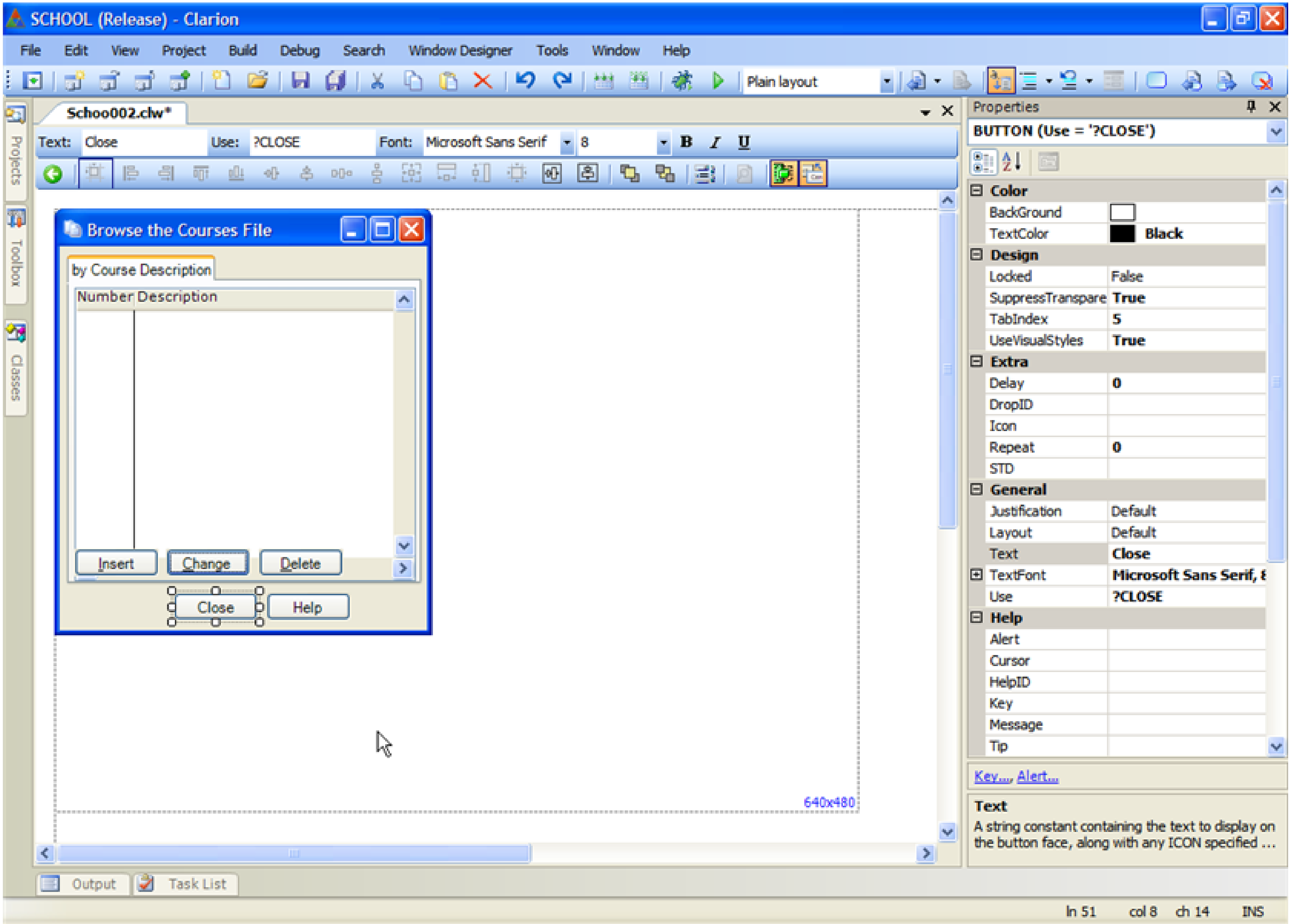
> *NOTE: The following text is taken verbatim from Bob Z's blog post. I've reproduced it here with the full size images included for easier readability. You will, however, need to view this article in printer-friendly mode to see the images in their entirety. I've also added a few comments to the text.*

Text and images by Bob Zaunere, comments by Dave Harms

Between Clarion 7 and Clarion.Net we have 5 new Designers, there are Designers for Windows, Reports, WinForms, Compact Forms, and Web Forms. Both the Window and Report Designers are new for Clarion 7. The Report Designer is shared by Clarion 7 and Clarion.Net. In other words both platforms; Win32 and .Net can use and share the same Reports.

Early on we made a design decision to ensure that all the Designers would share the same User Interface, so anyone who learns one Designer will have the same experience with any of the other Designers.

Here is a screen shot of the Window Designer for Clarion 7, and remember that it can also be used to design and modify Windows created for any version of Clarion for Windows.



Text: Close Use: ?CLOSE Font: Microsoft Sans Serif 8

BUTTON (Use = '?CLOSE')

Color	
BackGround	□
TextColor	■ Black
Design	
Locked	False
SuppressTranspare	True
TabIndex	5
UseVisualStyles	True
Extra	
Delay	0
DropID	
Icon	
Repeat	0
STD	
General	
Justification	Default
Layout	Default
Text	Close
TextFont	Microsoft Sans Serif, 8
Use	?CLOSE
Help	
Alert	
Cursor	
HelpID	
Key	
Message	
Tip	

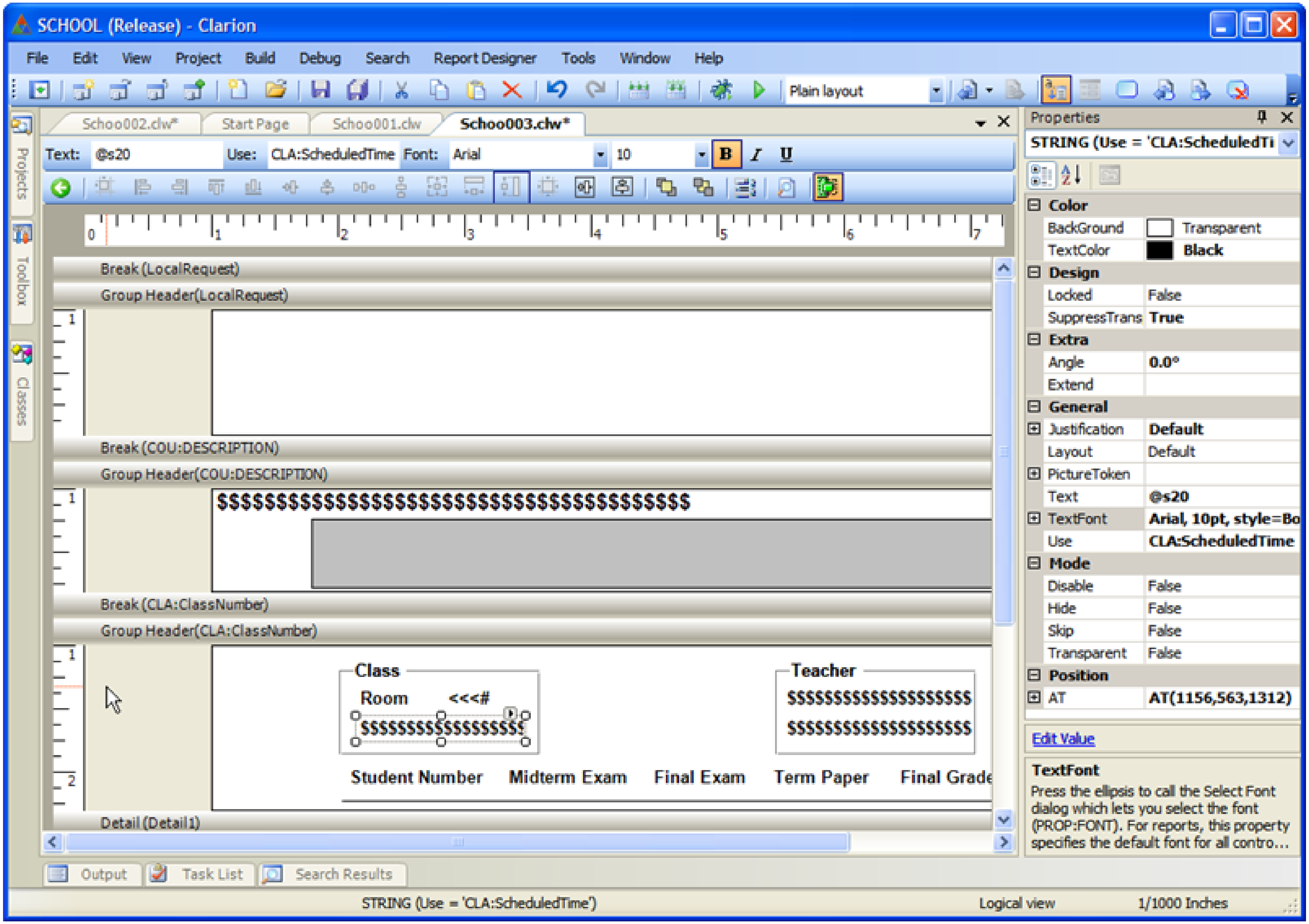
Text
A string constant containing the text to display on the button face, along with any ICON specified ...

ln 51 col 8 ch 14 INS

This is the formatter we've had in the first Alpha, and which has been quite stable.

Let's have a look at each of the other Designers, and then we can focus in on the common elements of the Designer UI.

Here's a shot of the Report Designer shared by Clarion 7 and Clarion.Net. Again, take note that the Report Designer can also be used for work on Reports created for any version of Clarion for Windows.



Text: @s20 Use: CLA:ScheduledTime Font: Arial 10

STRING (Use = 'CLA:ScheduledTi

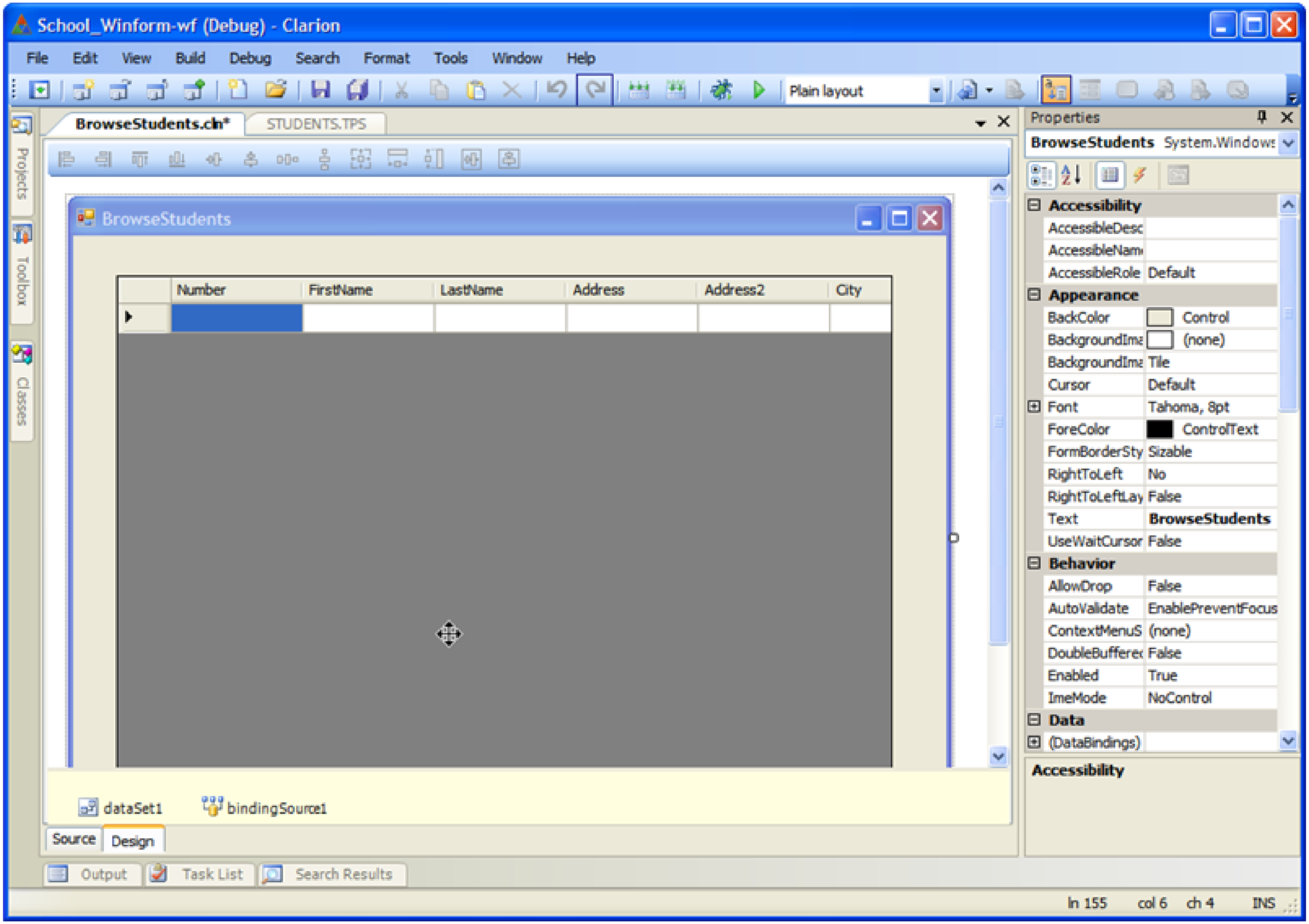
Color	
BackGround	□ Transparent
TextColor	■ Black
Design	
Locked	False
SuppressTrans	True
Extra	
Angle	0.0°
Extend	
General	
Justification	Default
Layout	Default
PictureToken	
Text	@s20
TextFont	Arial, 10pt, style=Bo
Use	CLA:ScheduledTime
Mode	
Disable	False
Hide	False
Skip	False
Transparent	False
Position	
AT	AT(1156,563,1312)

TextFont
Press the ellipsis to call the Select Font dialog which lets you select the font (PROP:FONT). For reports, this property specifies the default font for all contro...

STRING (Use = 'CLA:ScheduledTime') Logical view 1/1000 Inches

The report designer is also in the first alpha - nothing new here.

Now have a look at the WinForm for Designer.



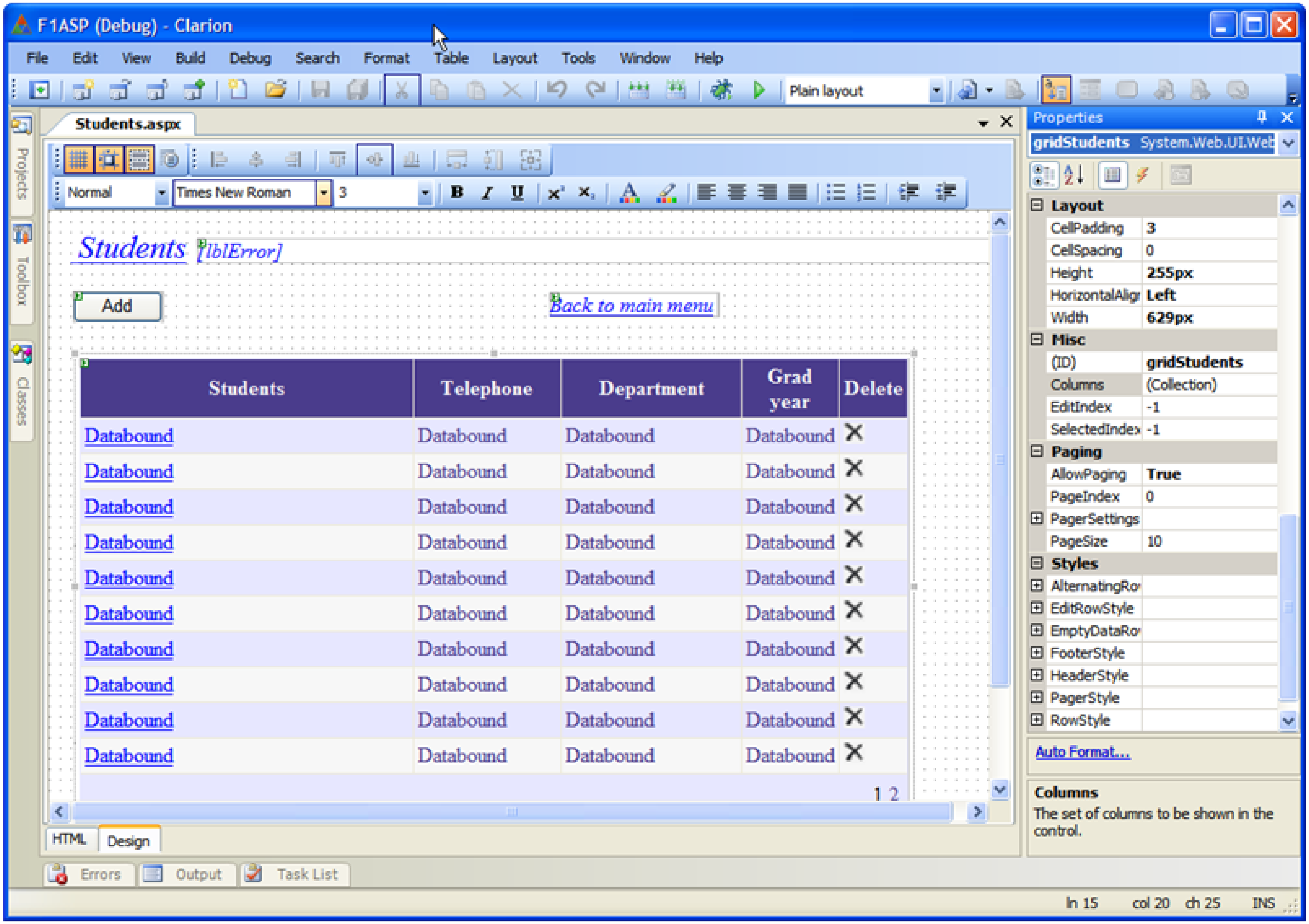
BrowseStudents System.Windows

Accessibility	
AccessibleDesc	
AccessibleNam	
AccessibleRole	Default
Appearance	
BackColor	Control
BackgroundIma	(none)
BackgroundIma	Tile
Cursor	Default
Font	Tahoma, 8pt
ForeColor	ControlText
FormBorderSty	Sizable
RightToLeft	No
RightToLeftLay	False
Text	**BrowseStudents**
UseWaitCursor	False
Behavior	
AllowDrop	False
AutoValidate	EnablePreventFocus
ContextMenuS	(none)
DoubleBuffere	False
Enabled	True
ImeMode	NoControl
Data	
(DataBindings)	

Accessibility

This is now new territory. The browse control looks different from what we're used to presumably because it's a native . NET control, although as Z indicated at the Oz DevCon SV has added the kind of functionality Clarion developers have come to expect in a browse, including page loading.

And here is a shot of the ASP.Net Web Form Designer.



Students [lblError]

Add

Back to main menu

Students	Telephone	Department	Grad year	Delete
Databound	Databound	Databound	Databound	✕
Databound	Databound	Databound	Databound	✕
Databound	Databound	Databound	Databound	✕
Databound	Databound	Databound	Databound	✕
Databound	Databound	Databound	Databound	✕
Databound	Databound	Databound	Databound	✕
Databound	Databound	Databound	Databound	✕
Databound	Databound	Databound	Databound	✕
Databound	Databound	Databound	Databound	✕
Databound	Databound	Databound	Databound	✕

1 2

Properties

gridStudents System.Web.UI.Web

Layout	
CellPadding	3
CellSpacing	0
Height	255px
HorizontalAlig	Left
Width	629px
Misc	
(ID)	gridStudents
Columns	(Collection)
EditIndex	-1
SelectedIndex	-1
Paging	
AllowPaging	True
PageIndex	0
PagerSettings	
PageSize	10
Styles	
AlternatingRo	
EditRowStyle	
EmptyDataRo	
FooterStyle	
HeaderStyle	
PagerStyle	
RowStyle	

Auto Format...

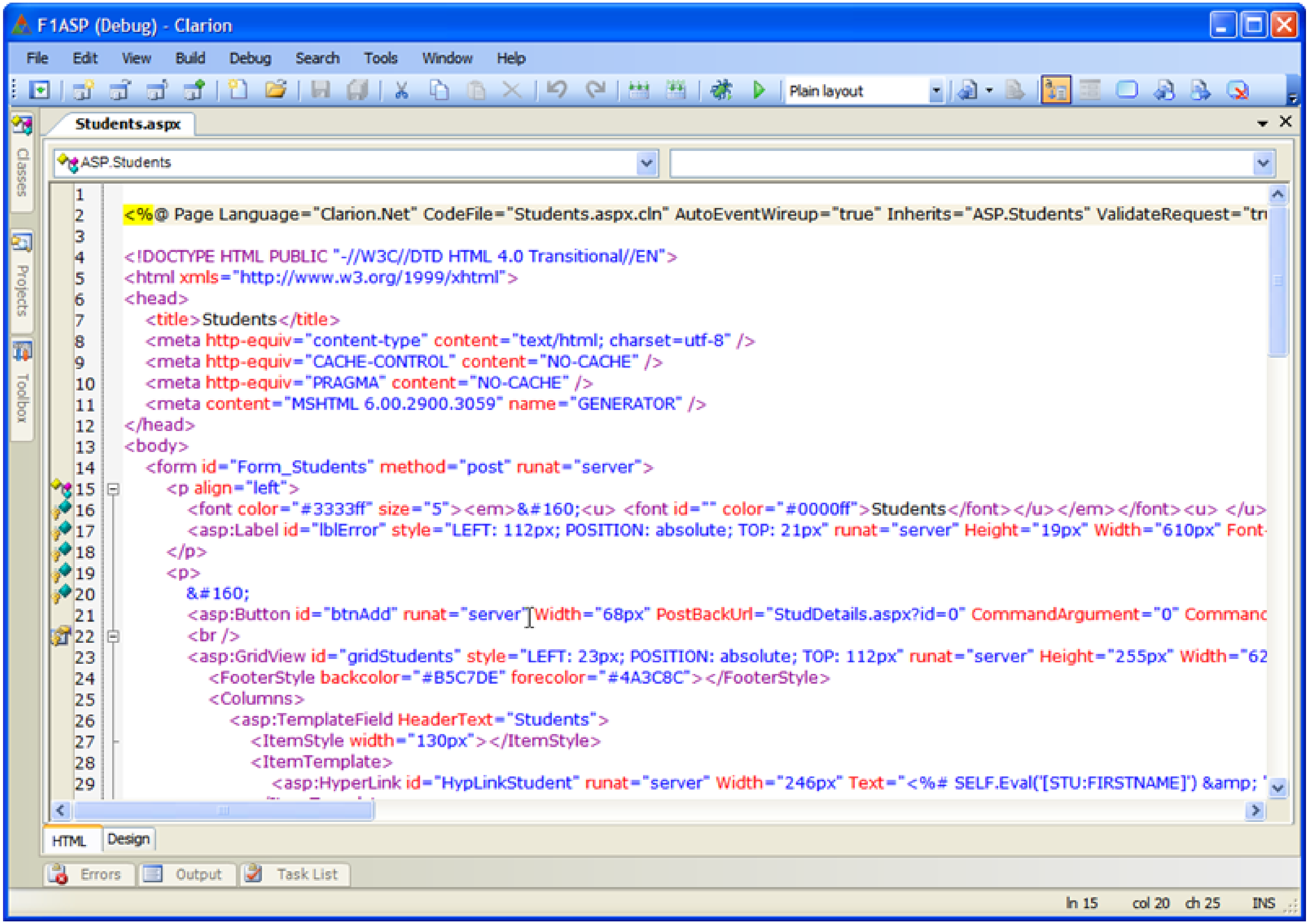
Columns
The set of columns to be shown in the control.

As Russ indicated in his DevCon coverage, the WebForm designer lets you see the page as it will appear in the browser. The code below shows the use of absolute positioning to achieve this.

The ASP.Net Designer has three possible Views, note the Tab text; HTM and Design. The screen shot above shows the

Design view, but you can also work directly with the HTML that the Designer creates for you.

Here is the ASP.Net HTML view

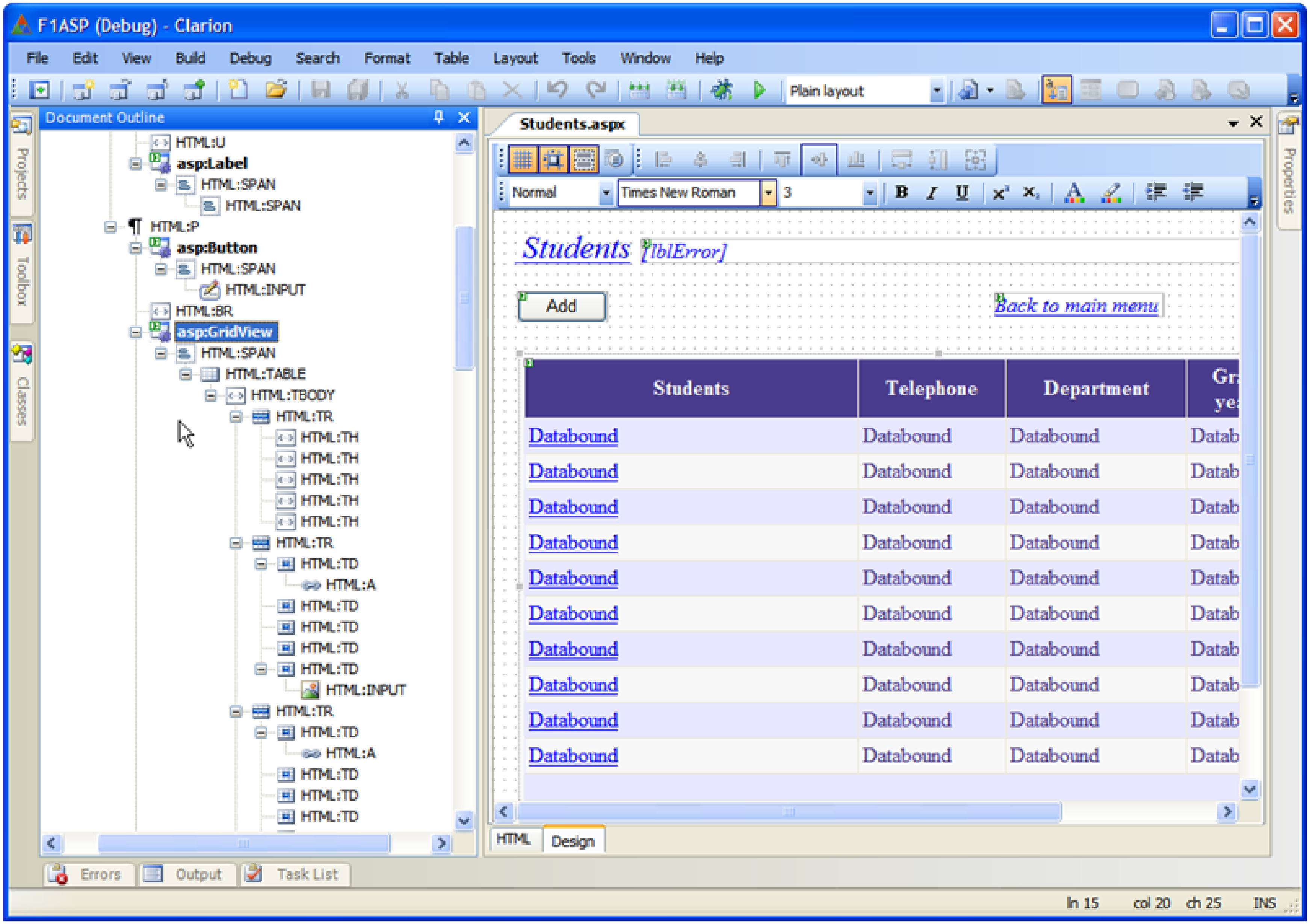


```

<%@ Page Language="Clarion.Net" CodeFile="Students.aspx.cln" AutoEventWireup="true" Inherits="ASP.Students" ValidateRequest="tr

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmls="http://www.w3.org/1999/xhtml">
<head>
   <title>Students</title>
   <meta http-equiv="content-type" content="text/html; charset=utf-8" />
   <meta http-equiv="CACHE-CONTROL" content="NO-CACHE" />
   <meta http-equiv="PRAGMA" content="NO-CACHE" />
   <meta content="MSHTML 6.00.2900.3059" name="GENERATOR" />
</head>
<body>
   <form id="Form_Students" method="post" runat="server">
      <p align="left">
         <font color="#3333ff" size="5"><em>&#160;<u> <font id="" color="#0000ff">Students</font></u></em></font><u> </u>
         <asp:Label id="lblError" style="LEFT: 112px; POSITION: absolute; TOP: 21px" runat="server" Height="19px" Width="610px" Font-
      </p>
      <p>
         &#160;
         <asp:Button id="btnAdd" runat="server" Width="68px" PostBackUrl="StudDetails.aspx?id=0" CommandArgument="0" Commanc
         <br />
         <asp:GridView id="gridStudents" style="LEFT: 23px; POSITION: absolute; TOP: 112px" runat="server" Height="255px" Width="62
            <FooterStyle backcolor="#B5C7DE" forecolor="#4A3C8C"></FooterStyle>
            <Columns>
               <asp:TemplateField HeaderText="Students">
                  <ItemStyle width="130px"></ItemStyle>
                  <ItemTemplate>
                     <asp:HyperLink id="HypLinkStudent" runat="server" Width="246px" Text="<%# SELF.Eval('[STU:FIRSTNAME]') &amp; '
```

And finally we can also look at an ASP.Net WebForm in Outline view, as shown here

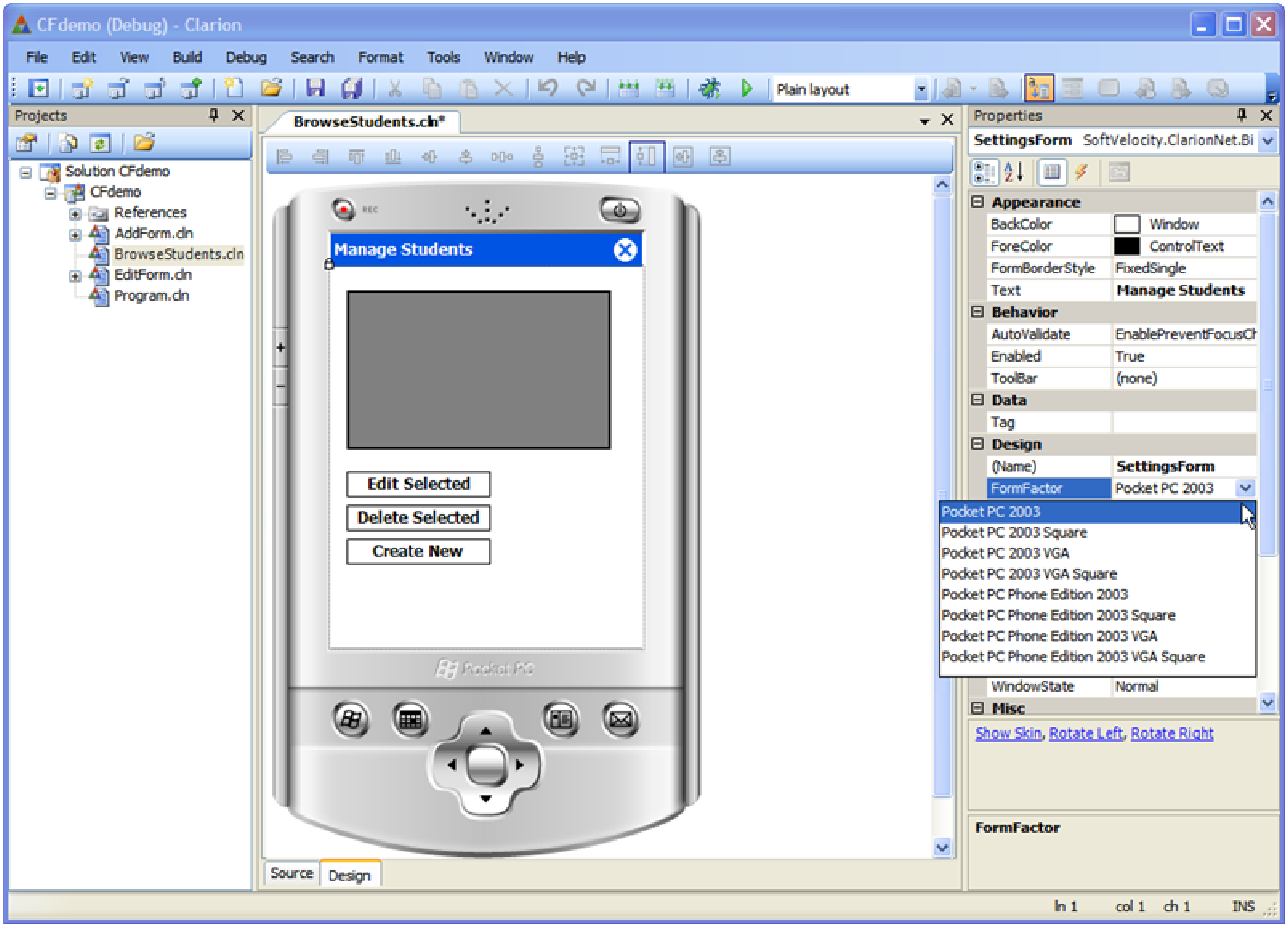


Students	Telephone	Department	Gr ye
Databound	Databound	Databound	Datab
Databound	Databound	Databound	Datab
Databound	Databound	Databound	Datab
Databound	Databound	Databound	Datab
Databound	Databound	Databound	Datab
Databound	Databound	Databound	Datab
Databound	Databound	Databound	Datab
Databound	Databound	Databound	Datab
Databound	Databound	Databound	Datab
Databound	Databound	Databound	Datab

> *Now I'm curious to see what appears when I click on one of the sub-nodes. If you've ever used FireFox's DOM Inspector you'll immediately recognize the usefulness of this view.*

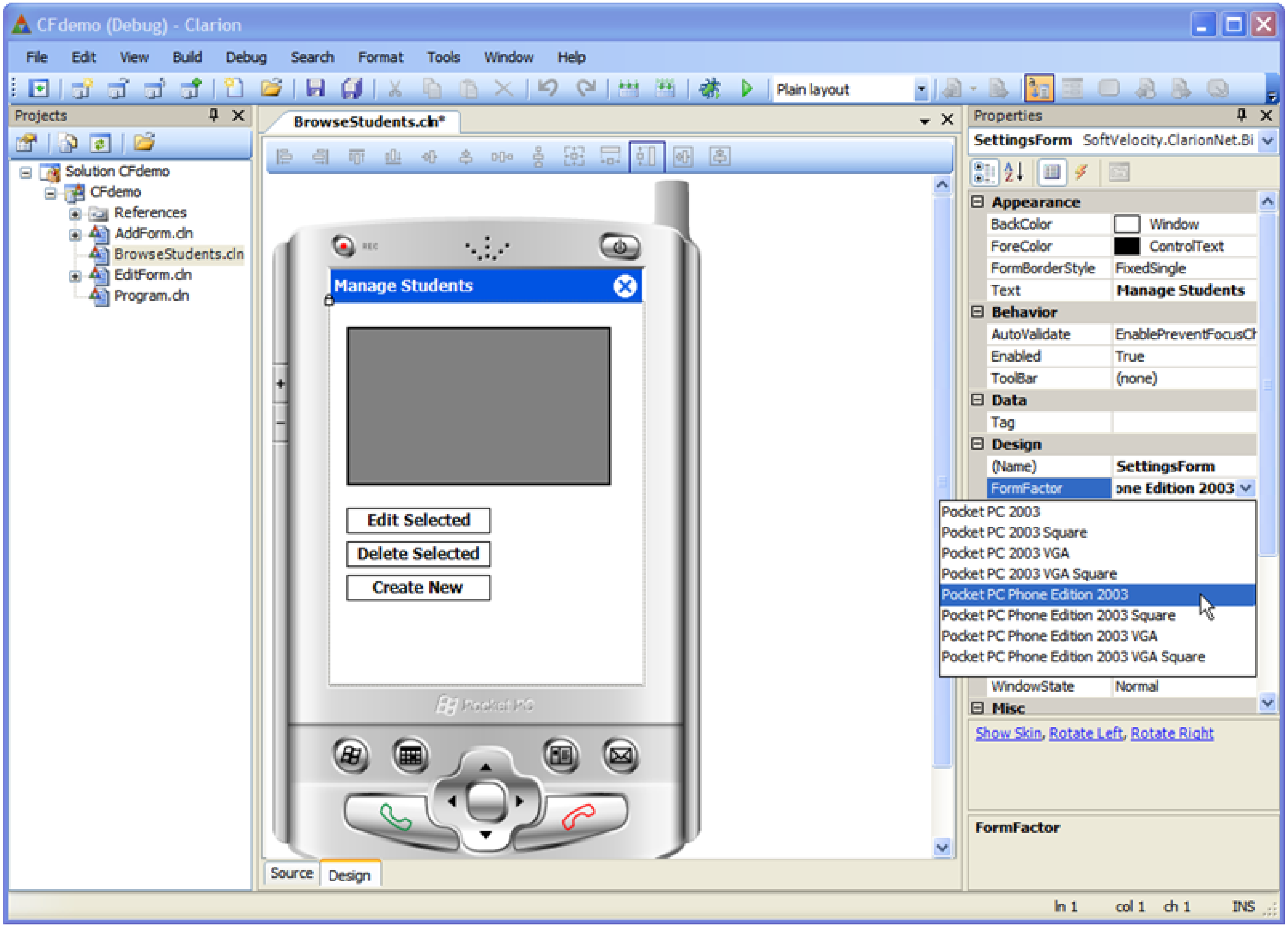
Now we'll take a look at a few screen shots of the Compact Forms Designer. As you can see the Designer can display the form sized for the target device and with a skinned representation of the device. Just select the desired FormFactor from the dropdown list.

PocketPC 2003



Appearance	
BackColor	Window
ForeColor	ControlText
FormBorderStyle	FixedSingle
Text	Manage Students
Behavior	
AutoValidate	EnablePreventFocusCh
Enabled	True
ToolBar	(none)
Data	
Tag	
Design	
(Name)	SettingsForm
FormFactor	Pocket PC 2003
WindowState	Normal
Misc	

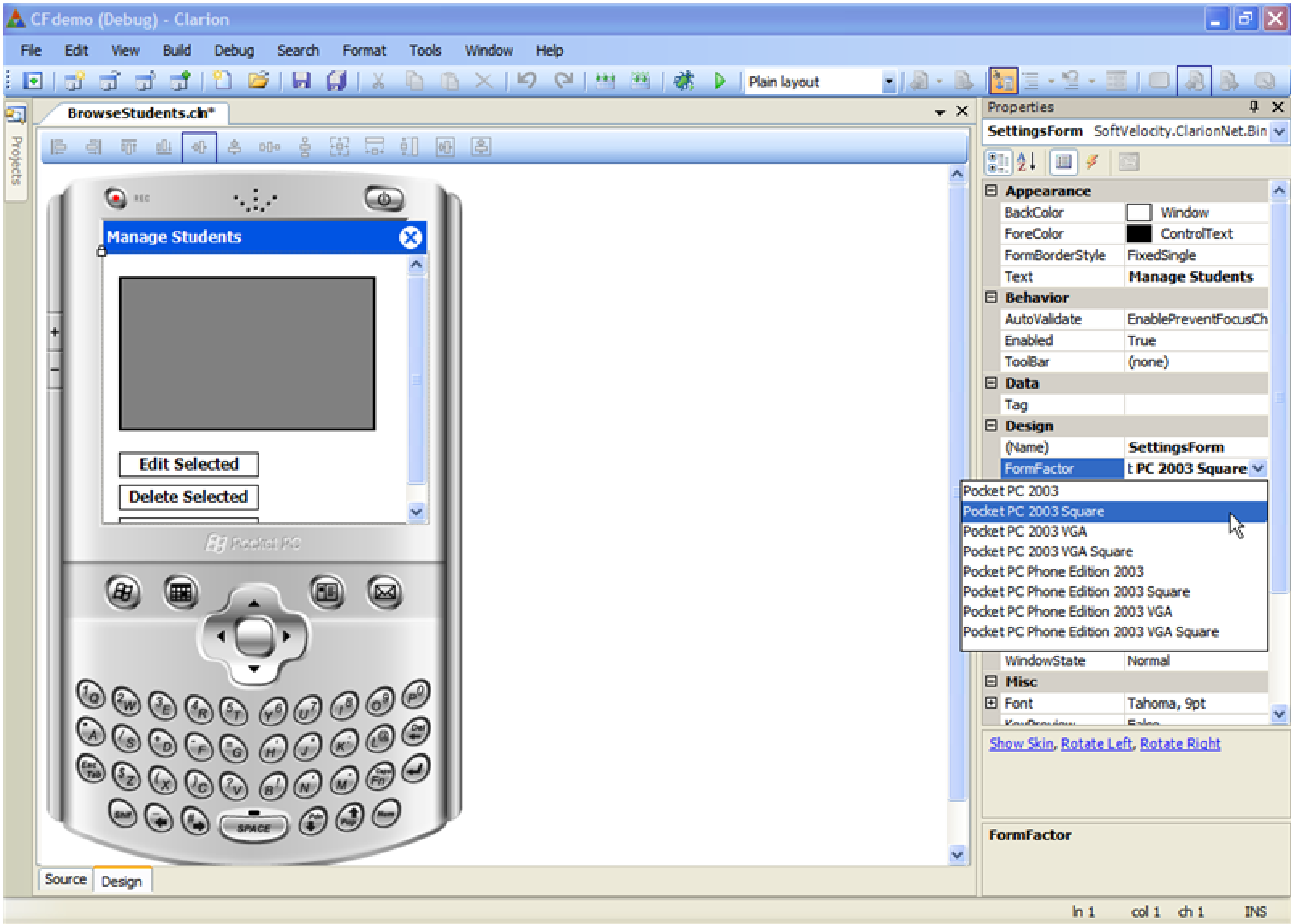
Pocket PC Phone Edition 2003



SettingsForm SoftVelocity.ClarionNet.Bi

BackColor	Window
ForeColor	ControlText
FormBorderStyle	FixedSingle
Text	Manage Students
AutoValidate	EnablePreventFocusCh
Enabled	True
ToolBar	(none)
Tag	
(Name)	SettingsForm
FormFactor	one Edition 2003
WindowState	Normal

Pocket PC 2003 Square



SettingsForm SoftVelocity.ClarionNet.Bin

Appearance
BackColor Window
ForeColor ControlText
FormBorderStyle FixedSingle
Text Manage Students

Behavior
AutoValidate EnablePreventFocusCh
Enabled True
ToolBar (none)

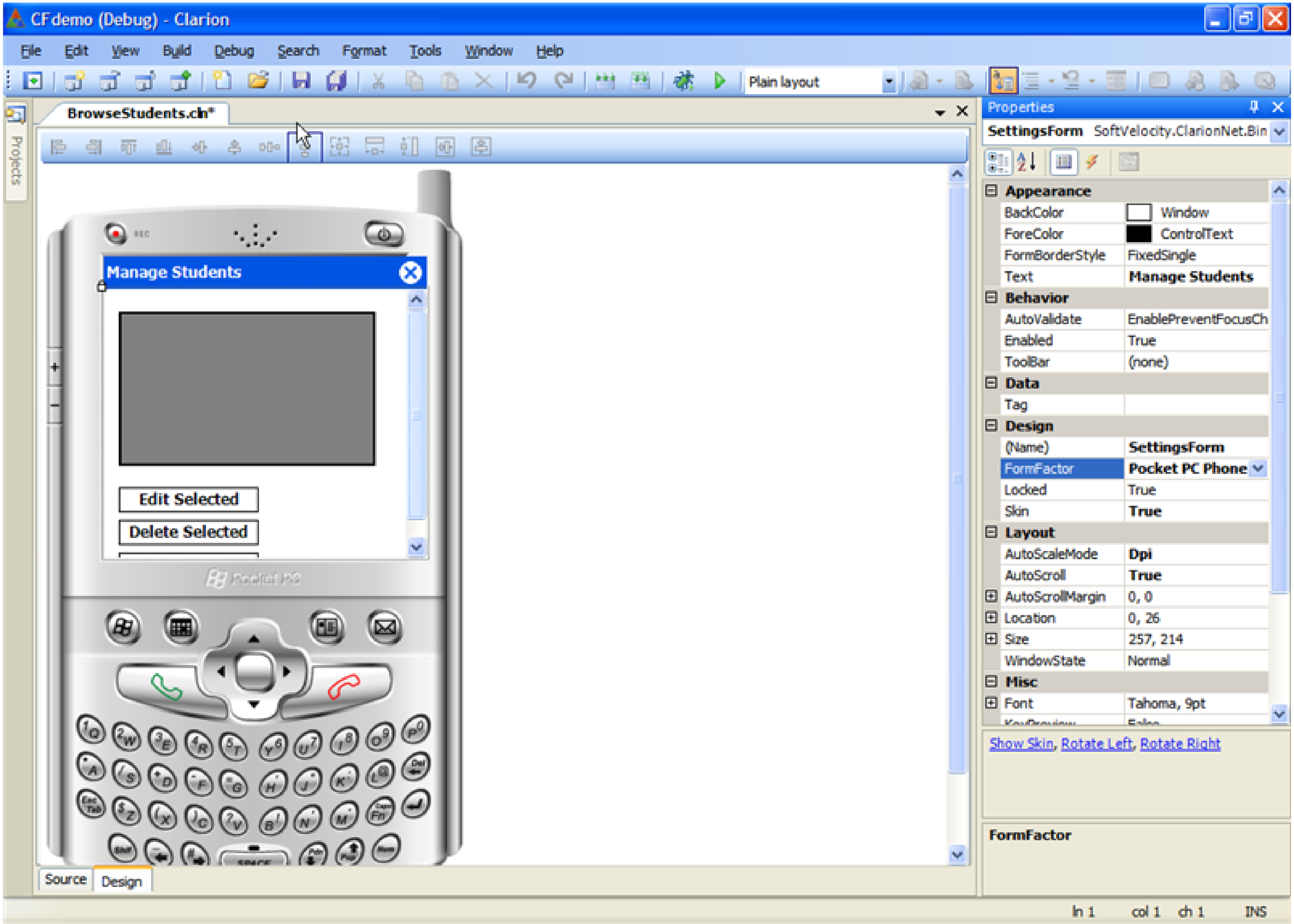
Data
Tag

Design
(Name) SettingsForm
FormFactor t PC 2003 Square
WindowState Normal

Misc
Font Tahoma, 9pt

FormFactor

Pocket PC Phone Edition 2003 Square



SettingsForm SoftVelocity.ClarionNet.Bin

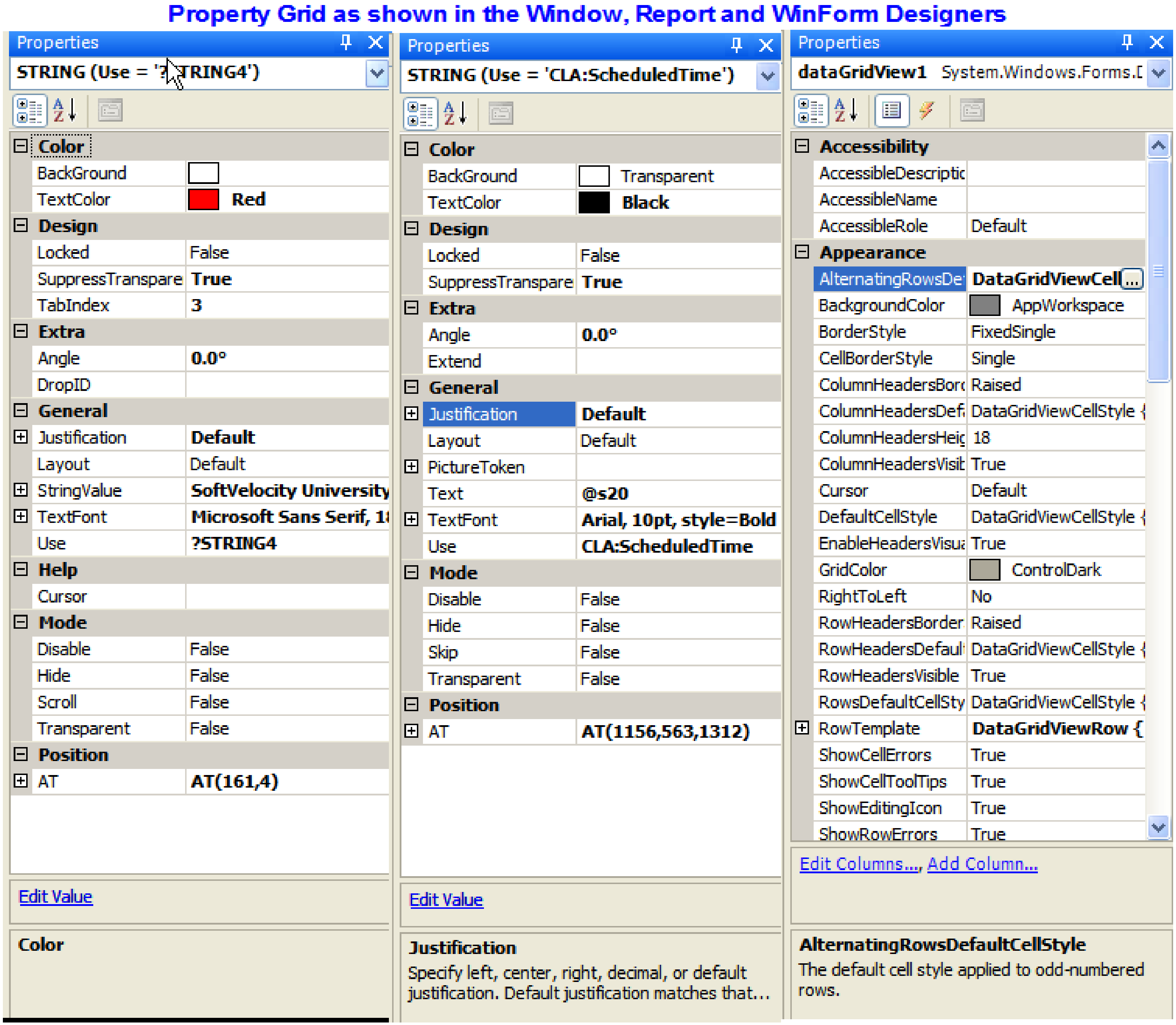
Appearance	
BackColor	Window
ForeColor	ControlText
FormBorderStyle	FixedSingle
Text	Manage Students
Behavior	
AutoValidate	EnablePreventFocusCh
Enabled	True
ToolBar	(none)
Data	
Tag	
Design	
(Name)	SettingsForm
FormFactor	Pocket PC Phone
Locked	True
Skin	True
Layout	
AutoScaleMode	Dpi
AutoScroll	True
AutoScrollMargin	0, 0
Location	0, 26
Size	257, 214
WindowState	Normal
Misc	
Font	Tahoma, 9pt
KeyPreview	False

Show Skin, Rotate Left, Rotate Right

FormFactor

> *The compact framework support in the first release of Clarion.NET is terrific news, and I know a lot of Clarion developers are anxious to take this product out for a spin.*

Now that we've seen all five Designers let's review the common UI they all share. The most obvious piece is the Properties Grid which is common to all the Designers, shown here side-by-side



Property Grid as shown in the Window, Report and WinForm Designers

Properties

STRING (Use = '?STRING4')

⊟ **Color**	
BackGround	□
TextColor	🟥 **Red**
⊟ **Design**	
Locked	False
SuppressTranspare	**True**
TabIndex	**3**
⊟ **Extra**	
Angle	**0.0°**
DropID	
⊟ **General**	
⊞ Justification	**Default**
Layout	Default
⊞ StringValue	**SoftVelocity University**
⊞ TextFont	**Microsoft Sans Serif, 1**
Use	**?STRING4**
⊟ **Help**	
Cursor	
⊟ **Mode**	
Disable	False
Hide	False
Scroll	False
Transparent	False
⊟ **Position**	
⊞ AT	**AT(161,4)**

Edit Value

Color

Properties

STRING (Use = 'CLA:ScheduledTime')

⊟ **Color**	
BackGround	□ Transparent
TextColor	■ **Black**
⊟ **Design**	
Locked	False
SuppressTranspare	**True**
⊟ **Extra**	
Angle	**0.0°**
Extend	
⊟ **General**	
⊞ Justification	**Default**
Layout	Default
⊞ PictureToken	
Text	**@s20**
⊞ TextFont	**Arial, 10pt, style=Bold**
Use	**CLA:ScheduledTime**
⊟ **Mode**	
Disable	False
Hide	False
Skip	False
Transparent	False
⊟ **Position**	
⊞ AT	**AT(1156,563,1312)**

Edit Value

Justification

Specify left, center, right, decimal, or default justification. Default justification matches that...

Properties

dataGridView1 System.Windows.Forms.D

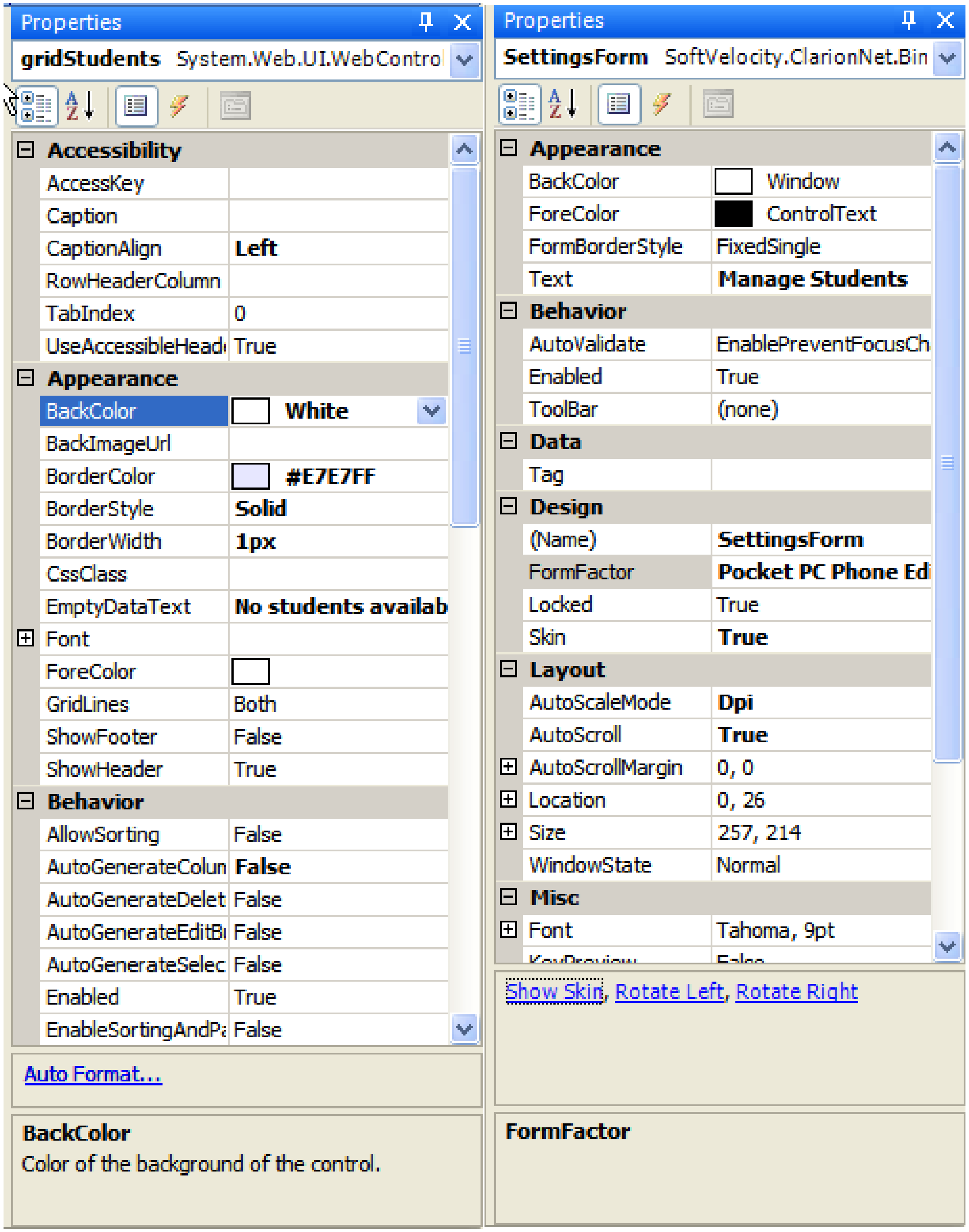
⊟ **Accessibility**	
AccessibleDescriptic	
AccessibleName	
AccessibleRole	Default
⊟ **Appearance**	
AlternatingRowsDe	**DataGridViewCell** ...
BackgroundColor	▨ AppWorkspace
BorderStyle	FixedSingle
CellBorderStyle	Single
ColumnHeadersBorc	Raised
ColumnHeadersDef	DataGridViewCellStyle {
ColumnHeadersHeic	18
ColumnHeadersVisit	True
Cursor	Default
DefaultCellStyle	DataGridViewCellStyle {
EnableHeadersVisua	True
GridColor	▨ ControlDark
RightToLeft	No
RowHeadersBorder	Raised
RowHeadersDefaul	DataGridViewCellStyle {
RowHeadersVisible	True
RowsDefaultCellSty	DataGridViewCellStyle {
⊞ RowTemplate	**DataGridViewRow {**
ShowCellErrors	True
ShowCellToolTips	True
ShowEditingIcon	True
ShowRowErrors	True

Edit Columns..., Add Column...

AlternatingRowsDefaultCellStyle

The default cell style applied to odd-numbered rows.

Property Grid in the ASP.Net and Compact Form Designers



Properties

gridStudents System.Web.UI.WebContro

⊟ **Accessibility**	
AccessKey	
Caption	
CaptionAlign	**Left**
RowHeaderColumn	
TabIndex	0
UseAccessibleHead	True
⊟ **Appearance**	
BackColor	□ **White**
BackImageUrl	
BorderColor	□ **#E7E7FF**
BorderStyle	**Solid**
BorderWidth	**1px**
CssClass	
EmptyDataText	**No students availab**
⊞ Font	
ForeColor	□
GridLines	Both
ShowFooter	False
ShowHeader	True
⊟ **Behavior**	
AllowSorting	False
AutoGenerateColun	**False**
AutoGenerateDelet	False
AutoGenerateEditBı	False
AutoGenerateSelec	False
Enabled	True
EnableSortingAndPa	False

Auto Format...

BackColor
Color of the background of the control.

Properties

SettingsForm SoftVelocity.ClarionNet.Bin

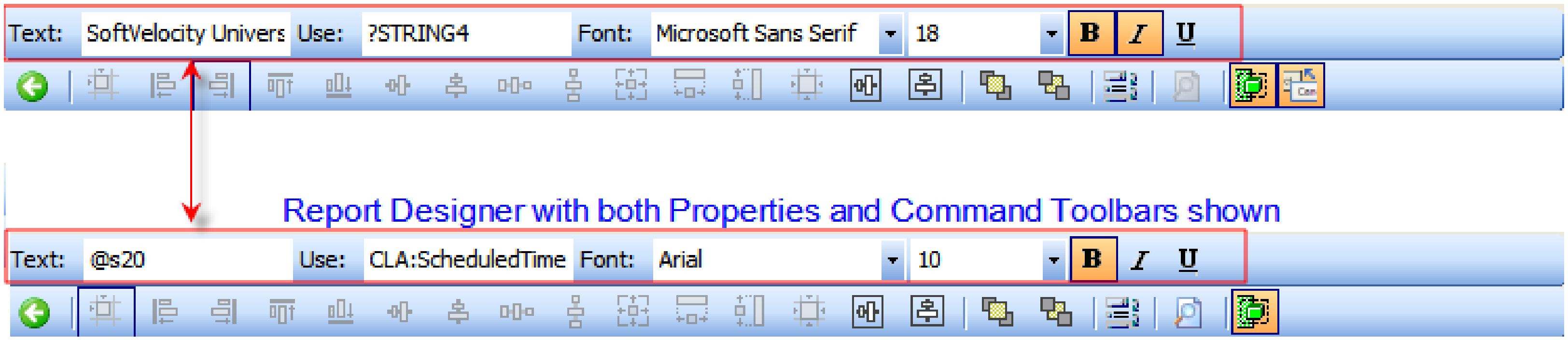
⊟ **Appearance**	
BackColor	□ Window
ForeColor	■ ControlText
FormBorderStyle	FixedSingle
Text	**Manage Students**
⊟ **Behavior**	
AutoValidate	EnablePreventFocusCh
Enabled	True
ToolBar	(none)
⊟ **Data**	
Tag	
⊟ **Design**	
(Name)	**SettingsForm**
FormFactor	**Pocket PC Phone Ed**
Locked	True
Skin	**True**
⊟ **Layout**	
AutoScaleMode	**Dpi**
AutoScroll	**True**
⊞ AutoScrollMargin	0, 0
⊞ Location	0, 26
⊞ Size	257, 214
WindowState	Normal
⊟ **Misc**	
⊞ Font	Tahoma, 9pt
KeyPreview	False

Show Skin, Rotate Left, Rotate Right

FormFactor

The Designer Toolbars also share almost the exact same UI, the differences are that the Window and Report Designers take a trick from Clarion 6, and can optionally have the most common properties available for editing right on the Properties Toolbar, as shown here in the red boxes.

Window Designer with both Properties and Command Toolbars shown

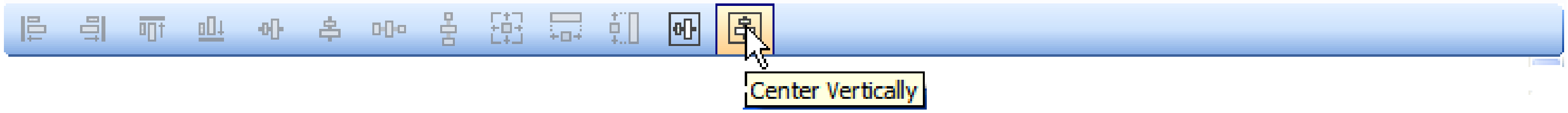


Text: SoftVelocity Univers Use: ?STRING4 Font: Microsoft Sans Serif 18

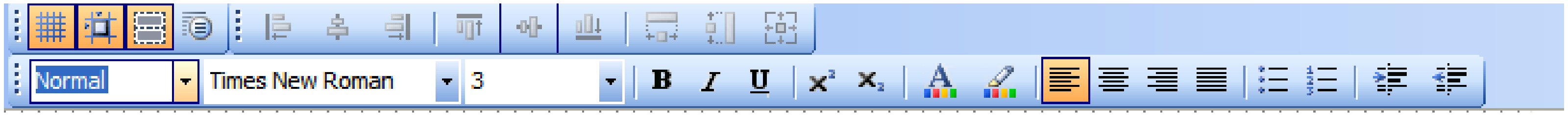
Report Designer with both Properties and Command Toolbars shown

Text: @s20 Use: CLA:ScheduledTime Font: Arial 10

WinForm Designer with Command Toolbar

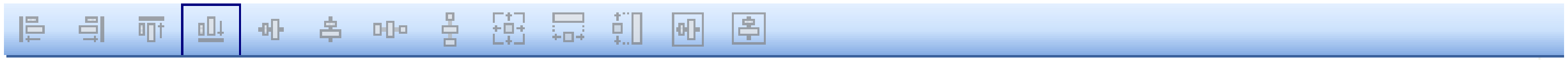


ASP.Net Designer with Properties and Command Toolbar



Normal Times New Roman 3

Compact Form Designer with Command Toolbar



As you can see, learn them once and you're ready to use any of the Designers. There are a lot of cool features in the Designers, one of them that I'm sure you'll really like using is the “Snap Lines” Grid option. It's impossible to get a screen shot of that feature so here is a very short video. I'll post a new article and a longer video tomorrow that goes a lot deeper into the new Designers.

Reader Comments

Posted on Tuesday, August 28, 2007 by Stephen Ryan

blown away, very nice, nice, nice.

Add a comment

Clarion Magazine

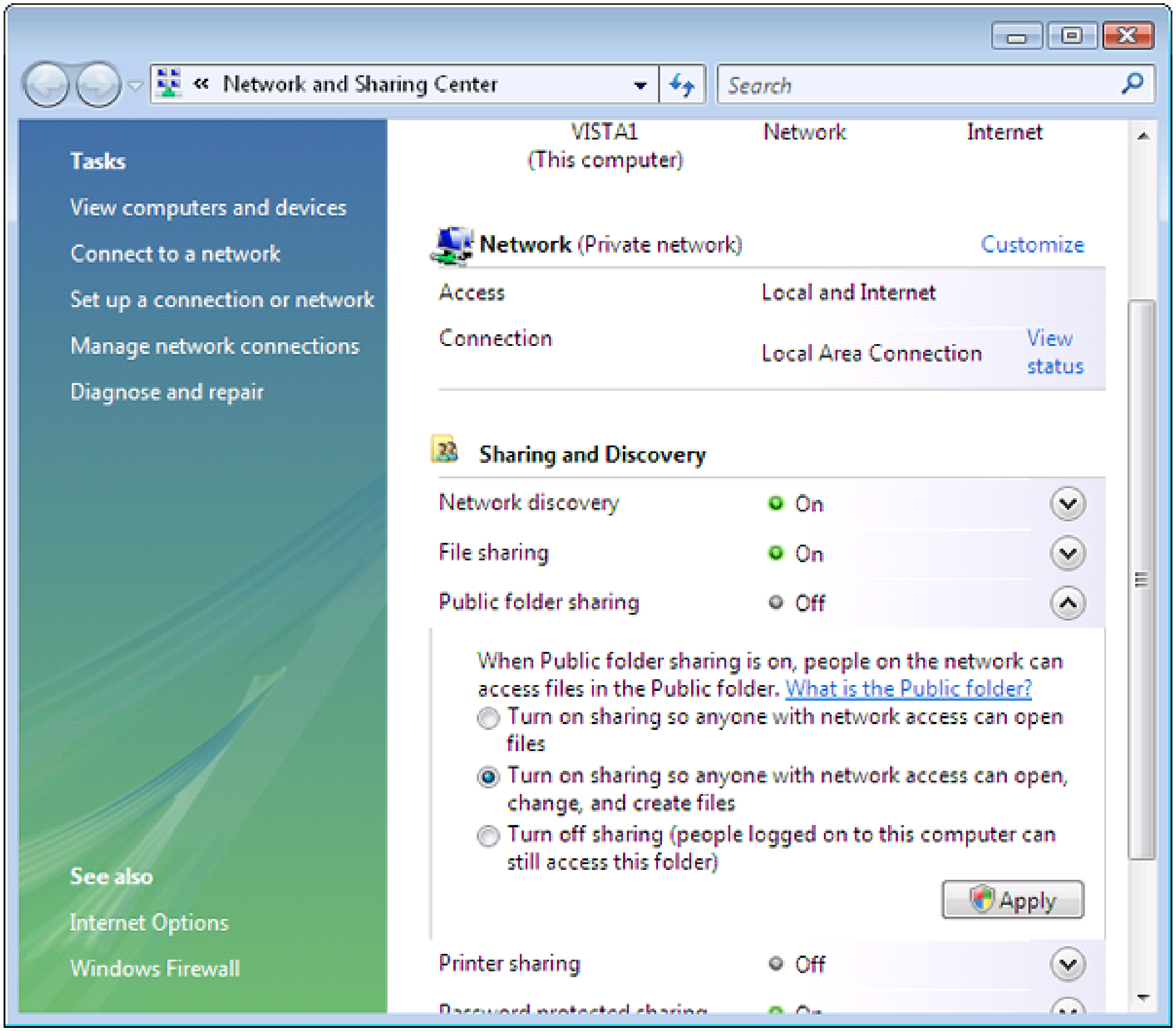
Coping With Vista - Creating Directory Shares

by Jane Fleming

Published 2007-08-07

In Part 1 of this article I discussed Vista's Public folder as a potential location for shared application data, and I showed how to set up network sharing. Now it's time to create a directory share, but as with many things Vista that's a bit more complicated than it should be.

Let's first see what happens when I turn on Public folder sharing in the Network and Sharing Center. You can see I have rather dumbed-down choices: read-only access for everybody with an account or read/write access (Figure 13).



Network (Private network)

Access Local and Internet

Connection Local Area Connection

Sharing and Discovery

Network discovery ● On

File sharing ● On

Public folder sharing ● Off

When Public folder sharing is on, people on the network can access files in the Public folder. What is the Public folder?

○ Turn on sharing so anyone with network access can open files

● Turn on sharing so anyone with network access can open, change, and create files

○ Turn off sharing (people logged on to this computer can still access this folder)

Printer sharing ● Off

Password protected sharing ● On

Figure 13. Public folder sharing choices

Looking at my Vista machine from an XP computer, after enabling Public folder sharing I see that (as I'd expect) the entire Public folder structure is now shared (Figure 14).

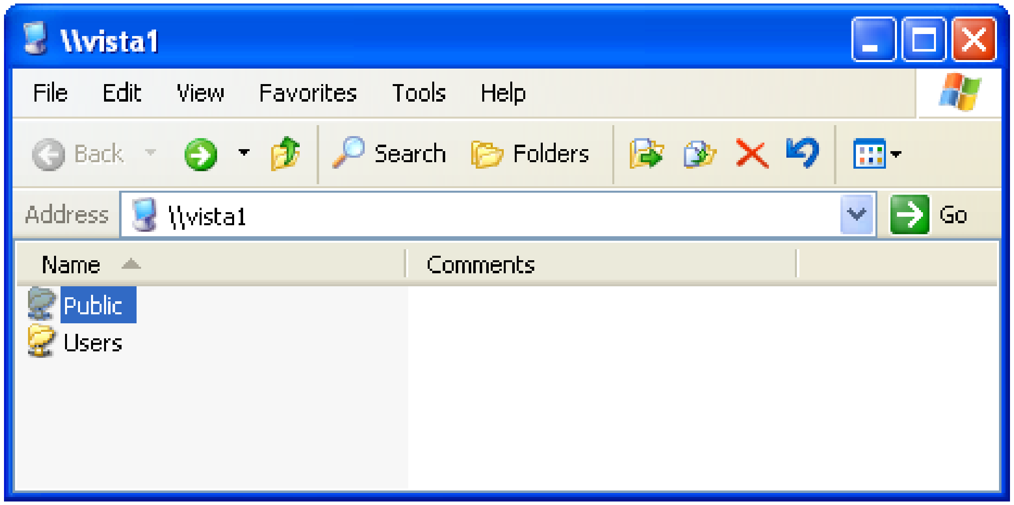
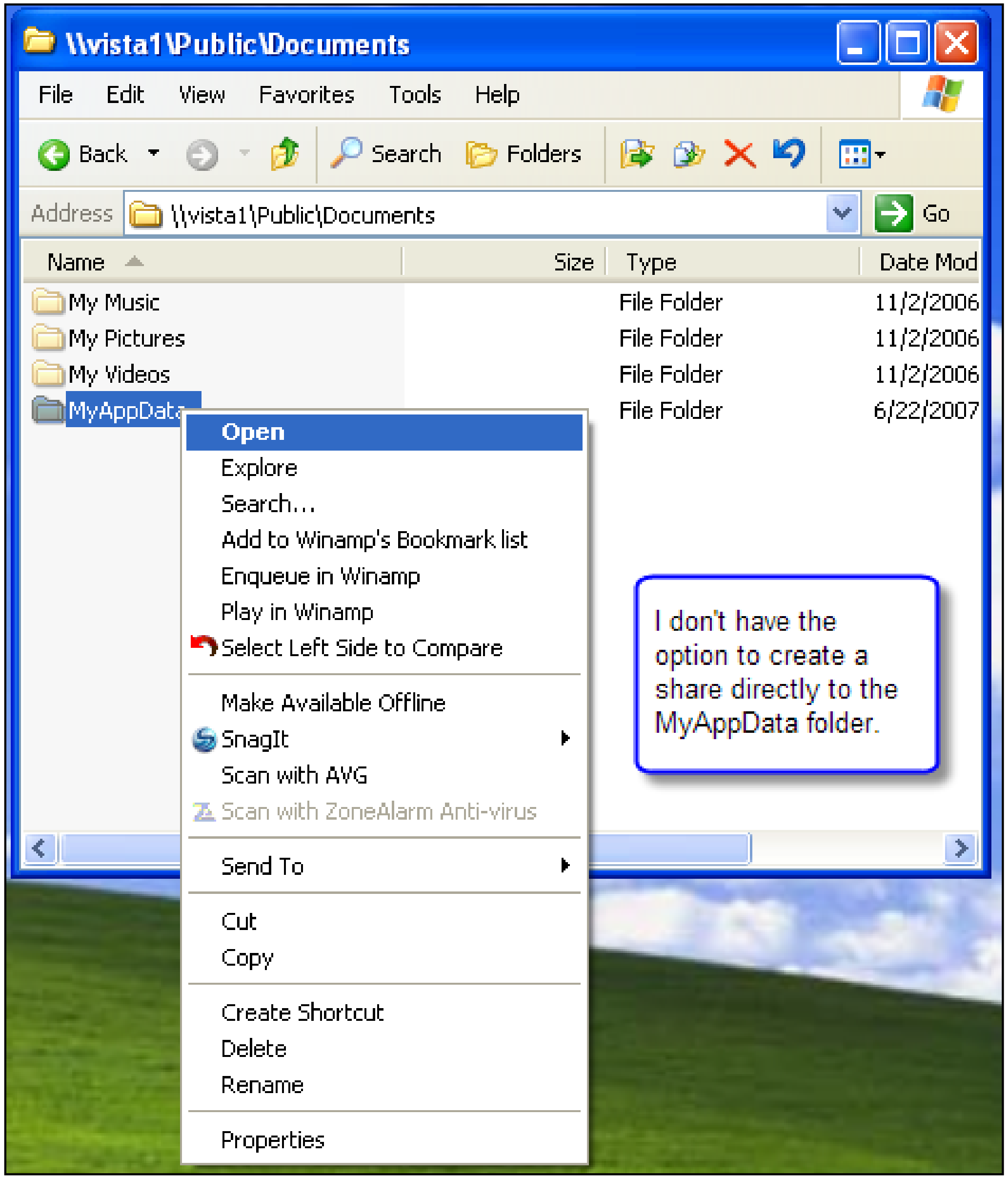


Figure 14. Viewing the shared Public folder from another computer

I created a folder on my Vista computer called MyAppData. I can browse to that from my XP machine by double-clicking my way through Public and then Documents (Figure 15). But as Figure 15 shows, I can't map a network drive directly to that folder so that may complicate my program's data access.



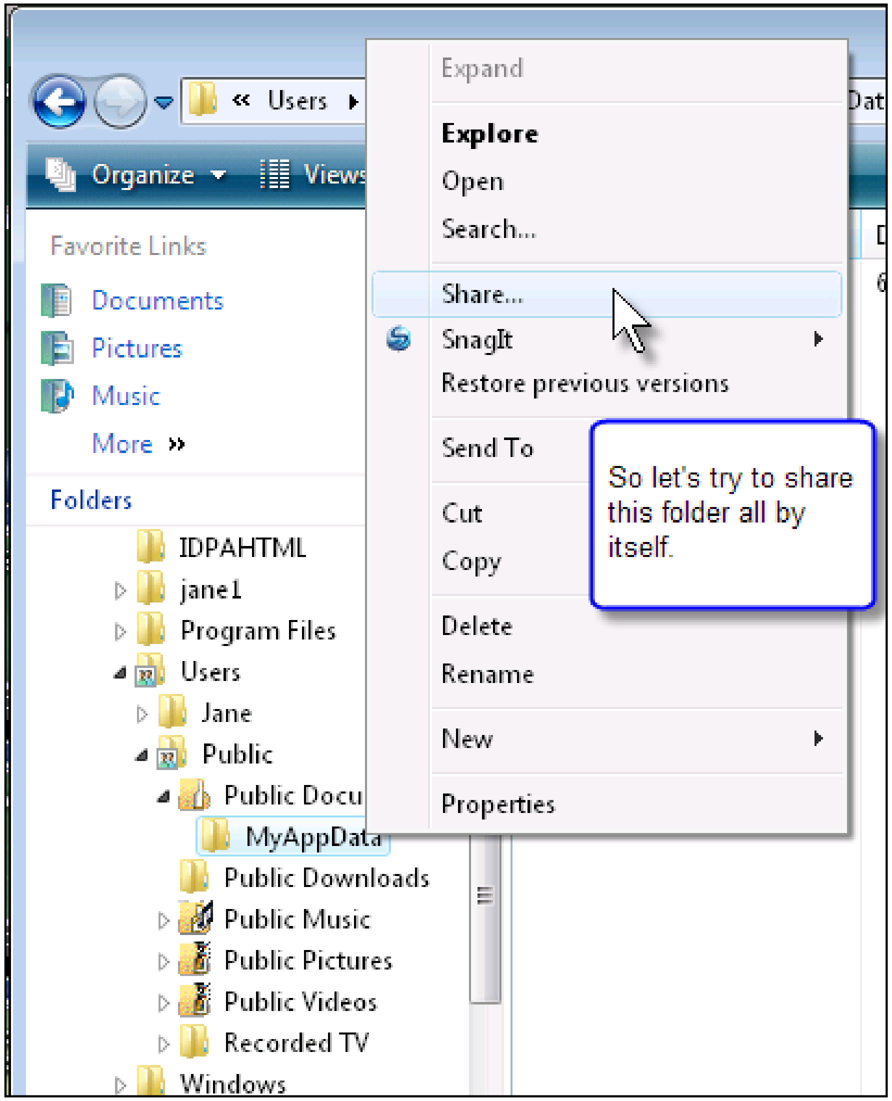
Address: \\vista1\Public\Documents

Name	Size	Type	Date Mod
My Music		File Folder	11/2/2006
My Pictures		File Folder	11/2/2006
My Videos		File Folder	11/2/2006
MyAppData		File Folder	6/22/2007

I don't have the option to create a share directly to the MyAppData folder.

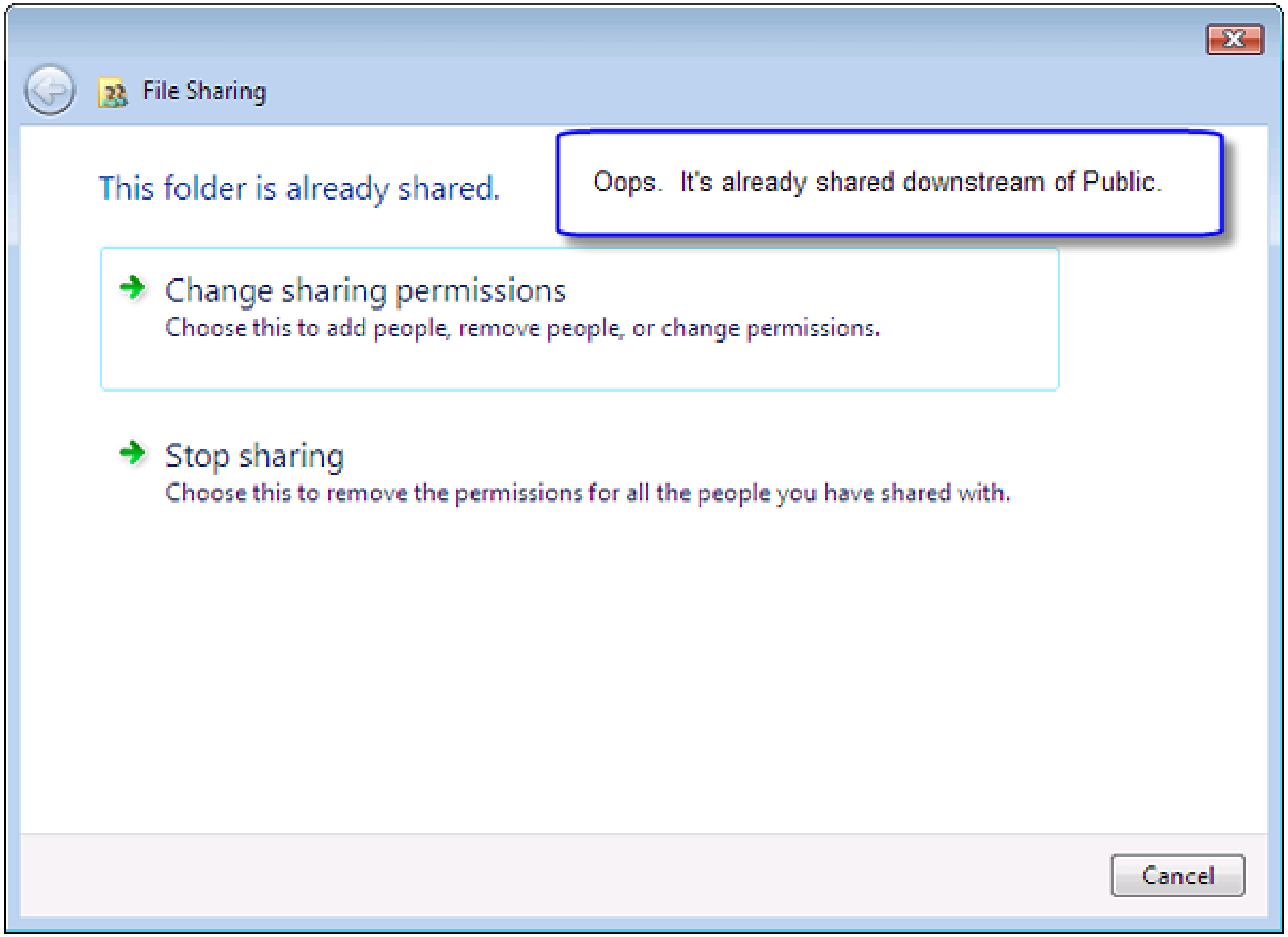
Figure 15. The MyAppData folder I created for my app's data

Obviously I need to create a share on the MyAppData folder itself. This should be simple, right? Well, if I right-click the folder on my Vista machine and click Share (Figure 16), the sharing wizard tells me it's already shared way down the food chain from the actual Public share (Figure 17).



So let's try to share this folder all by itself.

Figure 16. So I'll share my folder all by itself



This folder is already shared.

Oops. It's already shared downstream of Public.

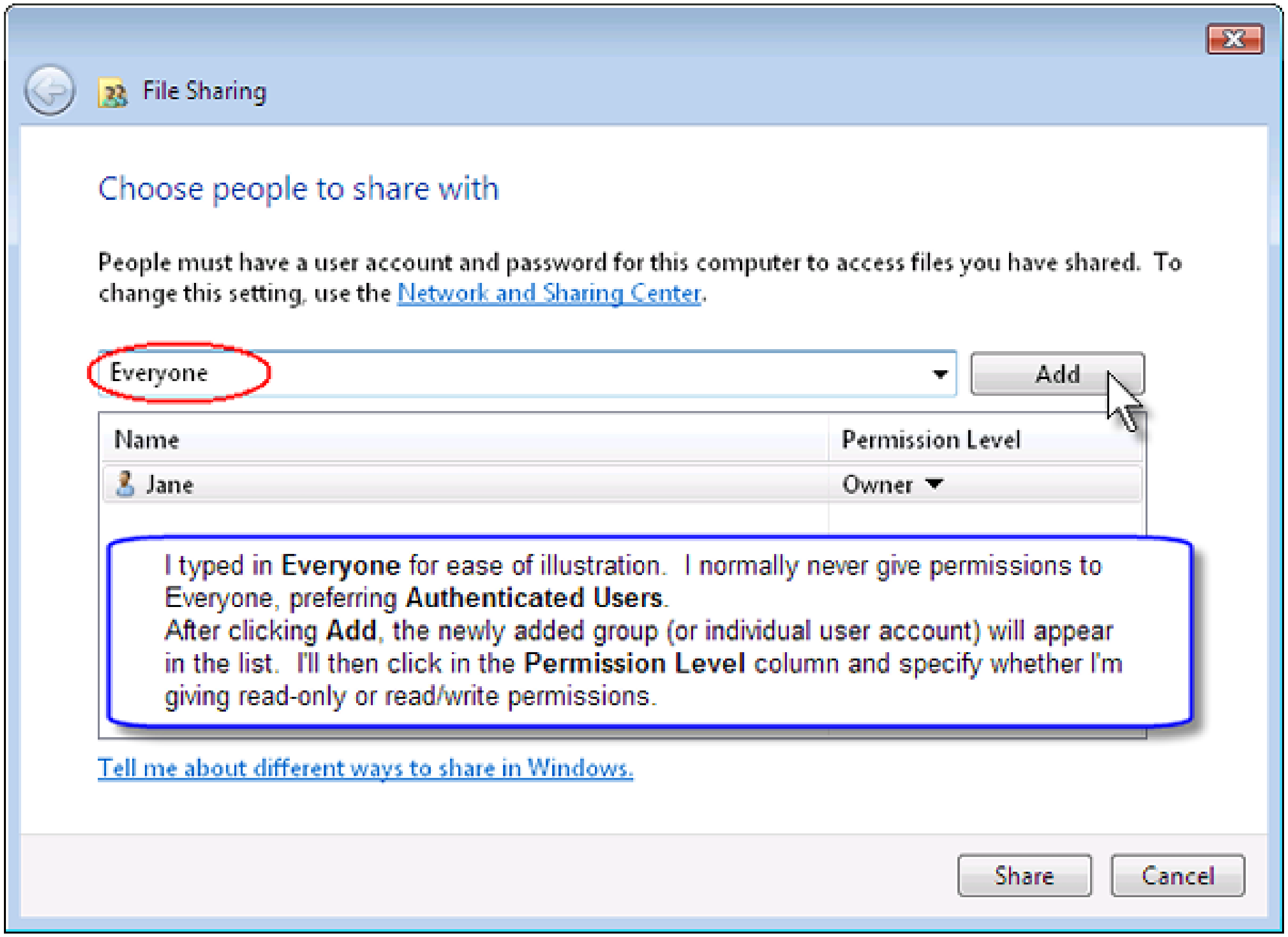
Change sharing permissions
Choose this to add people, remove people, or change permissions.

Stop sharing
Choose this to remove the permissions for all the people you have shared with.

Figure 17. Or will I?

Now, it *is* possible to share my folder all by itself, whether or not Public folder sharing is enabled in the Network and Sharing Center (Figure 12). I don't think that in most situations I'd want that entire folder tree shared, so I'm going to turn Public folder sharing off and get myself back to Figure 12.

OK. Did that. Now I'll go back to the MyAppData folder with sharing gleaming in my eyes. (To reset everything after the prior experiment, I actually deleted the MyAppData folder and then recreated it.) Now if I right-click the folder and click Share (Figure 16), I get a different screen (Figure 18).



Choose people to share with

People must have a user account and password for this computer to access files you have shared. To change this setting, use the Network and Sharing Center.

Everyone

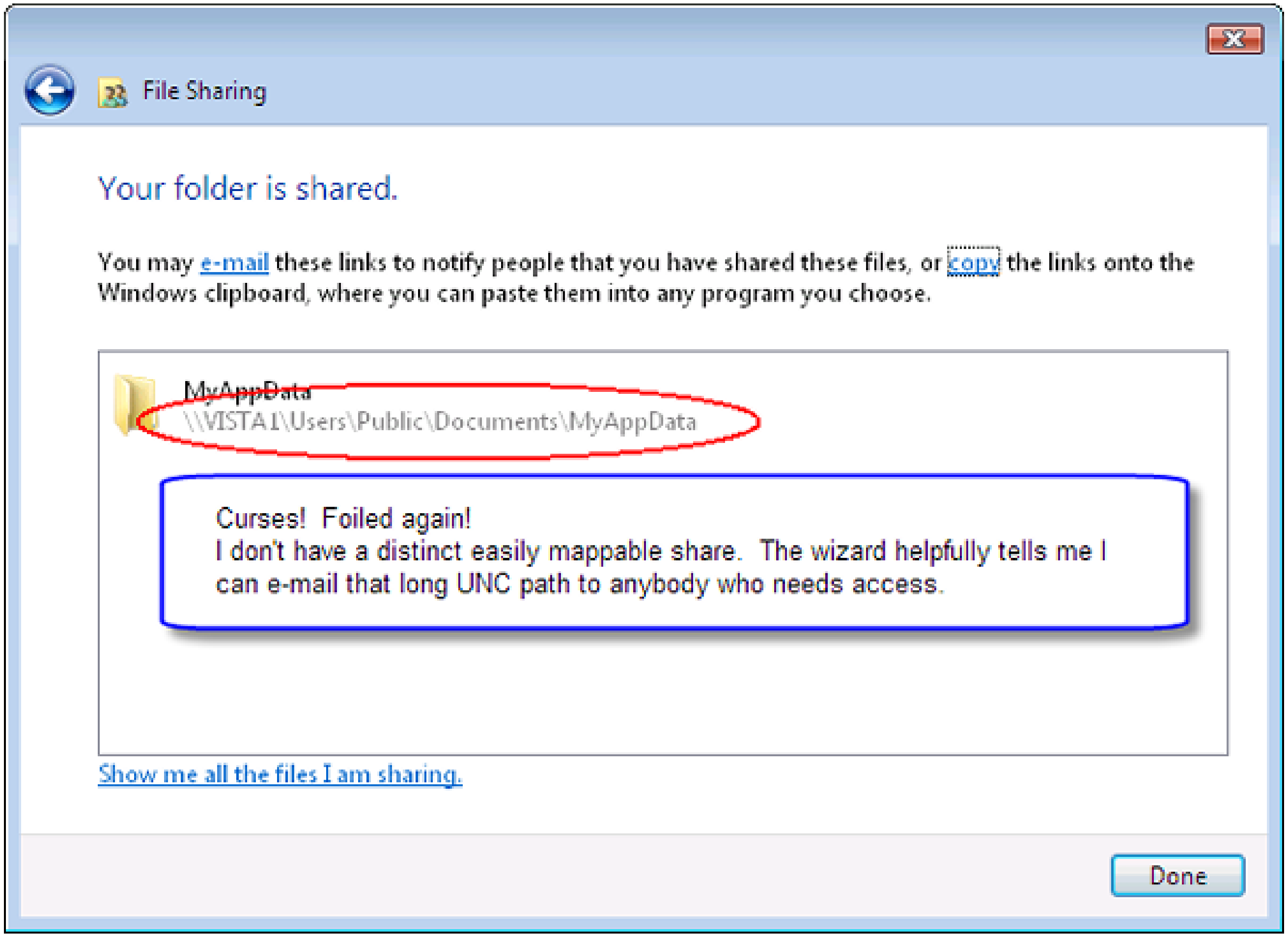
Name	Permission Level
Jane	Owner

I typed in **Everyone** for ease of illustration. I normally never give permissions to Everyone, preferring **Authenticated Users**.
After clicking **Add**, the newly added group (or individual user account) will appear in the list. I'll then click in the **Permission Level** column and specify whether I'm giving read-only or read/write permissions.

Tell me about different ways to share in Windows.

Figure 18. After I stop sharing the entire Public folder, I can create a share on my folder

Cool. This looks more promising. But as Figure 19 shows, looks can be deceiving! Instead of a simple individual shared folder as in earlier versions of Windows, the full path is shared. So again, a network user will have to navigate through multiple levels of folders to get to my data.



Your folder is shared.

You may e-mail these links to notify people that you have shared these files, or copy the links onto the Windows clipboard, where you can paste them into any program you choose.

MyAppData
\\VISTA1\Users\Public\Documents\MyAppData

Curses! Foiled again!
I don't have a distinct easily mappable share. The wizard helpfully tells me I can e-mail that long UNC path to anybody who needs access.

Figure 19. That dumb wizard didn't give me a 'real' share after all!

All right. Time to give up on the wizard and do it the old-fashioned way.

I've removed the share(s) I'd created and am starting fresh. (Again, for purpose of clearing out the prior experiment I've deleted MyAppData and then created a new folder using that name.)

This time, I'll right-click the MyAppData folder and click Properties rather than Share... (Figure 20). Note that I'm illustrating doing this within C:\Public, rather than within the Public folder structure at the top of the list in Windows Explorer. It makes no difference; you can do it either way.

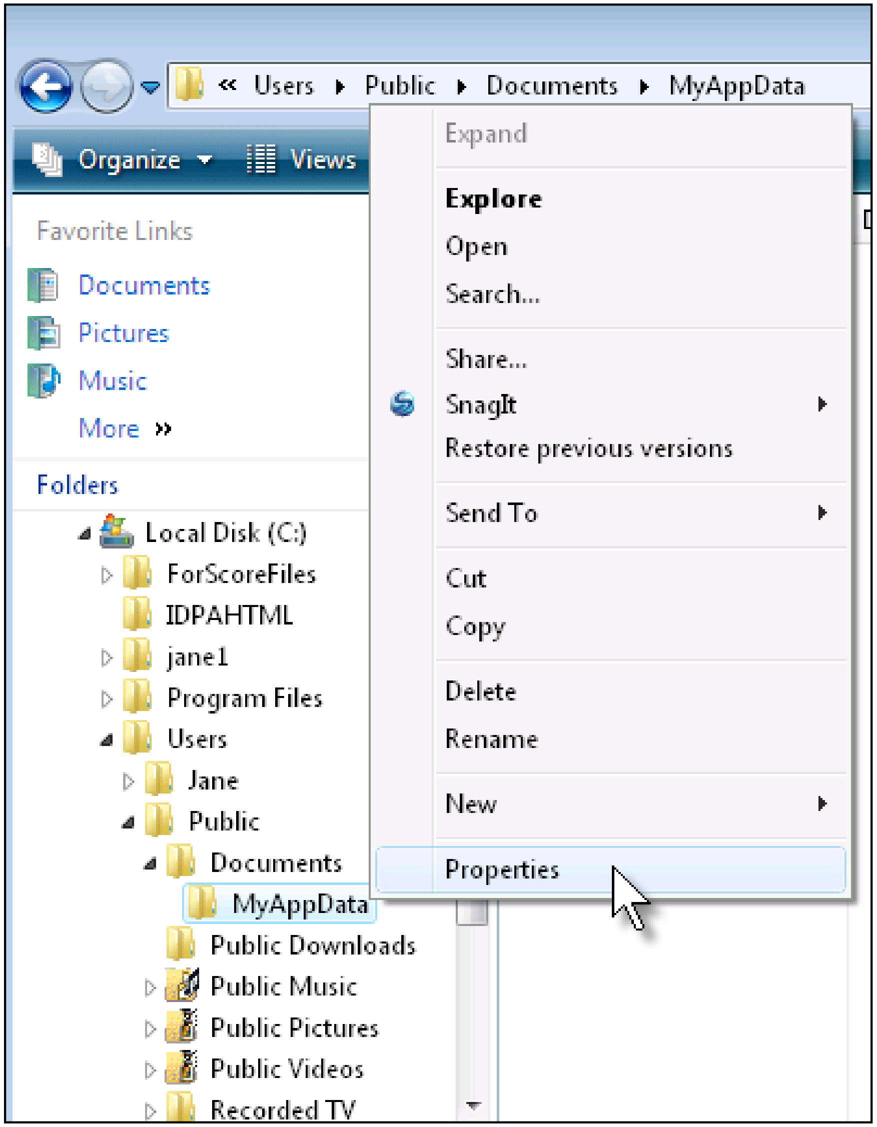
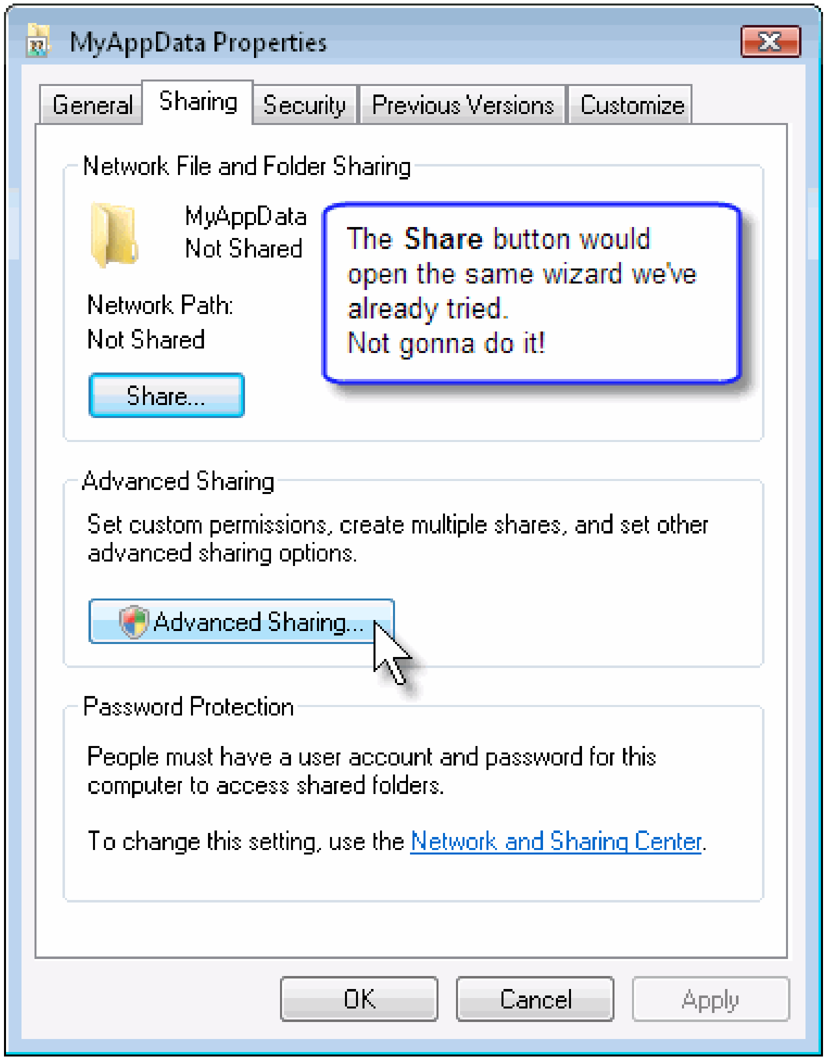


Figure 20. Select Properties rather than Share from the right-click menu



Network File and Folder Sharing

MyAppData
Not Shared

Network Path:
Not Shared

The **Share** button would open the same wizard we've already tried.
Not gonna do it!

Advanced Sharing

Set custom permissions, create multiple shares, and set other advanced sharing options.

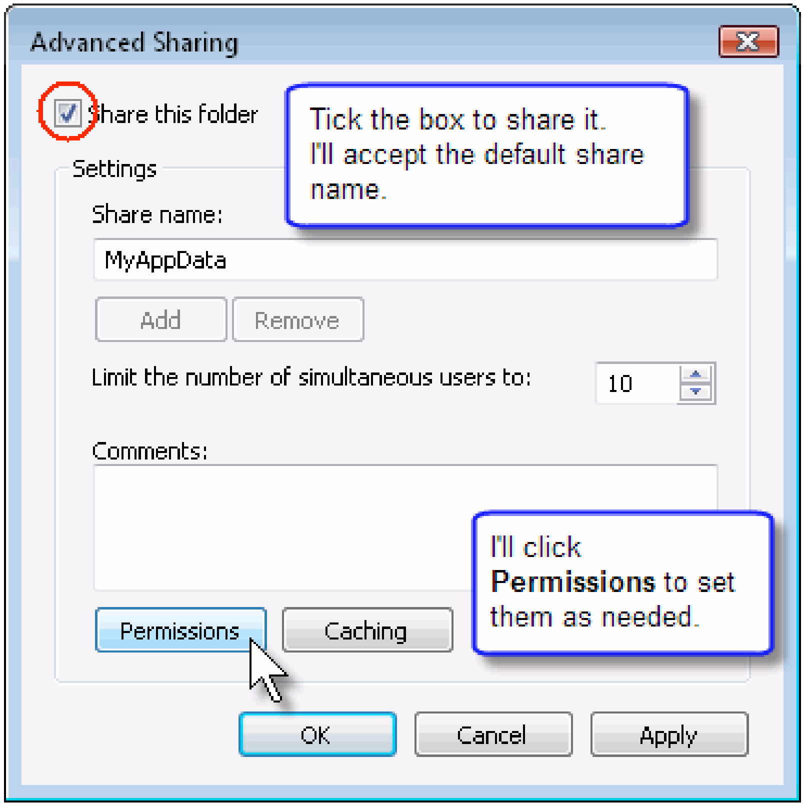
Password Protection

People must have a user account and password for this computer to access shared folders.

To change this setting, use the Network and Sharing Center.

Figure 21. On the Sharing tab, I'll click Advanced Sharing

The shield icon on the Advanced Sharing... button (Figure 21), as I've explained in an earlier article, is the warning that Vista will pop up the "are you sure" elevation dialog box.



Advanced Sharing

☑ Share this folder

Tick the box to share it. I'll accept the default share name.

Settings

Share name: MyAppData

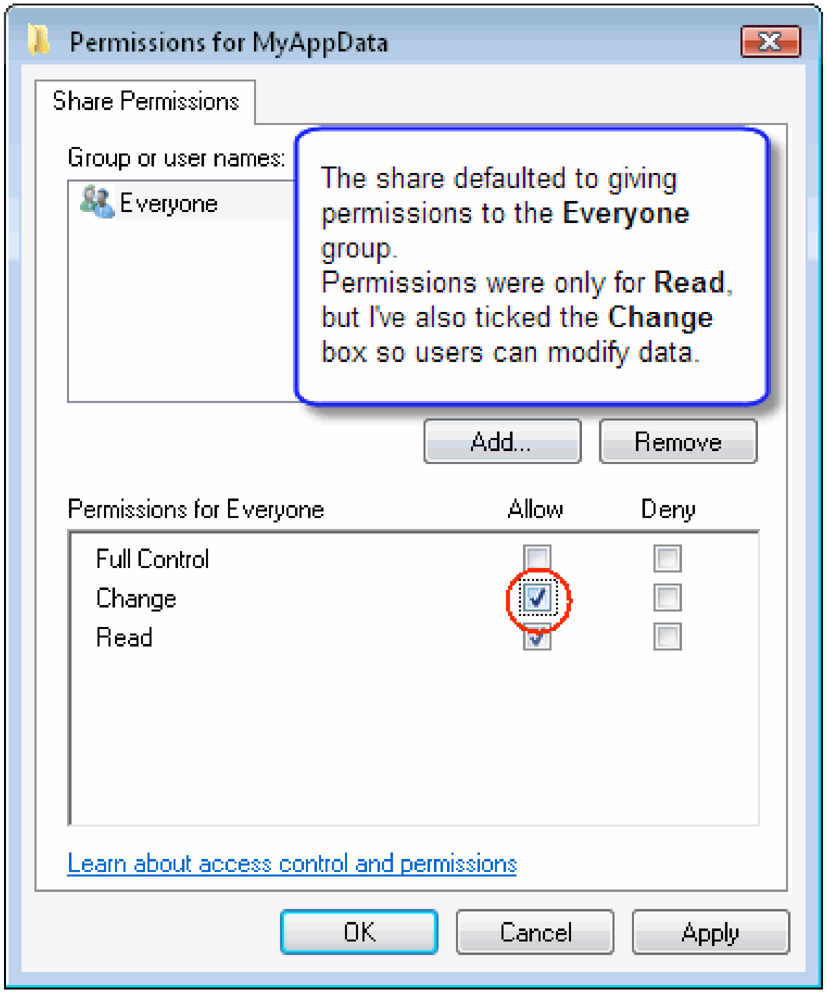
Limit the number of simultaneous users to: 10

Comments:

I'll click **Permissions** to set them as needed.

Figure 22. The Advanced Sharing settings box

On the Advanced Sharing window (Figure 22) I'll tick the Share this folder box. At this point, I can choose a different name for the share or just accept the default. I'll click the Permissions button to view and perhaps change permissions. I see that Vista has given Everyone read-only permissions. I'll add Change permissions (Figure 23).



Permissions for MyAppData

Share Permissions

Group or user names:

Everyone

The share defaulted to giving permissions to the **Everyone** group.
Permissions were only for **Read**, but I've also ticked the **Change** box so users can modify data.

Permissions for Everyone

Permissions for Everyone	Allow	Deny
Full Control	□	□
Change	☑	□
Read	☑	□

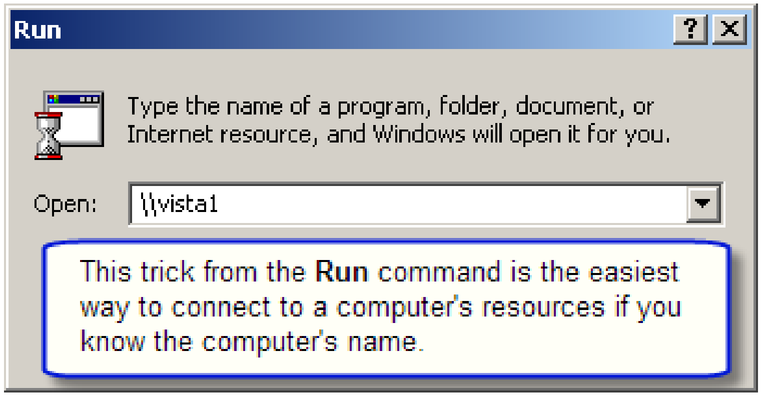
Learn about access control and permissions

Figure 23. I've added Change permissions so users can write to the data files in this share

I click OK a bunch of times, and wind up with a shared folder.

Now I need to be sure my users can connect to the shared folder. If you'll recall, the Network and Sharing Center default that I accepted (in Figure 12) was for Password protected sharing. On an individual computer (one that's not part of a domain), this means that a user will need to have an account name and password that this computer recognizes before it will grant access.

The easiest method for users to deal with is for you to create accounts on the computer that has the shared resource using the same logon names and passwords that your users use to log onto their own computers. Then,when they go to connect to the networked computer that has the shared resources (Figure 24) the experience is transparent. Otherwise they'll be asked to provide a user name and password (Figure 24).

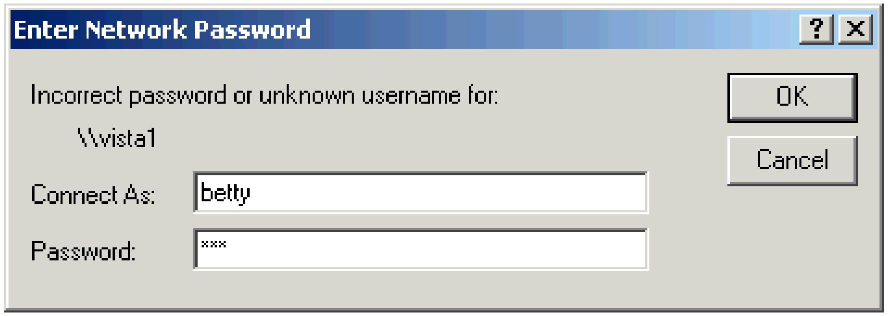


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: \\vista1

This trick from the **Run** command is the easiest way to connect to a computer's resources if you know the computer's name.

Figure 24. Connecting from a Win2K Pro machine to my Vista machine



Incorrect password or unknown username for:

\\vista1

Connect As: betty

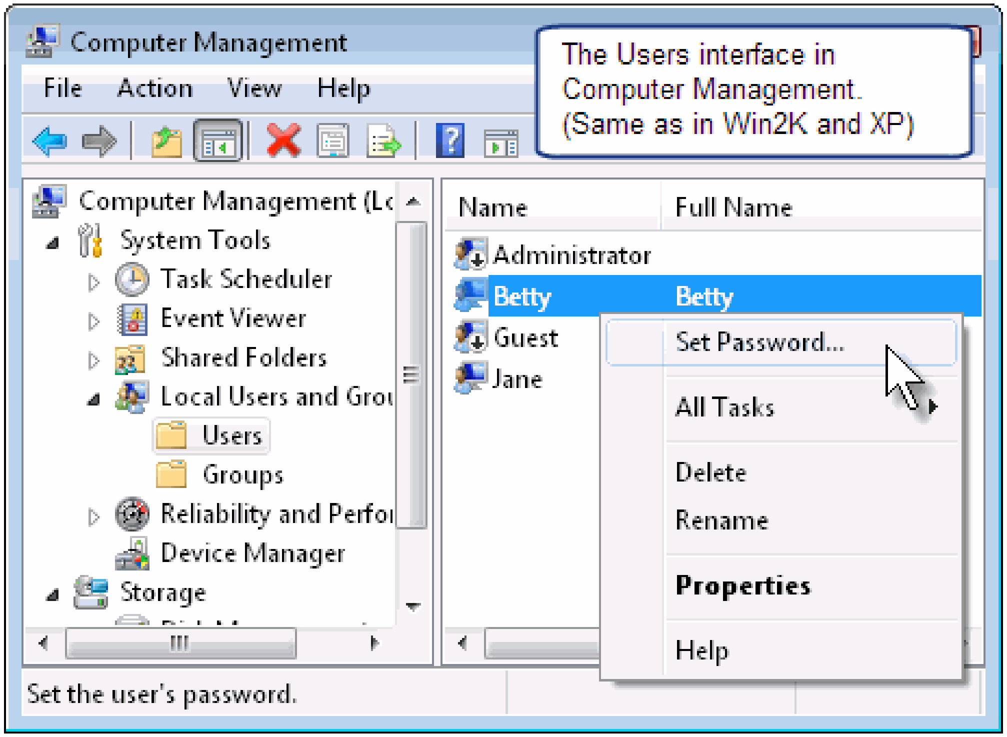
Password: ***

Figure 25. My logged-on name and password don't match one in the Vista box, so I need to enter a name/password combination it recognizes

By the way, I realize some people are comfortable configuring user accounts from the applet in Control Panel. I much prefer using the interface in Computer Management (Figure 26). Right-click My Computer (or Computer in Vista), then select Manage. This is the same interface as in Windows 2000 and XP. Unfortunately, this snap-in is not available in Vista Home Premium, so on that OS you'll need to use Control Panel.

Another option is to run an *elevated* command prompt (Run as Administrator) and use the net command. You can add an account like this for a user named Betty who has as her password xxx :

```
net user betty xxx /add
```

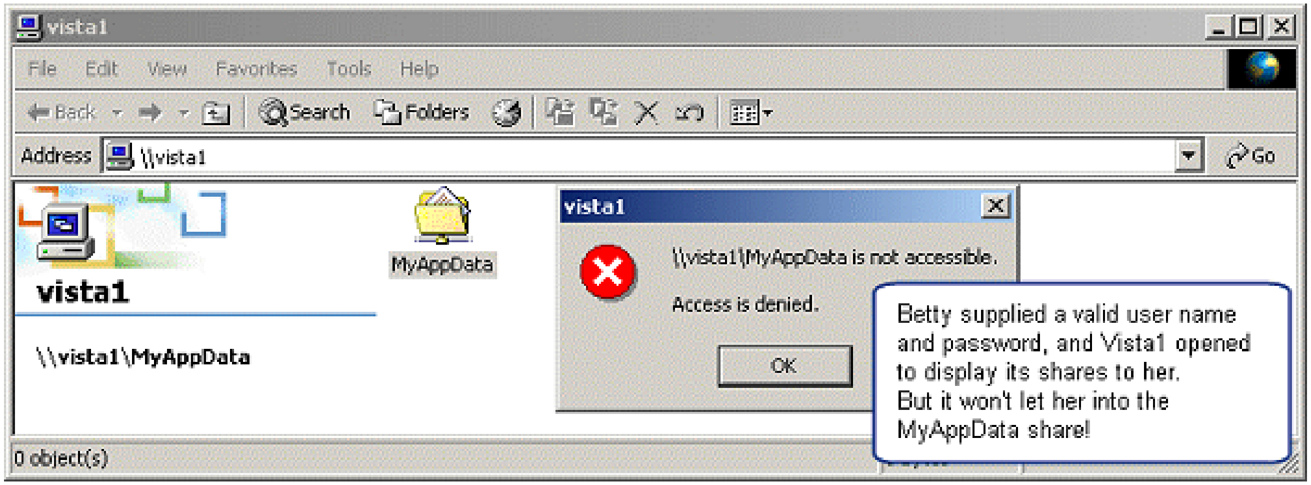


The Users interface in Computer Management. (Same as in Win2K and XP)

Figure 26. Configuring the password on a user account

Whether through having a matching account name and password, or correctly entering a recognized user name and password, Betty has now opened the Vista1 computer and can view its shares. And my MyAppData share now looks like a regular share, not some long UNC name. Can my toil really be over?

Ummm... even though Vista accepted Betty's account name and password to display its wares, there's still trouble afoot when she tries to access the share (Figure 27); Betty gets an access denied message.



vista1

\\vista1\MyAppData is not accessible.

Access is denied.

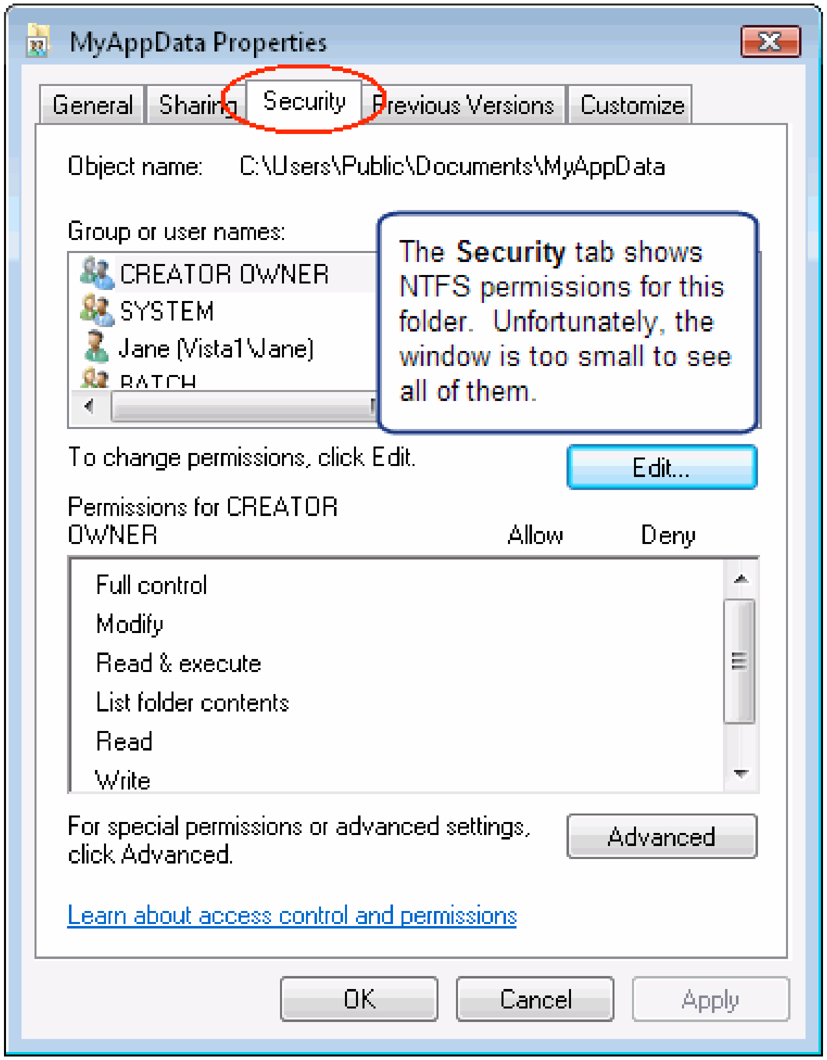
OK

Betty supplied a valid user name and password, and Vista1 opened to display its shares to her. But it won't let her into the MyAppData share!

Figure 27. Betty can access the Vista1 computer, but not the MyAppData share

While I created the share and gave read-write permissions to Everyone (Figure 23), those were *share* permissions. In that Vista must be installed on an NTFS partition, I also have NTFS permissions to contend with. (Understanding how NTFS and share permissions combine can be a little tricky. Basically, whichever of the two permissions on a particular resource is more restrictive will dictate a user's access to that resource over the network.)

I right-click the MyAppData folder on the Vista machine and click Properties again, and this time select the Security tab (Figure 28) to look at the NTFS permissions on the folder.



Object name: C:\Users\Public\Documents\MyAppData

Group or user names:

- CREATOR OWNER
- SYSTEM
- Jane (Vista1\Jane)
- BATCH

The **Security** tab shows NTFS permissions for this folder. Unfortunately, the window is too small to see all of them.

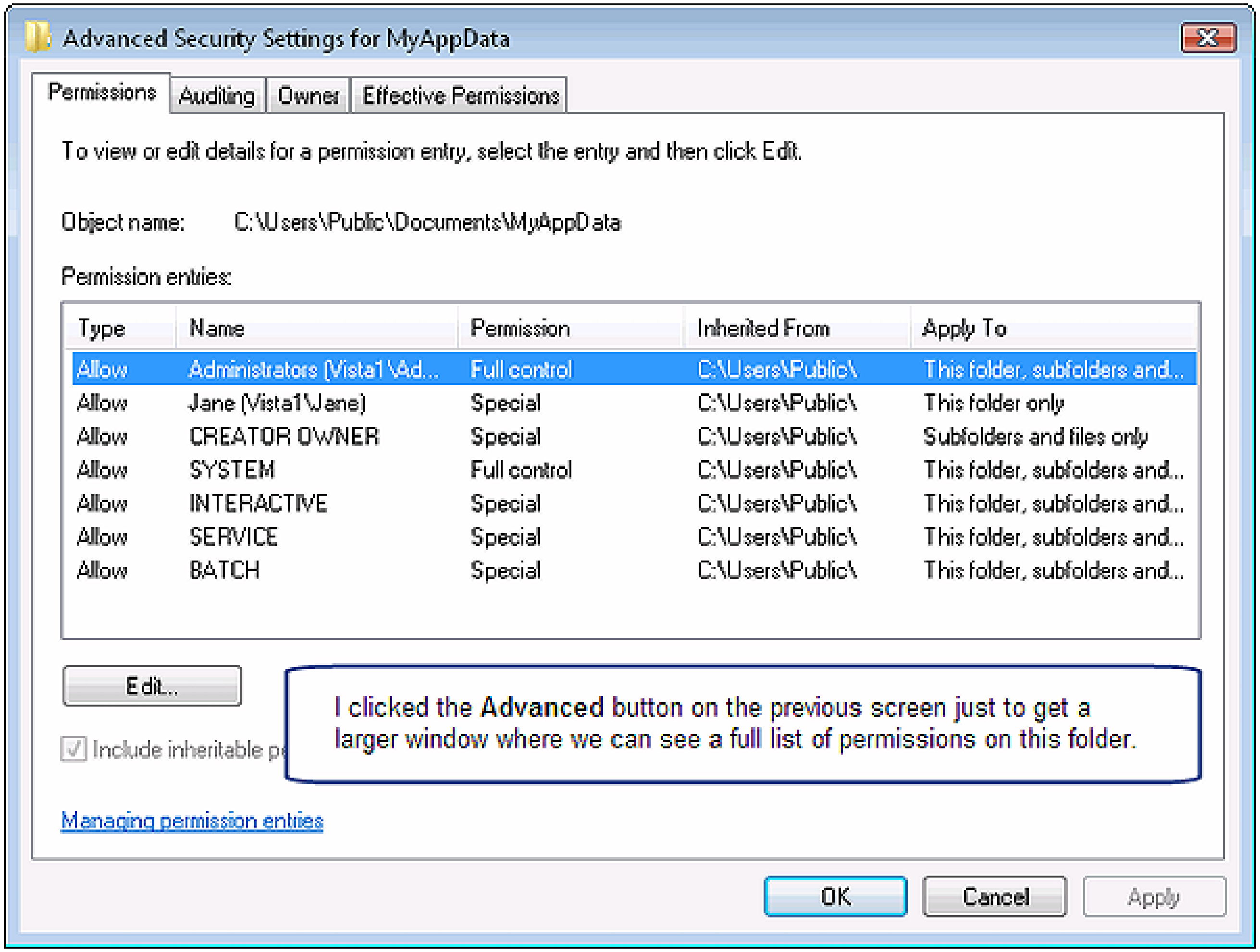
To change permissions, click Edit.

Permissions for CREATOR OWNER	Allow	Deny
Full control		
Modify		
Read & execute		
List folder contents		
Read		
Write		

For special permissions or advanced settings, click Advanced.

Figure 28. Viewing NTFS permissions for my folder

While I don't need to play with "advanced" permissions to solve this, I've clicked the Advanced button just to get a larger listing showing who has any permissions to this folder (Figure 29).



To view or edit details for a permission entry, select the entry and then click Edit.

Object name: C:\Users\Public\Documents\MyAppData

Permission entries:

Type	Name	Permission	Inherited From	Apply To
Allow	Administrators (Vista1\Ad...	Full control	C:\Users\Public\	This folder, subfolders and...
Allow	Jane (Vista1\Jane)	Special	C:\Users\Public\	This folder only
Allow	CREATOR OWNER	Special	C:\Users\Public\	Subfolders and files only
Allow	SYSTEM	Full control	C:\Users\Public\	This folder, subfolders and...
Allow	INTERACTIVE	Special	C:\Users\Public\	This folder, subfolders and...
Allow	SERVICE	Special	C:\Users\Public\	This folder, subfolders and...
Allow	BATCH	Special	C:\Users\Public\	This folder, subfolders and...

I clicked the Advanced button on the previous screen just to get a larger window where we can see a full list of permissions on this folder.

Figure 29. Advanced NTFS permissions for the folder (view full size image**)**

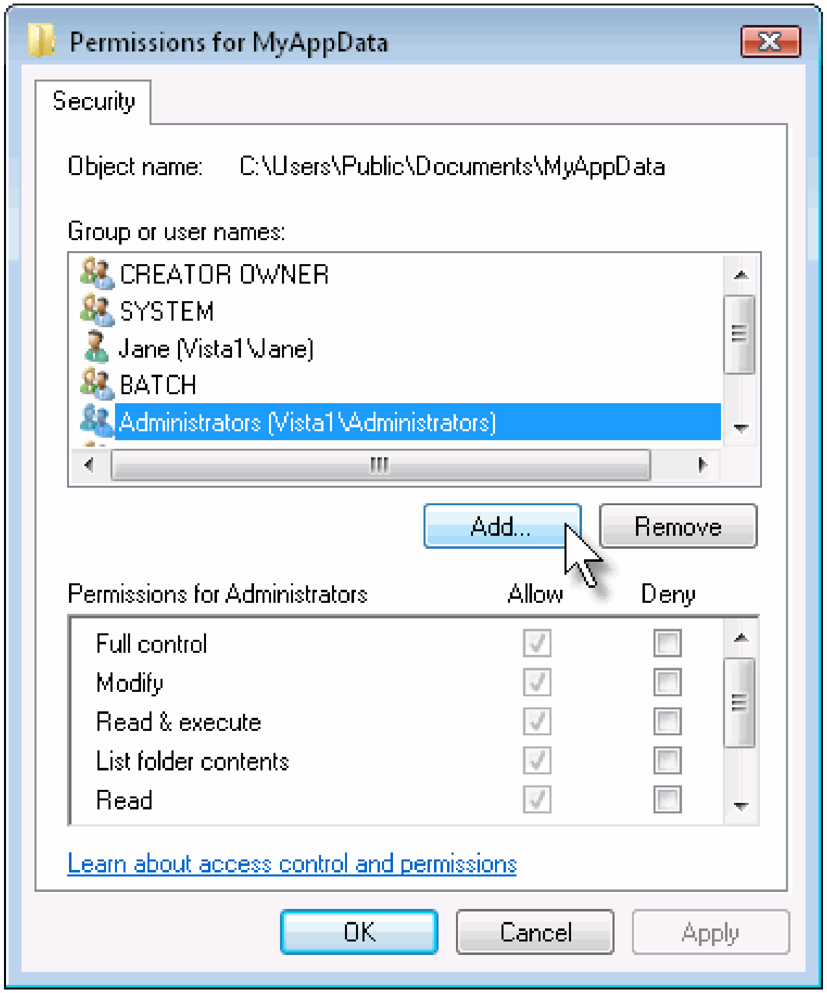
Users listed include Administrators, Jane (me), CREATOR OWNER, SYSTEM, INTERACTIVE, SERVICE, and BATCH.

Betty is an ordinary user, not a member of the Administrators group. She isn't the person who created this folder (its creator owner), and she's certainly not part of the system, a service, or a batch job. What does that Interactive mean?

"Interactive" is a term Microsoft uses for a user who's physically sitting at a keyboard of a computer, rather than accessing it across a network. Now that I've created an account for Betty on the Vista computer, she could sit down at that computer's keyboard and log on to it and access the data in MyAppData... as can anybody who's authorized to log onto the computer. But she doesn't have permissions to do so across the network from a different computer.

So I need to add something to the list of NTFS permissions for the folder. I could add Betty's user account individually. I could create a group, add Betty's and other user accounts to the group, then give the group permissions to the folder. Or if it's a small network within a small business and I'm not hyper concerned about security, I could grant access permissions to Everyone.

For this example, I'll add Betty's user account and give her read-write permissions. I'll click the Edit button (Figure 28). Then click Add (Figure 30).


Permissions for MyAppData
Security
Add...
Remove
Learn about access control and permissions
OK
Cancel
Apply

Object name: C:\Users\Public\Documents\MyAppData

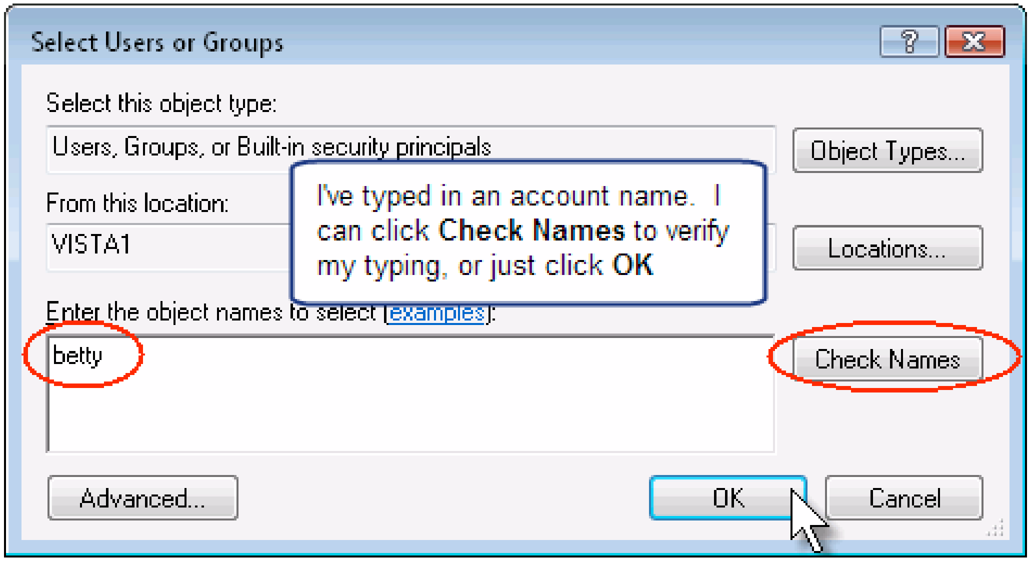
Group or user names:

- CREATOR OWNER
- SYSTEM
- Jane (Vista1\Jane)
- BATCH
- Administrators (Vista1\Administrators)

Permissions for Administrators	Allow	Deny
Full control	☑	☐
Modify	☑	☐
Read & execute	☑	☐
List folder contents	☑	☐
Read	☑	☐

Figure 30. Click the Add... button to add an account or group to the permissions list

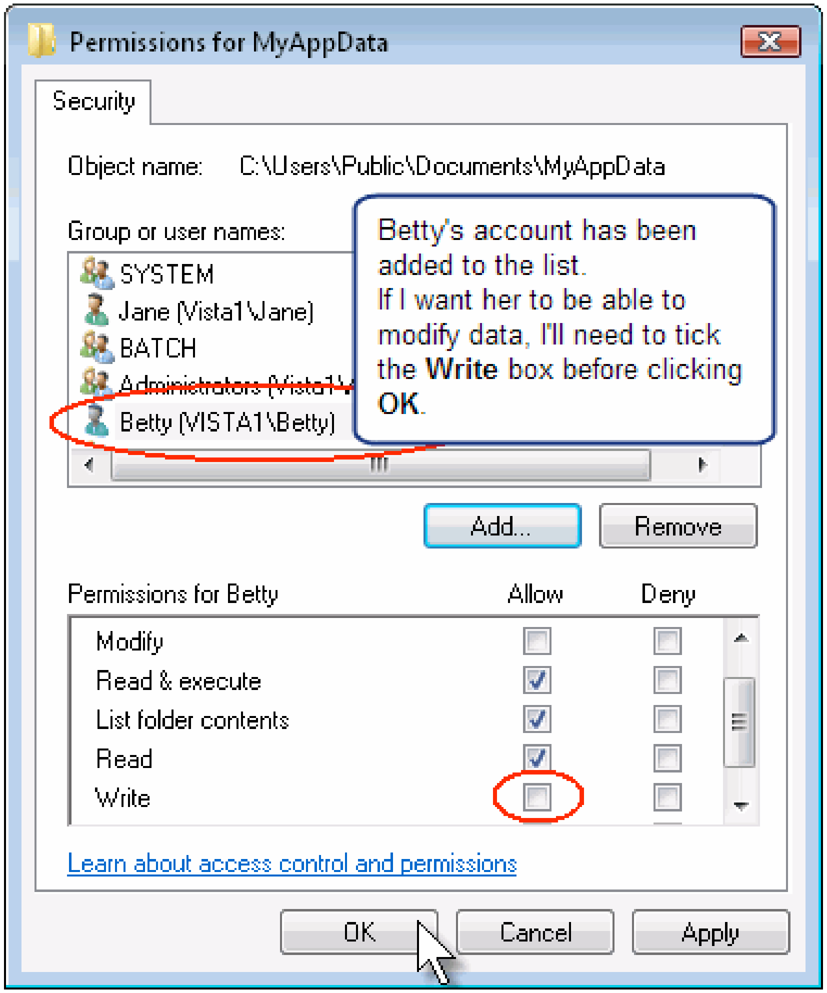
I'll add Betty by name (Figure 31), then click OK.



I've typed in an account name. I can click **Check Names** to verify my typing, or just click **OK**

Figure 31. Adding an individual user account

And for a normal business database-app environment, I'll need to give her write permissions as well (Figure 32). Notice that her account is listed as VISTA1\Betty. This means that the account exists within the security database on the computer called Vista1. Of course, in my small network scenario an identical user name/password account exists in the security database of the Windows 2000 machine where Betty is logged on.



Permissions for MyAppData

Security

Object name: C:\Users\Public\Documents\MyAppData

Group or user names:

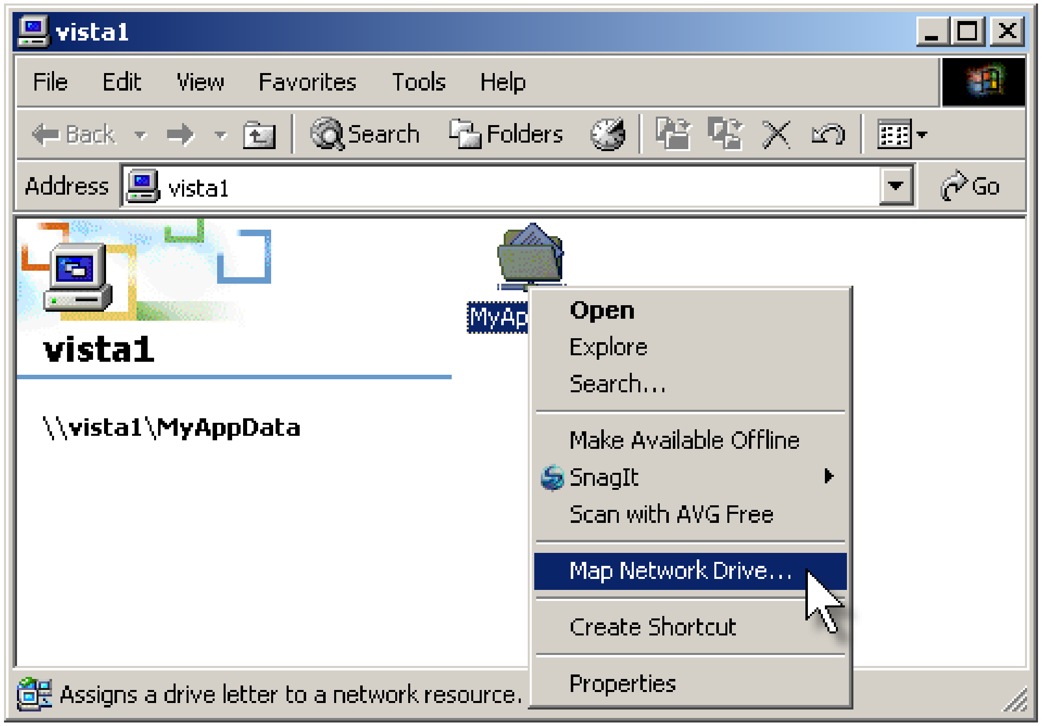
- SYSTEM
- Jane (Vista1\Jane)
- BATCH
- Administrators (Vista1\
- Betty (VISTA1\Betty)

Betty's account has been added to the list.
If I want her to be able to modify data, I'll need to tick the **Write** box before clicking **OK**.

Permissions for Betty	Allow	Deny
Modify	□	□
Read & execute	☑	□
List folder contents	☑	□
Read	☑	□
Write	□	□

Figure 32. I need to tick the Write box before closing this screen

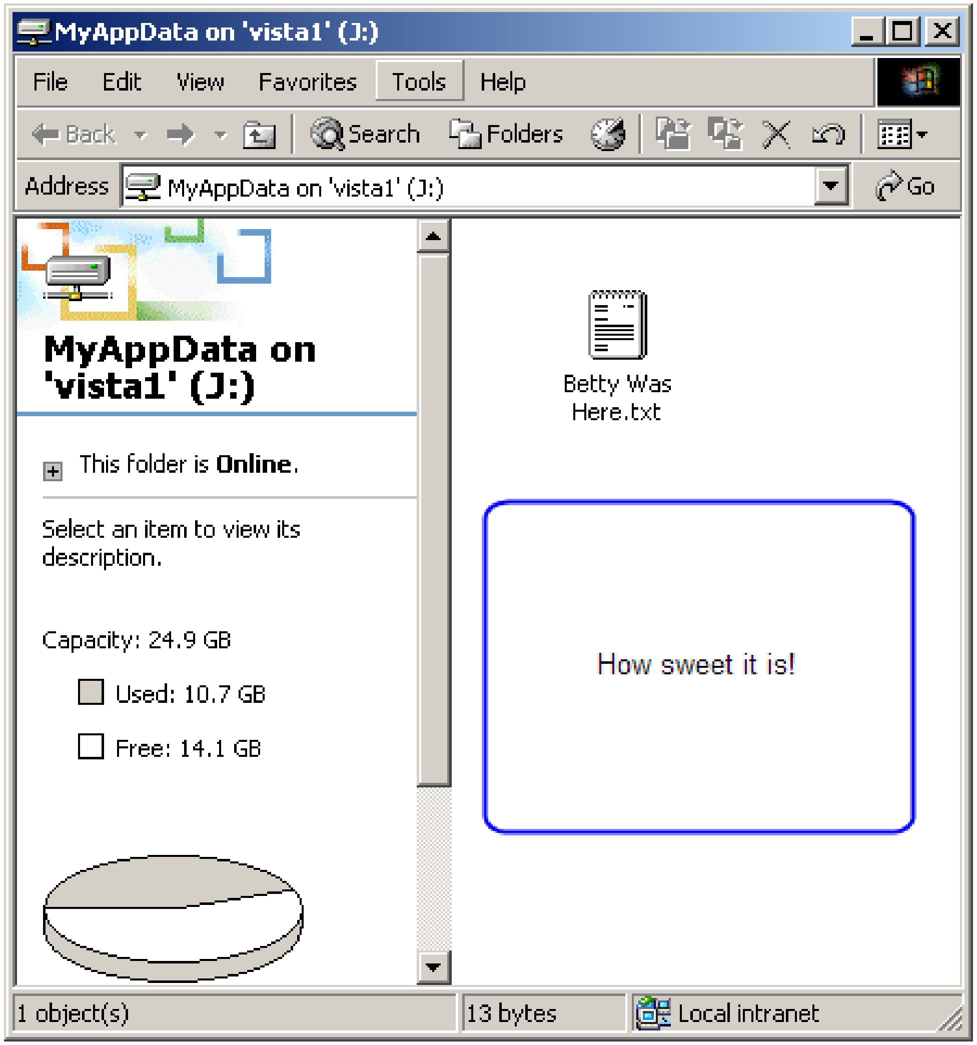
So Betty tries again. And lo, the shared resources screen not only opens, but lets her map a drive (Figure 33).



Assigns a drive letter to a network resource.

Figure 33. Successfully mapping a drive!

And she can write to the folder (Figure 34).



MyAppData on 'vista1' (J:)
File Edit View Favorites Tools Help
Back Search Folders
Address MyAppData on 'vista1' (J:) Go

MyAppData on 'vista1' (J:)

This folder is **Online**.

Select an item to view its description.

Capacity: 24.9 GB

□ Used: 10.7 GB

□ Free: 14.1 GB

Betty Was Here.txt

How sweet it is!

1 object(s) | 13 bytes | Local intranet

Figure 34. An ordinary user successfully mapped and writes to the share

Real men

For those who live in the "real men don't use no stinkin' mice" strata, you could create your folder and share it with a few lines typed at an elevated (Run as Administrator) command prompt.

First, create the folder:

```
md c:\users\public\documents\MyAppData
```

Then share the folder (everthing needs to be typed on one line):

```
net share MyAppData=c:\users\public\documents\myappdata /grant:everyone,change
```

Finally, grant Betty Change NTFS permissions on the MyAppData folder:

```
cacls c:\users\public\documents\myappdata /e /g betty:c
```

All right. I'll admit I really dragged this out. Once you understand what's involved, the closing executive summary will be short and sweet. But I think it doesn't hurt to understand how I arrived at this list.

Executive Summary #3:

1. CSIDL_COMMON_DOCUMENTS is where Microsoft intends us to put data that needs to be modified by multiple users who share a computer.
2. CSIDL_COMMON_DOCUMENTS is part of the new Public structure in Vista. Sharing the Public folder through the Network and Sharing Center (Figure 12) exposes the entire Public folder hierarchy to the network. Users will need to thread their way through that to get to your data folder.
3. Vista's folder sharing wizard treats folders within the Public structure differently from other folders. It won't let you create a folder and then share it as a "normal" share.
4. Rather than using the wizard, you can create a "normal" share by using the "Advanced" option on the Sharing tab for your folder.
5. If you create a share using the "Advanced" option, you'll also need to open up NTFS permissions so that other non-administrator users on your network can access your share.

Jane Fleming is a college dropout who subsequently lived four years in Europe, a year and a half in Mexico, and three years in India, and later taught yoga for a living in California (she’s been vegetarian since 1970). She developed circuits and wrote assembly code for several embedded microcontroller projects during the 1980s. She began using Clarion Professional

Developer for in-house projects back when Clarion was running display ads in InfoWorld and has used it very intermittently since. She is a former Microsoft Certified Trainer and taught Microsoft and Novell network administration at a business college for four years. Now widowed ten years, Jane plays classical piano and has found her métier as a semi-retired NRA-certified pistol instructor.

Reader Comments

Posted on Monday, August 20, 2007 by Peter Gysegem

Hi Jane,

Great series of articles, thanks. Do you know if you can share the MyApp folder across the network using the Set Access Control in SetupBuilder?

Add a comment

Clarion Magazine

Coping With Vista - It's Nice To Share

by Jane Fleming

Published 2007-08-03

As mentioned in some of my previous ClarionMag articles, Vista has complicated our lives a bit as far as deciding where to store database files on an individual computer (client-server applications aren't significantly changed in this regard).

> **Executive Summary #1:** Yes, CSIDL_COMMON_DOCUMENTS is a good choice, with some possible permissions caveats. Return (level:benign).

Now... for those of us who aren't executives, I'd like to take a look at the new Vista "Public" file structure. I'll just touch on permissions in this article. Some of the comments I've read on the news groups such as "the application has total access to the files but the average user will not be able to change a file" lead me to think that Windows (and Vista) permissions is worthy of its own article.

Last February I strapped on the old automobile and drove down to Los Angeles for a Vista "Roll-Out Event". The IT Pro track was full, but I got a ticket for the developers' track.

I had a couple of specific questions I hoped to pose. One had to do with data storage locations.

As I mentioned in a previous ClarionMag article, data storage for an individual user isn't terribly different under Vista – each user has a Documents folder, just as each user in prior versions of Windows had his own My Documents folder. And data storage for domain-based networks isn't radically changed either – a server is still a server.

The challenge comes when more than one person needs to use an app on a particular machine, and in so doing needs to share data and perhaps configuration settings with the other users of that computer. Vista presents the issues of file and registry virtualization, and the fact that users cannot write to a folder within Program Files or to the Windows folder without using an elevated token (unless you explicitly change permissions for the areas to which your app needs to write.).

I went into these issues in some detail in several previous ClarionMag articles. Thus, INI files can't go into C:\Windows (one of the Clarion default options). And if you put your apps into C:\Program Files, you can no longer put INI files or data in the same folder with your apps (another default Clarion option).

So, what does Microsoft suggest??

I walked up to the front of the auditorium during one of the first breaks in the meeting and told the main speaker I hoped she'd get into that. "Nope," said she. "You can get that information from MSDN." Thanks a lot! The roll-out was basically a sell-the-glitz pep rally for developers. No wonder some of these Microsofties have the job title of Evangelist.

Well, I've wandered around MSDN some. It would be really nice if Microsoft had a definitive answer to this. But I wasn't able to find it on MSDN. A man whose experience I respect quite a lot has observed that Microsoft makes a large number of its decisions based on what is going to appeal to the Fortune 500. Those are the big dogs who write Microsoft a fat check for one installation DVD and 10,000 licenses. And MS obviously also puts some attention into attracting individuals who are going to buy OS upgrades (or, more likely, are going to go out and buy new PCs). Those of us in the funky landscape in between are somewhat on our own.

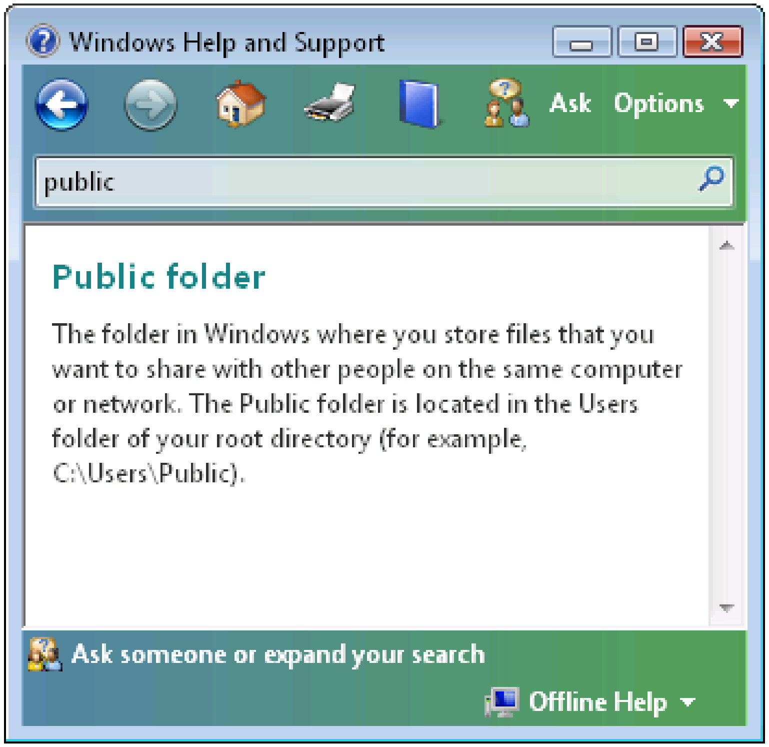
In a prior article, I said that Microsoft's UAC for Developers document suggests using CSIDL_COMMON_APPDATA. (Randy Rogers' recent ClarionMag article has a great little demo app enumerating the various CSIDLs (constant special item ID lists).)

Note that as many of us (well, me anyway) were just getting our heads around CSIDLs, Microsoft has deprecated them and maintains them in Vista "for backward compatibility". The new latest-and-greatest (Vista and future systems) folder constantsare KNOWNFOLDERIDs. But the API to use them is only valid on Vista and above. Argh!

Two things make CSIDL_COMMON_APPDATA, which on a Vista machine resolves to C:\ProgramData, less than ideal. First, it's a hidden folder. Second, default folder permissions give a user the ability to create and modify a document; but for anybody other than the creator, permissions to that document are read-only.

At the end of my prior article there's a link to a discussion thread Friedrich Linder came across on one of the MDSN forums. In this thread the suggestion is made to use CSIDL_COMMON_DOCUMENTS. A number of subsequent Clarion news group threads have picked up on this location. I want to take a look at that option in this article. (And reveal my own latest experience with "when all else fails, read the manual".)

As I said, the "look on MSDN" advice from the cheerleader at the Microsoft event wasn't particularly fruitful. Well, wouldya believe F1 to the rescue? In Vista's Help, searching with the word public brings up this explicit little tidbit on the new Public folder (Figure 1).

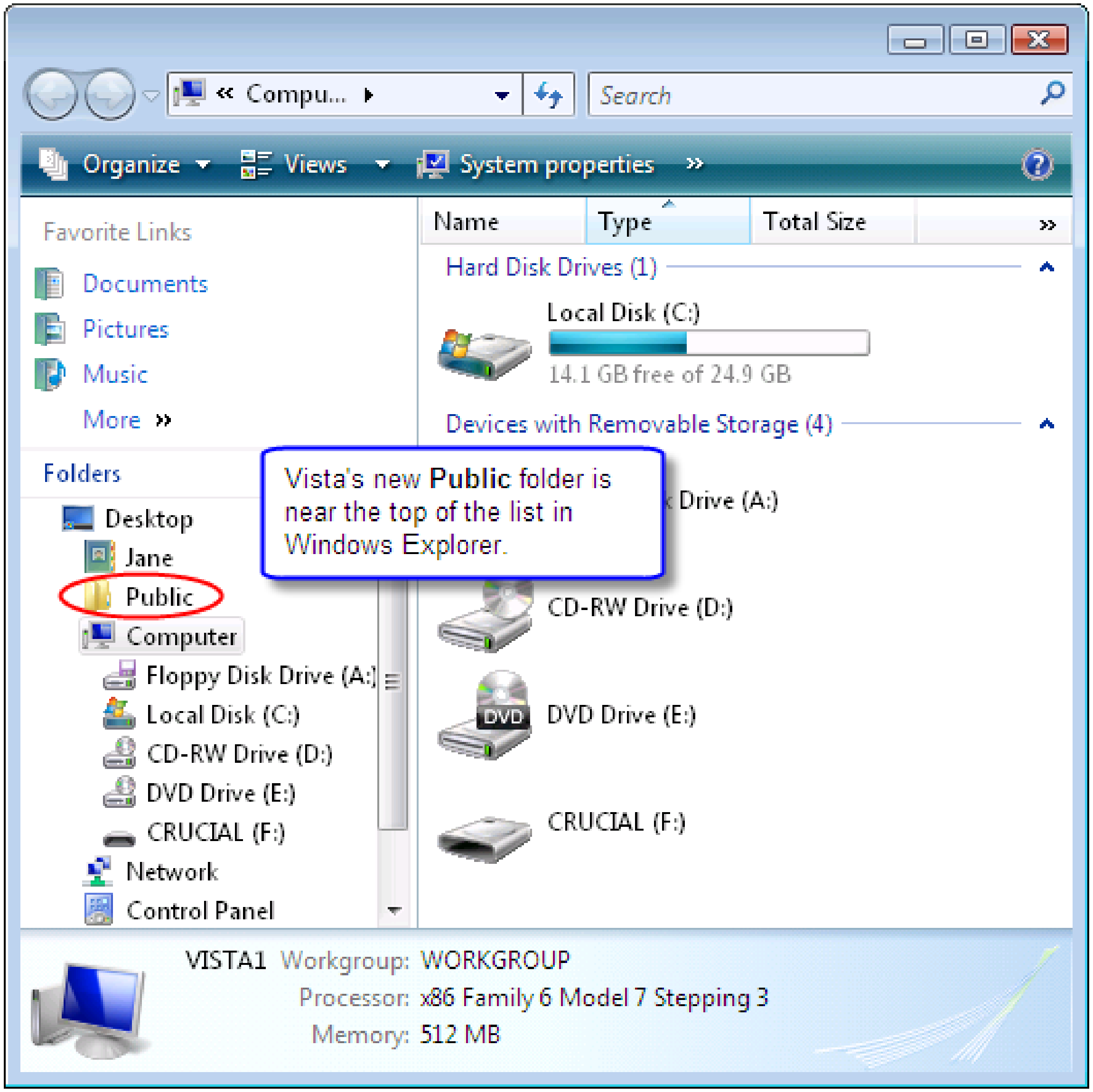


Public folder

The folder in Windows where you store files that you want to share with other people on the same computer or network. The Public folder is located in the Users folder of your root directory (for example, C:\Users\Public).

Figure 1. Vista help file description of Public folder

Public is a new file structure, conspicuously near the top of the listing in Windows Explorer (Figure 2).



14.1 GB free of 24.9 GB

Vista's new **Public** folder is near the top of the list in Windows Explorer.

VISTA1 Workgroup: WORKGROUP
Processor: x86 Family 6 Model 7 Stepping 3
Memory: 512 MB

Figure 2. Public folder in Windows Explorer

Expanding Public, there are a number of other Public... folders within it (Figure 3).

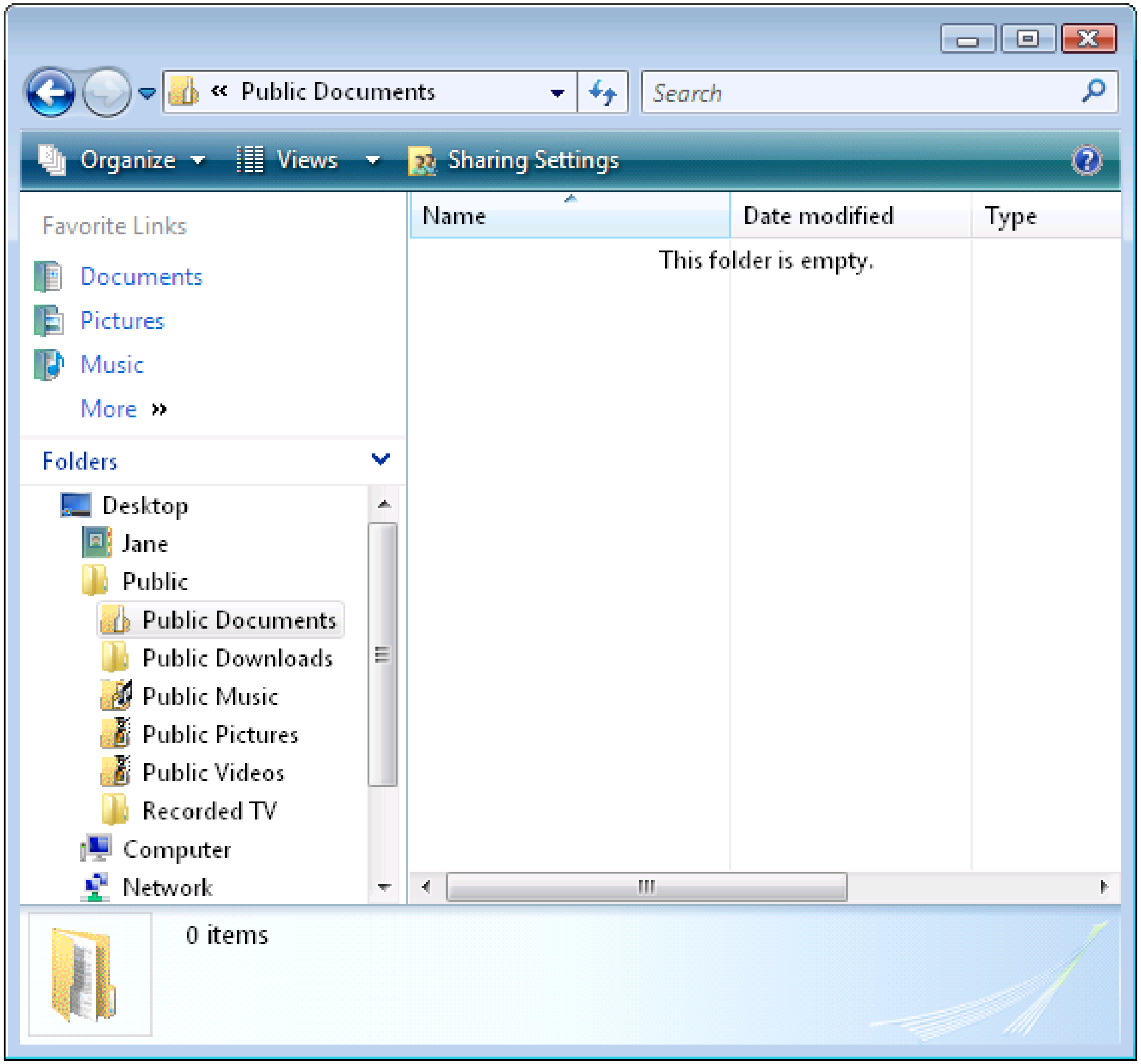
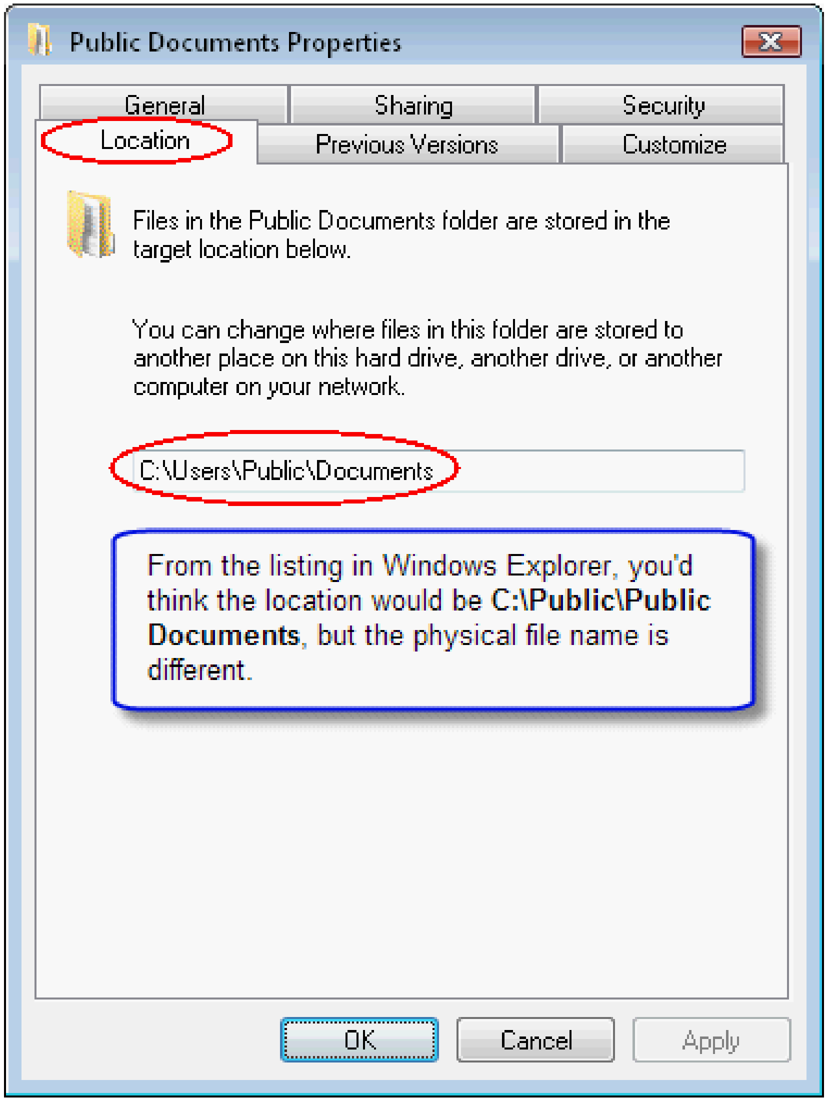


Figure 3. Folders within the "Public" structure

Right-clicking Public Documents and selecting Properties shows a tab that's not on the properties sheet for regular folders – Location (Figure 4).



Files in the Public Documents folder are stored in the target location below.

You can change where files in this folder are stored to another place on this hard drive, another drive, or another computer on your network.

C:\Users\Public\Documents

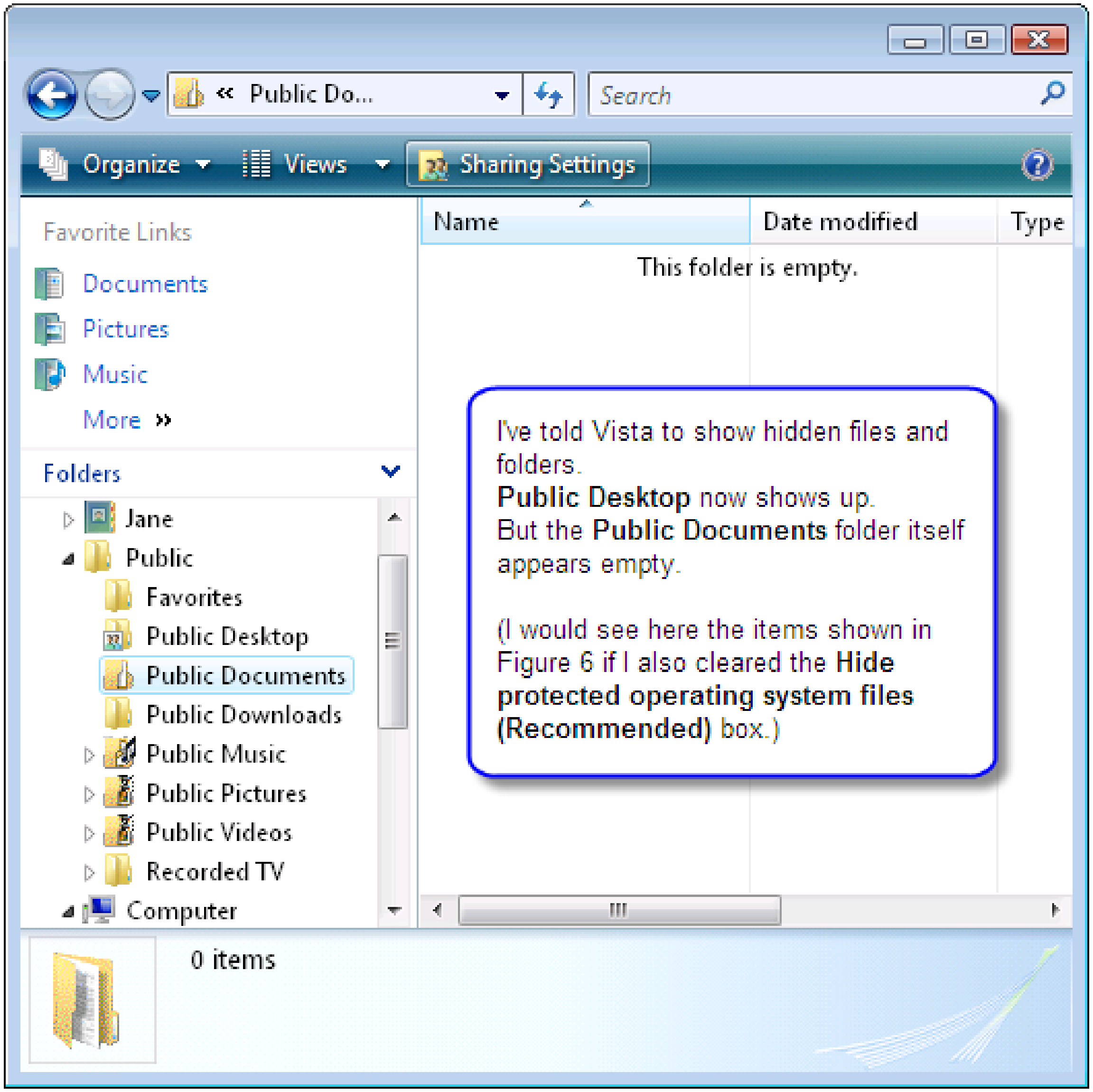
From the listing in Windows Explorer, you'd think the location would be **C:\Public\Public Documents**, but the physical file name is different.

Figure 4. The "Location" tab. Note that the physical location is slightly different from what's displayed in Explorer

That's one clue that this structure has some surprises in store.

Now, guess what CSIDL_COMMON_DOCUMENTS resolves to under Vista by default? You got it! C:\Users\Public\Documents.

Here's one tantalizing glimpse for a future exploration, if you'll pardon the thread drift. If you tell Windows to show hidden files and folders, the Public Documents folder still appears empty (Figure 5.)

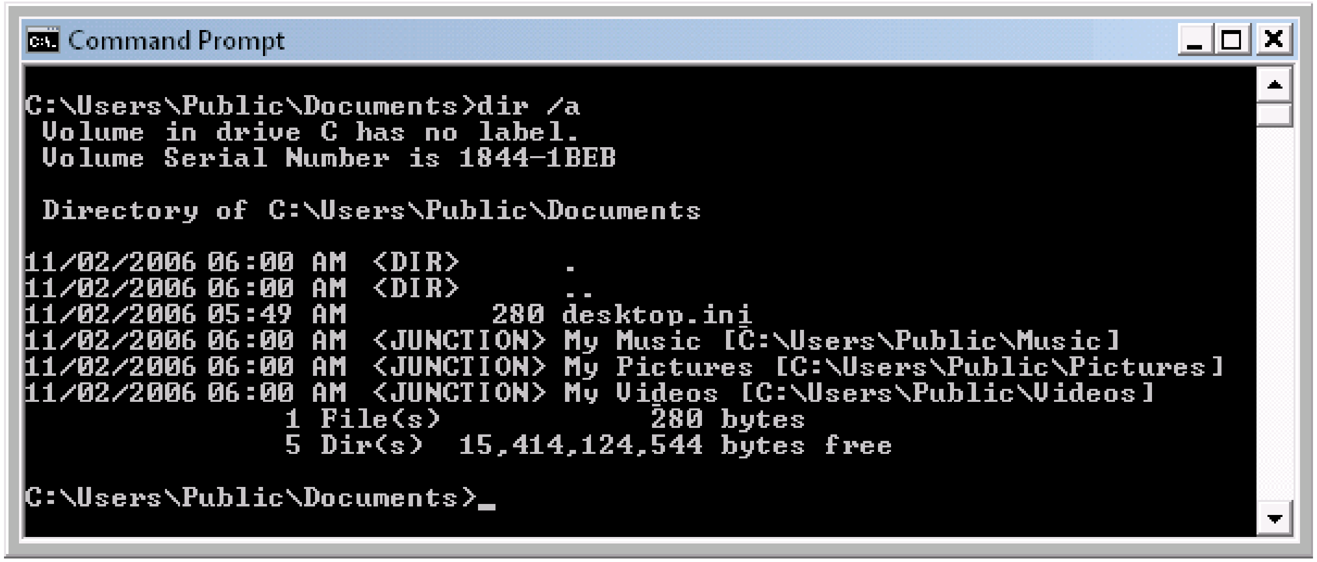


I've told Vista to show hidden files and folders.
Public Desktop now shows up.
But the **Public Documents** folder itself appears empty.

(I would see here the items shown in Figure 6 if I also cleared the **Hide protected operating system files (Recommended)** box.)

Figure 5. "Public Documents" is empty

But is it really empty? At a command prompt, running a directory command with the "show all files" switch reveals..... junctions! (Figure 6).



```
C:\Users\Public\Documents>dir /a
 Volume in drive C has no label.
 Volume Serial Number is 1844-1BEB

 Directory of C:\Users\Public\Documents

11/02/2006  06:00 AM    <DIR>          .
11/02/2006  06:00 AM    <DIR>          ..
11/02/2006  05:49 AM               280 desktop.ini
11/02/2006  06:00 AM    <JUNCTION>     My Music [C:\Users\Public\Music]
11/02/2006  06:00 AM    <JUNCTION>     My Pictures [C:\Users\Public\Pictures]
11/02/2006  06:00 AM    <JUNCTION>     My Videos [C:\Users\Public\Videos]
               1 File(s)            280 bytes
               5 Dir(s)  15,414,124,544 bytes free

C:\Users\Public\Documents>_
```

Figure 6. Junctions? Junctions???

NTFS has had junctions, or volume mount points, since Windows 2000. A junction links one directory entry to another; in Figure 6 c:\Users\Public\Documents\My Music and c:\Users\Public\Music point to the same physical location. But this usage is new. MSDN has several articles on the use we're seeing here: you may want to get an overview of junction points and consider their relevance to backup programs Junction points only work with folders; Vista has added symbolic links, like soft links in UNIX, which also work for individual files. There's a good description in this article.

So, back to our issue. It would seem that creating a MyAppData folder within the Public Documents folder would do what I want as far as being able to have database files and perhaps INI files shared regardless of who's logged onto the computer.

Before I go deeper into this – and there are some interesting gotchas whether you use the wizard or configure permissions manually – I'd like to go back to the CSIDL_COMMON_DOCUMENTS versus CSIDL_COMMON_APPDATA question.

One issue I've found with MSDN is a certain "boneyard" element. It's hard to tell at what point during which beta period various documents were originally written. Even though the UAC for Developers article I referenced above shows as being "Updated: February 2007", it doesn't even mention CSIDL_COMMON_DOCUMENTS.

The Windows Vista (tm) Resource Kit was released by Microsoft Press in April. Like previous Resource Kits, it's aimed more toward network administrators and support people than toward developers. At 5 pounds 2 ounces (including a DVD with a full PDF copy of the text and additional scripts and tools) and 1,568 pages, it's hardly "light" reading. The resource kit states right at the outset that "this book does not cover features that are primarily used in home environments, such as parental controls, games, Windows Media Player, Windows Media Center, and Windows Movie Maker."

That's not to say that there isn't material relevant to all of us. Did you know, for example, that

> "Windows Vista includes a new disk defragmenter that runs in the background... The disk defragmenter runs when the computer is idle, and takes advantage of low priority I/O to minimize the performance impact. In Windows Vista, you should never think about fragmentation – Windows Vista keeps it under control without requiring the user or administrator to do anything."[pp 14-15]

Of course, you can manually initiate defrag as with previous versions.

So... how about our data storage question? With your indulgence, I'm going to quote several paragraphs from the Resource Kit:

From page 596 onward (my comments in [brackets]):

> "Local sharing, also known as *same computer sharing*, refers to sharing files so that all users on the same computer can access them. ... Local sharing has undergone several changes in Windows Vista compared to the earlier Windows XP platform." [This "local sharing" sounds like exactly what I generally want for data on a computer – it's available regardless of who's logged on. The equivalent in Windows XP is the *Shared Documents* folder (actually a junction point) which is typically mapped to C:\Documents and Settings\All Users\Documents. By default, an ordinary XP user can write to that folder, but can't modify anyone else's data unless hs's logged on as an Administrator or Power User. But of course, with XP that's not as much of an issue. After all, how many times have you or your customers ever actually done work on an XP machine without being logged on as something above an ordinary user?

Again from the Resource Kit:]

> "In Vista, users of the same computer can still share files but the model has changed both under the hood and in terms of collaborative experience.... application and user data for the All Users profile has now been separated in Vista to provide better security. Specifically:
>
> - Application data for all users of the computer is now stored in the %SystemDrive%\ProgramData hidden folder [AKA CSIDL_COMMON_APPDATA]
> - User data for all users of the computer is now stored in the %SystemDrive%\Users\Public

> folder [CSIDL_COMMON_DOCUMENTS is inside the Public folder.]
>
> "The Public folder is new to Vista and replaces the Shared Documents folder of Windows XP. Any files that a user copies or moves to the Public folder become available to all users on the computer ... The effective result of the permissions listed in Table 17-2 [table omitted] is that users can read, execute, modify, and even delete files saved by other users to Public Documents and other subfolders of Public, regardless of whether they are standard users or administrators on the computer." ...

Even in the Resource Kit (written by committee as are all such tomes) there's a bit of contradiction on page 1456

> \ProgramData [CSIDL_COMMON_APPDATA] The standard location for applications to store non-user-specific data, such as shared settings that all users should be able to update.

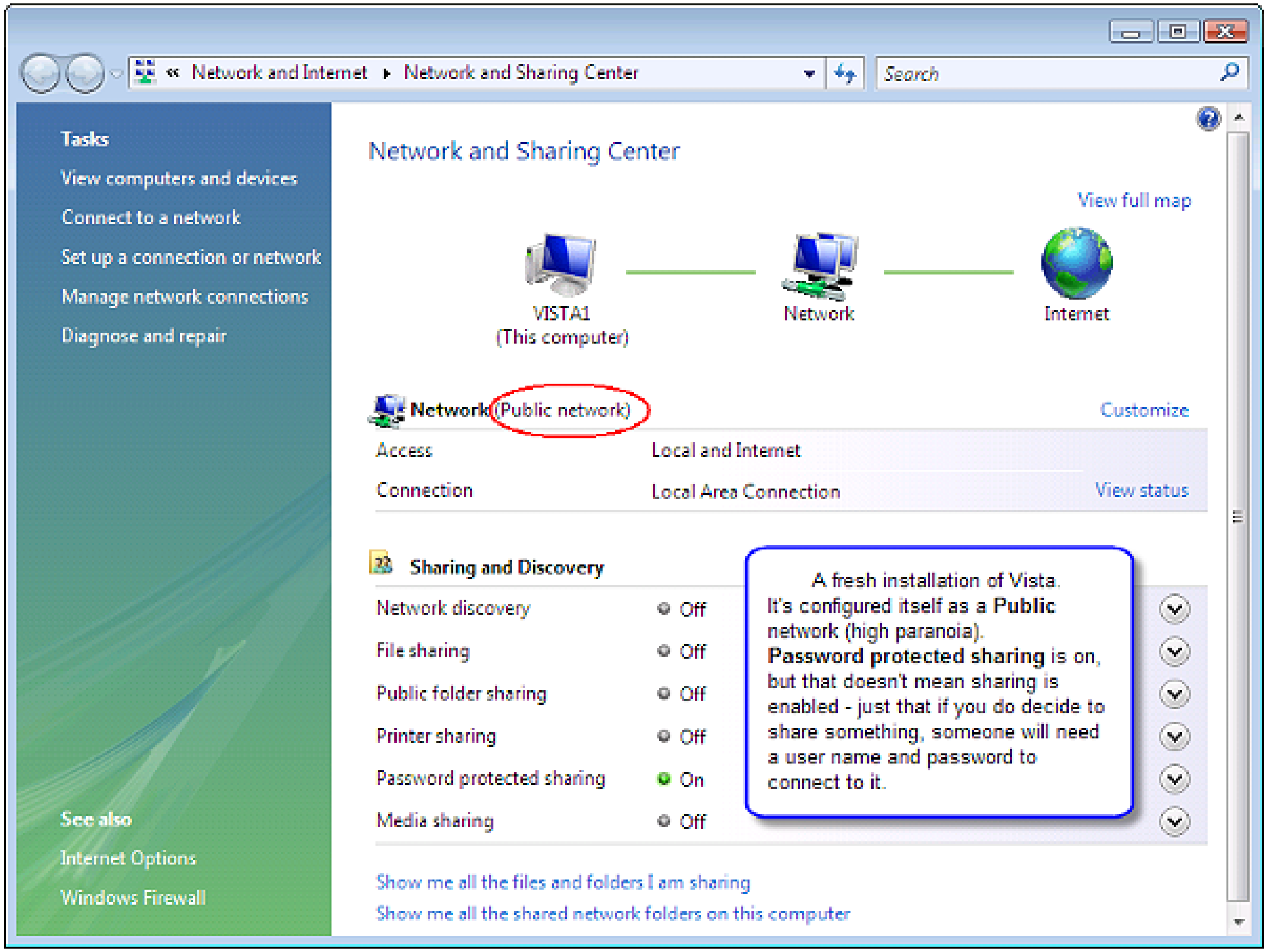
Argh! As I pointed out in a previous ClarionMag article, ordinary users can write to this (hidden) folder, but one ordinary user can't modify a file created by another user.

> **Executive Summary #2:** Default permissions dictate that shared files ordinary users need to be able to create and/or modify should go in CSIDL_COMMON_DOCUMENTS. Read-only data for your app can be stored in CSIDL_COMMON_APPDATA.

Now let's go really wild and assume that not only do I want various users logging onto one computer to use the same data files, but my app is running in a small business environment where a few other computers need to access its data. Obviously, I'm going to need to create a folder share.

Vista has a sharing wizard that's a bit different from the one in XP. As mentioned, a broader look at sharing and permissions will be the scope for a future article. Here I just want to point out a few caveats with Vista's Public structure and take a glance at Vista networking.

A freshly installed Vista box's Network and Sharing Center presents a field of thorns (Figure 7).


« Network and Internet ▸ Network and Sharing Center
Search
Tasks
View computers and devices
Connect to a network
Set up a connection or network
Manage network connections
Diagnose and repair
See also
Internet Options
Windows Firewall
View full map
VISTA1
(This computer)
Network
Internet
Customize
View status
Show me all the files and folders I am sharing
Show me all the shared network folders on this computer

Network and Sharing Center

Network (Public network)

Access: Local and Internet

Connection: Local Area Connection

Sharing and Discovery

Network discovery ○ Off

File sharing ○ Off

Public folder sharing ○ Off

Printer sharing ○ Off

Password protected sharing ● On

Media sharing ○ Off

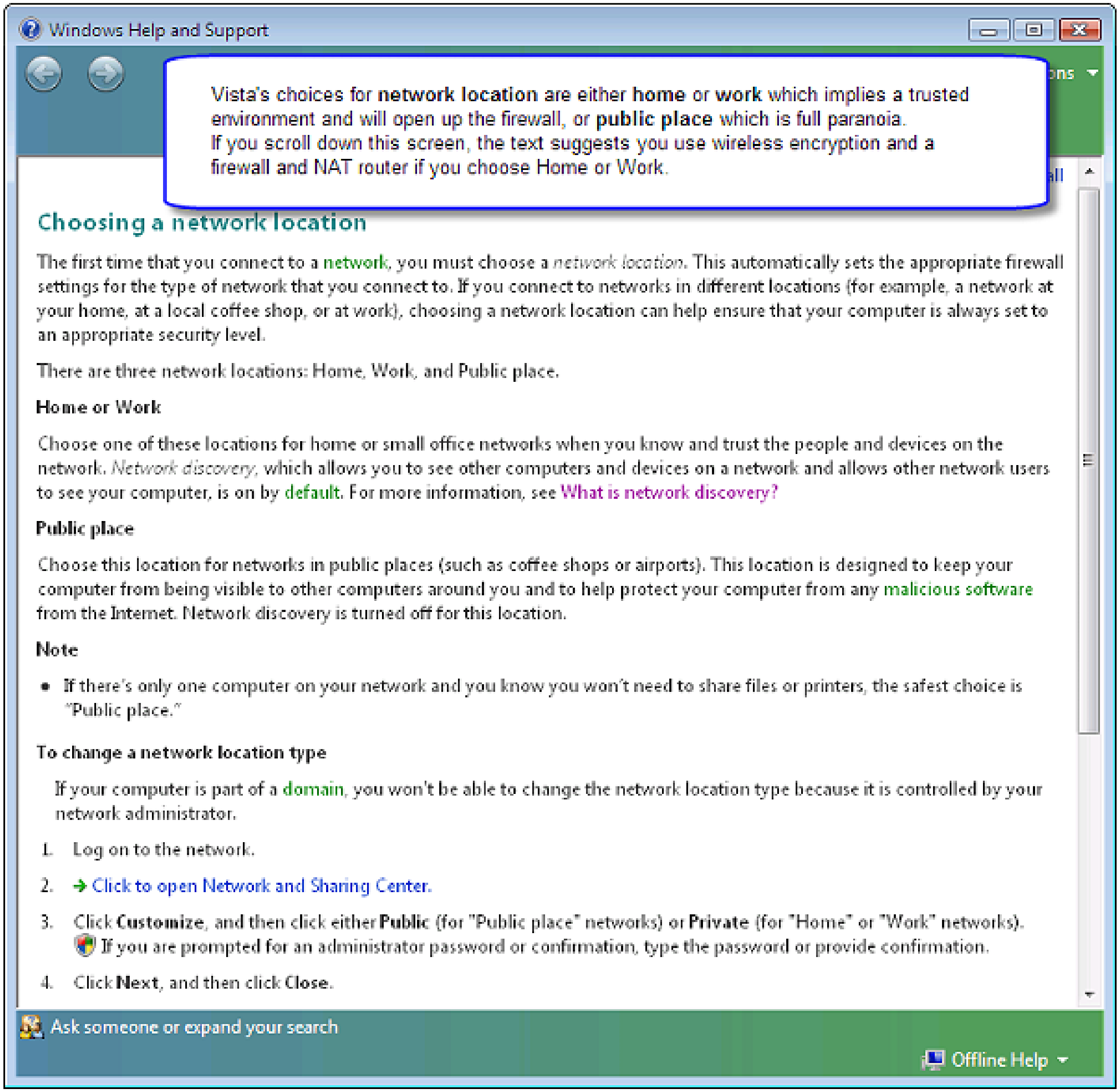
A fresh installation of Vista. It's configured itself as a **Public** network (high paranoia). **Password protected sharing** is on, but that doesn't mean sharing is enabled - just that if you do decide to share something, someone will need a user name and password to connect to it.

Figure 7. Initial network settings on a fresh Vista installation (view full size image**)**

You'll notice that File sharing and Network discovery are off. There's also a separate option for Public folder sharing. I'll go into that in a bit.

On this screen, you can configure individual items in the Sharing and Discovery list, and/or change the type of network Vista assumes it's connected to. Let's look at the network options first.

To change that setting, you'll click Customize. That screen includes a help text describing the different configurations (Figure 8).



Vista's choices for **network location** are either **home** or **work** which implies a trusted environment and will open up the firewall, or **public place** which is full paranoia.
If you scroll down this screen, the text suggests you use wireless encryption and a firewall and NAT router if you choose Home or Work.

Choosing a network location

The first time that you connect to a network, you must choose a *network location*. This automatically sets the appropriate firewall settings for the type of network that you connect to. If you connect to networks in different locations (for example, a network at your home, at a local coffee shop, or at work), choosing a network location can help ensure that your computer is always set to an appropriate security level.

There are three network locations: Home, Work, and Public place.

Home or Work

Choose one of these locations for home or small office networks when you know and trust the people and devices on the network. *Network discovery*, which allows you to see other computers and devices on a network and allows other network users to see your computer, is on by default. For more information, see What is network discovery?

Public place

Choose this location for networks in public places (such as coffee shops or airports). This location is designed to keep your computer from being visible to other computers around you and to help protect your computer from any malicious software from the Internet. Network discovery is turned off for this location.

Note

- If there's only one computer on your network and you know you won't need to share files or printers, the safest choice is "Public place."

To change a network location type

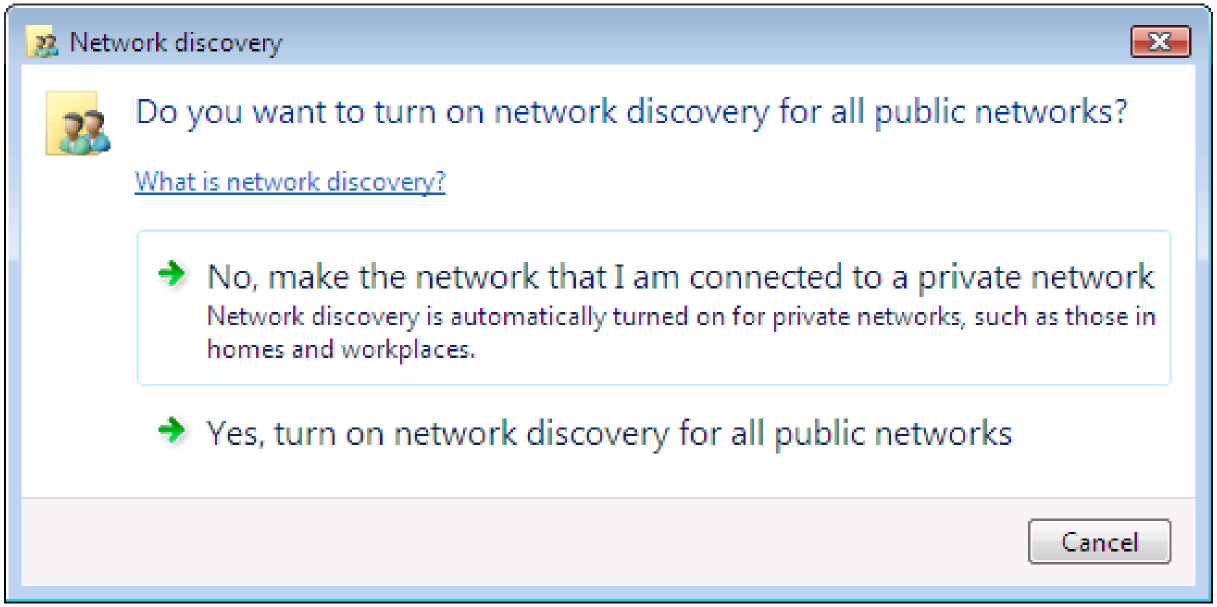
If your computer is part of a domain, you won't be able to change the network location type because it is controlled by your network administrator.

1. Log on to the network.
2. → Click to open Network and Sharing Center.
3. Click **Customize**, and then click either **Public** (for "Public place" networks) or **Private** (for "Home" or "Work" networks). If you are prompted for an administrator password or confirmation, type the password or provide confirmation.
4. Click **Next**, and then click **Close**.

Figure 8. Description of different network location choices (view full size images)

If you wish, you can roll up your sleeves and configure Vista's firewall manually with a fair amount of detail. The presets shown in Figure 8 present a broad choice between a completely insecure situation, called a "Public Place" (you're using a Wi-Fi card on your laptop at Starbucks) or a relatively safe environment, called "Home or Work" (you're on a network that has a firewall or router separating it from the Internet.)

I might as well get one other bit of terminology out of the way at this point – network discovery. Somewhat to my surprise, I found that even without enabling sharing in the screen shown in Figure 7 I could still share a folder from within Windows Explorer. But the first time I shared a folder that way (using the File Sharing wizard), a screen popped up asking me about Network Discovery (Figure 9).



Network discovery

Do you want to turn on network discovery for all public networks?

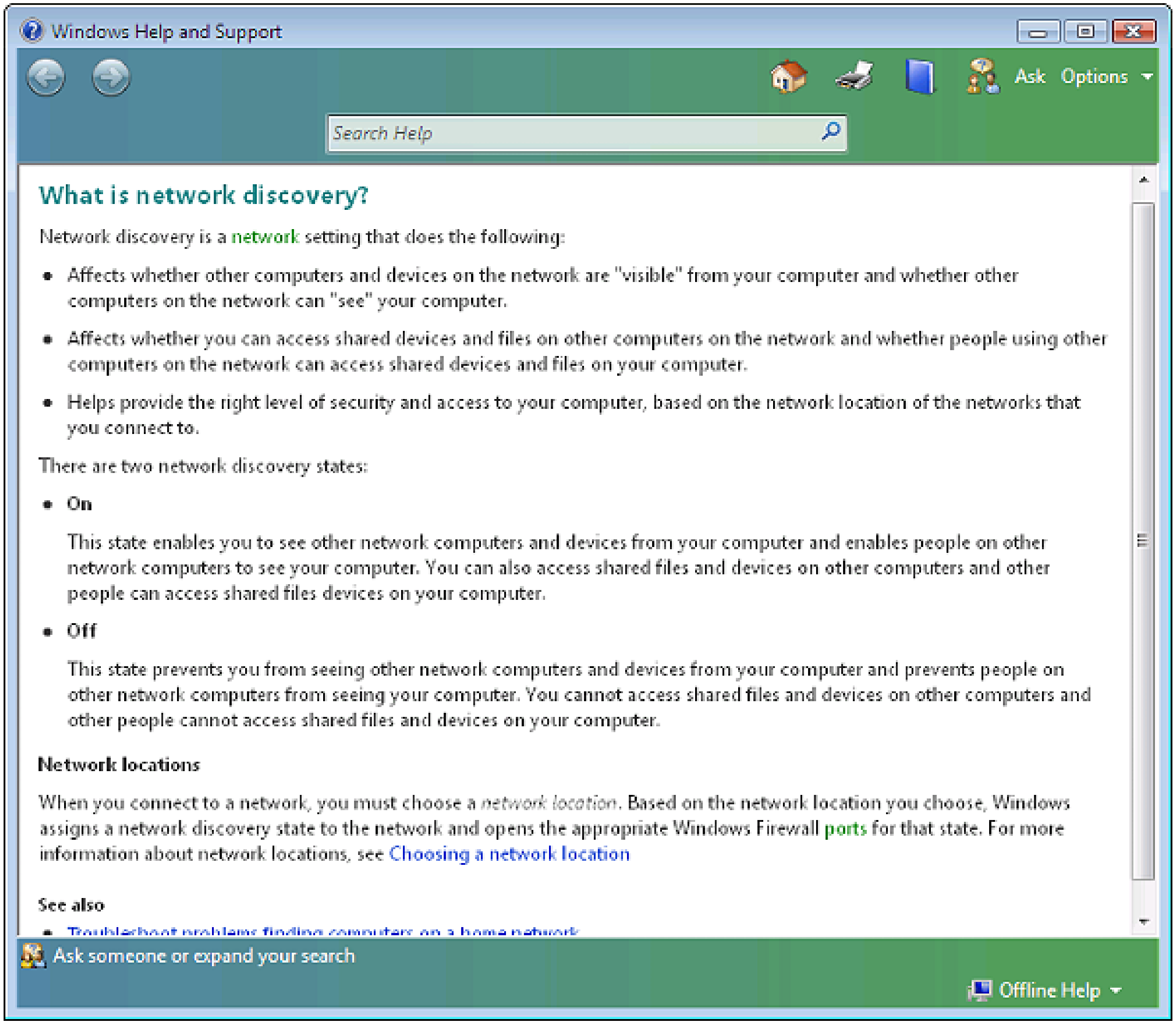
What is network discovery?

No, make the network that I am connected to a private network
Network discovery is automatically turned on for private networks, such as those in homes and workplaces.

Yes, turn on network discovery for all public networks

Figure 9. When I hadn't enabled network discovery and tried to share a folder, I got this question

The Network and Sharing Center screen links to a more detailed description of network discovery (Figure 10).



What is network discovery?

Network discovery is a network setting that does the following:

- Affects whether other computers and devices on the network are "visible" from your computer and whether other computers on the network can "see" your computer.
- Affects whether you can access shared devices and files on other computers on the network and whether people using other computers on the network can access shared devices and files on your computer.
- Helps provide the right level of security and access to your computer, based on the network location of the networks that you connect to.

There are two network discovery states:

- On

  This state enables you to see other network computers and devices from your computer and enables people on other network computers to see your computer. You can also access shared files and devices on other computers and other people can access shared files devices on your computer.

- Off

  This state prevents you from seeing other network computers and devices from your computer and prevents people on other network computers from seeing your computer. You cannot access shared files and devices on other computers and other people cannot access shared files and devices on your computer.

Network locations

When you connect to a network, you must choose a *network location*. Based on the network location you choose, Windows assigns a network discovery state to the network and opens the appropriate Windows Firewall ports for that state. For more information about network locations, see Choosing a network location

See also

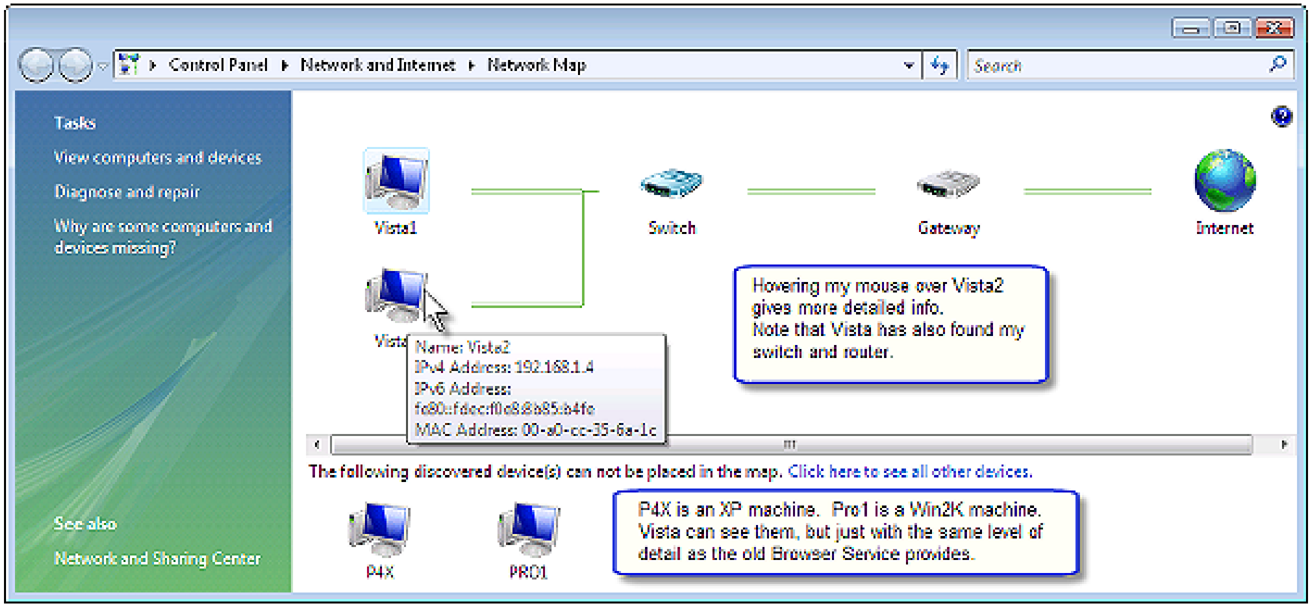
- Troubleshoot problems finding computers on a home network

Figure 10. More detailed description of network discovery

Windows networks prior to Vista have used something called the "Browser Service" to locate resources. When a computer that has shared folders or printers boots, it broadcasts a "here am I" message. Over time, Master Browsers, Backup Browsers, and Domain Master Browsers pick up those messages, consolidate them into lists, and provide the lists to populate the information you see in My Network Places or Network Neighborhood on client computers. The Browser Service is old and ugly. It can take quite a while for a freshly-booted machine to appear on lists, and a machine can still show up on lists for 45 minutes or more after it's been shut down.

Vista's Network Discovery uses a new protocol to provide information about what computers are available on the network. In addition to the simple list of the old Browser Service it also gives more information about the network and about machine details (Figure 11.)

If, by the way, you want to update your XP machines so they can enter your mapped world, there's a download from the Microsoft website to add the Link-Layer Topology Discovery driver.


Control Panel ▸ Network and Internet ▸ Network Map
Search
Tasks
View computers and devices
Diagnose and repair
Why are some computers and devices missing?
See also
Network and Sharing Center
Vista1
Switch
Gateway
Internet
Vista
P4X
PRO1

Name: Vista2
IPv4 Address: 192.168.1.4
IPv6 Address:
fe80::fdec:f0c8:8b85:b4fe
MAC Address: 00-a0-cc-35-6a-1c

Hovering my mouse over Vista2 gives more detailed info.
Note that Vista has also found my switch and router.

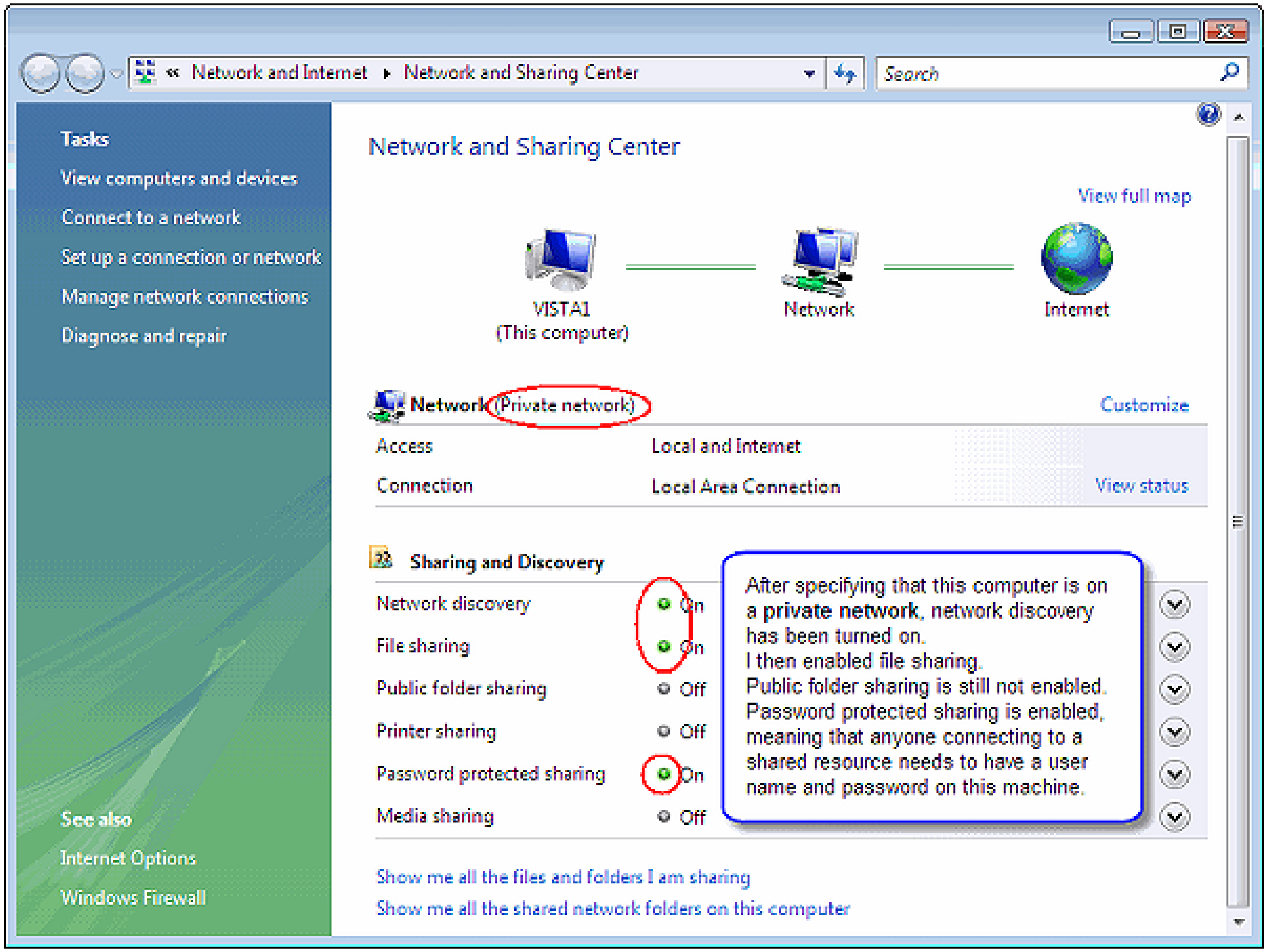
The following discovered device(s) can not be placed in the map. Click here to see all other devices.

P4X is an XP machine. Pro1 is a Win2K machine. Vista can see them, but just with the same level of detail as the old Browser Service provides.

Figure 11. Vista's Network Discovery is an update on the old My Network Places (view full size image)

Assuming that you're in a small business network and want computers to be able to access your shared data (which is why you're creating a share, right?), you'll want network discovery to be turned on.

If you use the Network and Sharing Center screen to specify that you're on a private network, it will automatically turn on network discovery. I've also turned on file sharing (Figure 12). Note that for the moment I've left Public folder sharing disabled.



Network and Internet ▸ Network and Sharing Center
Search
Tasks
View computers and devices
Connect to a network
Set up a connection or network
Manage network connections
Diagnose and repair
See also
Internet Options
Windows Firewall
View full map
VISTA1
(This computer)
Network
Internet
Customize
View status
Show me all the files and folders I am sharing
Show me all the shared network folders on this computer

Network and Sharing Center

Network (Private network)

Access: Local and Internet

Connection: Local Area Connection

Sharing and Discovery

- Network discovery: ● On
- File sharing: ● On
- Public folder sharing: ○ Off
- Printer sharing: ○ Off
- Password protected sharing: ● On
- Media sharing: ○ Off

After specifying that this computer is on a **private network**, network discovery has been turned on.
I then enabled file sharing.
Public folder sharing is still not enabled.
Password protected sharing is enabled, meaning that anyone connecting to a shared resource needs to have a user name and password on this machine.

Figure 12. Configured as private network

A word about the Password protected sharing setting. This is going to be a shock to people who became accustomed to the ease of small network sharing in the XP era. XP's default is "simple sharing", which turns on the (otherwise-disabled) guest user account and gives permissions to the Everyone group. (I'll go into this more in the article on permissions, or if you just can't wait you can Google xp forceguest .) For somebody to use a share you create on a Vista machine, however, he'll need a user account and password on that computer.

So I've enabled sharing in Figure 12. But what's special about the Public folder to warrant its having its own setting on this screen?

Well, it's possible to create a "normal" share in the Public folder hierarchy, but it takes a little finagling. I'll explain how it's done next time.

Jane Fleming is a college dropout who subsequently lived four years in Europe, a year and a half in Mexico, and three years in India, and later taught yoga for a living in California (she’s been vegetarian since 1970). She developed circuits and wrote assembly code for several embedded microcontroller projects during the 1980s. She began using Clarion Professional Developer for in-house projects back when Clarion was running display ads in InfoWorld and has used it very intermittently since. She is a former Microsoft Certified Trainer and taught Microsoft and Novell network administration at a business college for four years. Now widowed ten years, Jane plays classical piano and has found her métier as a semi-retired NRA-certified pistol instructor.

Reader Comments

Add a comment

Clarion Magazine

Showing ABC Errors

by Steven Parker

Published 2007-07-20

In days of old, when knights were bold and ABC was recently invented, many Clarion programmers found that ABC was seriously counter-intuitive. For instance,

```
Relate:Customer.Update
```

and

```
Relate:Customer.Delete
```

are valid method calls. But

```
Relate:Customer.Insert
```

is not. There is no Relate: .Insert method. There is an Access: .Insert method. There is no Access: .Append.

Similarly,

```
If ~Relate:Customer.Open()
```

or

```
If Relate:Customer.Open() = 0
```

or

```
If Relate:Customer.Open() = Level:Benign
```

all open the Customer file (well, sort of ... but that's another story). That is, a return value of zero indicates success. In the Clarion language zero is "False" and non-zero is "True" and to reverse this in ABC is most confusing.

The previous norm was:

```
Open(Customer)
If ErrorCode() ...
```

I really miss the ability to check the ErrorCode and to take specific action on it or different actions on different error codes. For example:

```
Open(Customer)
```

```
Case ErrorCode()
Of 2          ! file does not exit
  Create(Customer)
  Open(Customer)
Of 52       ! file is already open
  Message('Customer file is in use.' & |
           'Please try later.','File In ' & |
           'Use',ICON:Exclamation)
Else
  Message('Unable to open the customer file.' & |
           ErrorCode() & ': ' & Error(), |
          'Error',ICON:Hand)
  Return
End
```

In ABC, this became:

```
If Relate:Customer.Open (optionally, <> Level:Benign)
  Message( ...
End
```

But, ErrorCode() and Error() are useless (you can trust me on this for a moment or jump ahead and check out Figure 6 below). The data returned from these functions are incorrect because ABC methods, while combining the operation and an error check, return the severity of the error, not the actual error. Trying to add a record that results in a duplicate key might not throw an error at all. For example:

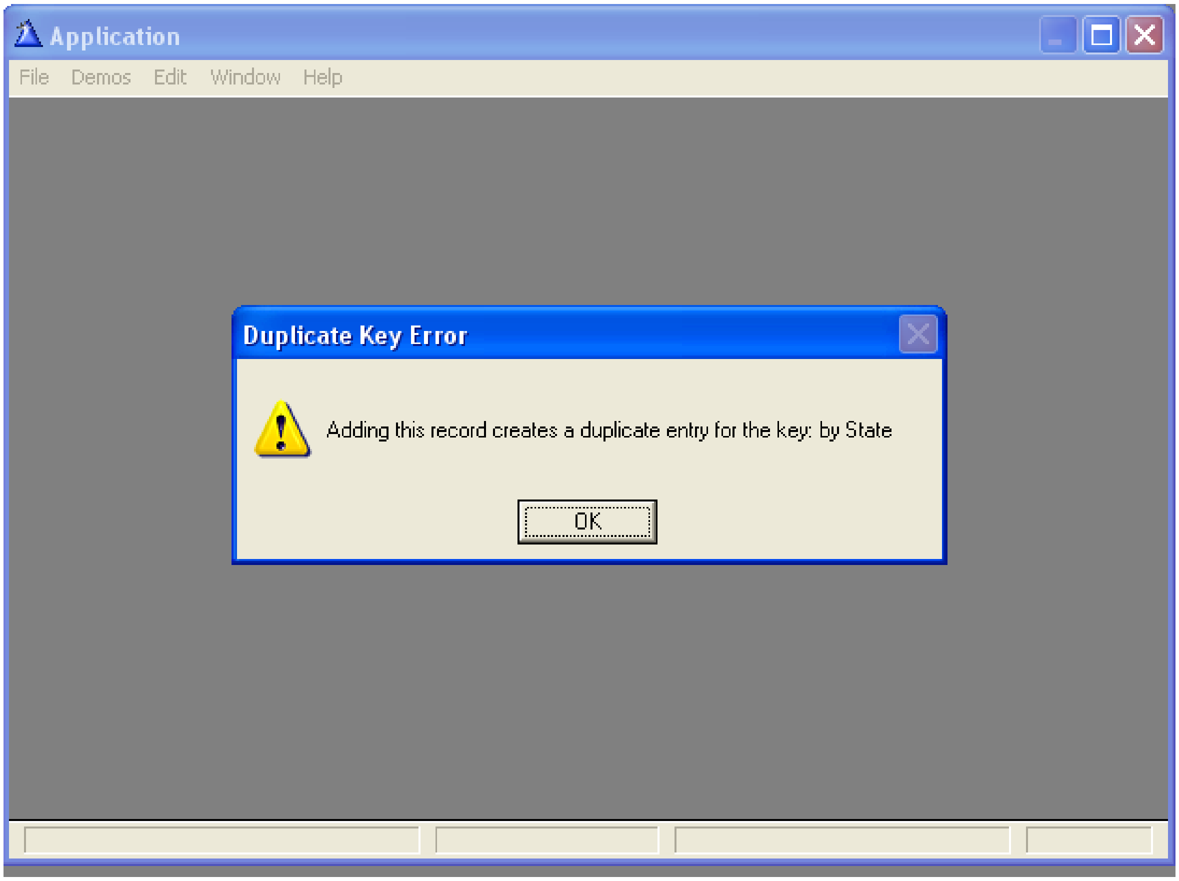
```
Relate:People.Open
PEO:SysID = 2
PEO:Surname = 'Jones'
PEO:GivenName = 'Guy'
PEO:State = 'IL'
Access:People.Insert
Relate:People.Close
```

where PEO:SysID is an auto numbered key and where there is already a "2" in the file, throws no error. Access: . Insert "corrects" the numbering. (See the 6.3 demonstration app in the downloadable source, and select Demos | Forced Error. After trying this a couple of times, check the People file in Topscan. This is a 6.3 example because not all of the required classes are available in 5.5.)

However, if there is a unique but not auto numbered key on state code, attempting the following insert:

```
Relate:States.Open
STA:State = 'AK'
Access:States.Insert
Relate:States.Close
```

does throw an error.

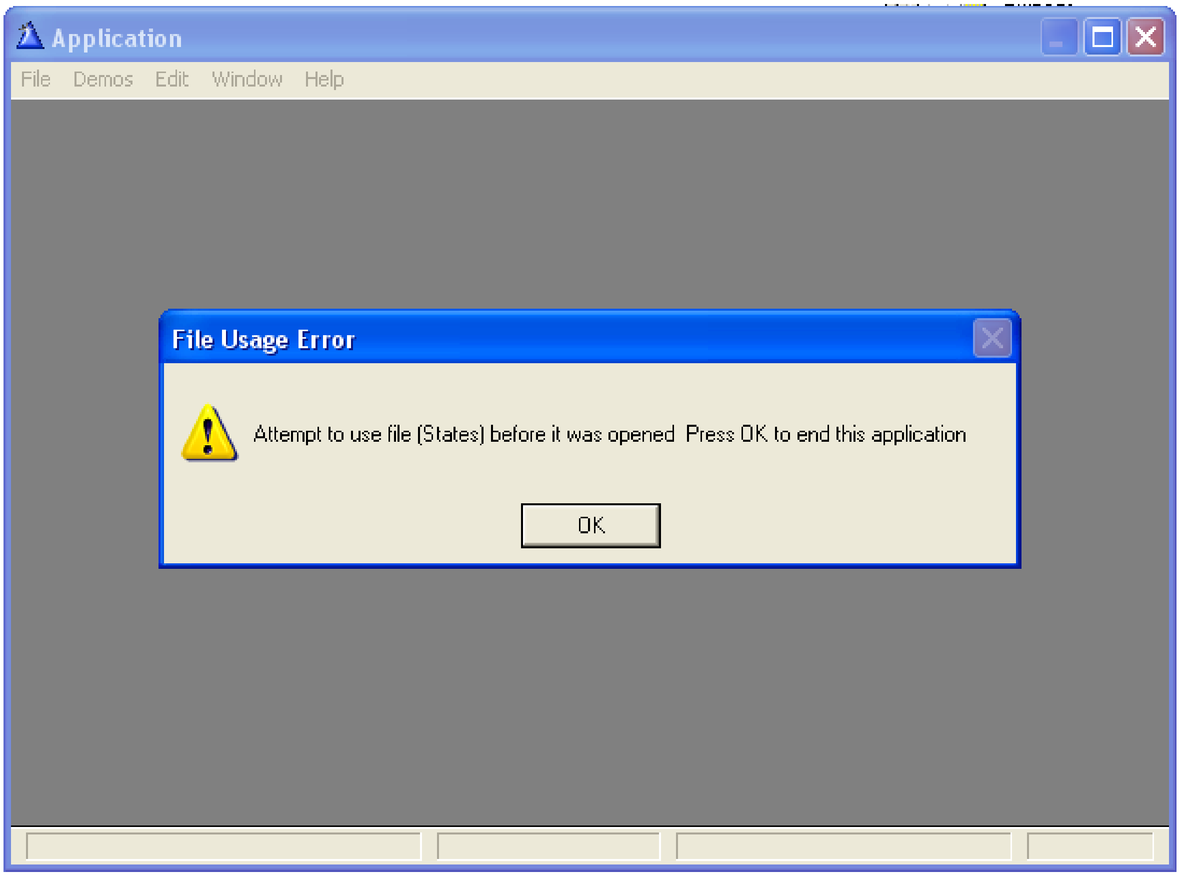


Duplicate Key Error

Adding this record creates a duplicate entry for the key: by State

Figure 1. Default ABC error message for Duplicate Key

If the file is not open, I also get a message. Note that these messages share the feature that they are not easily understood by my users:



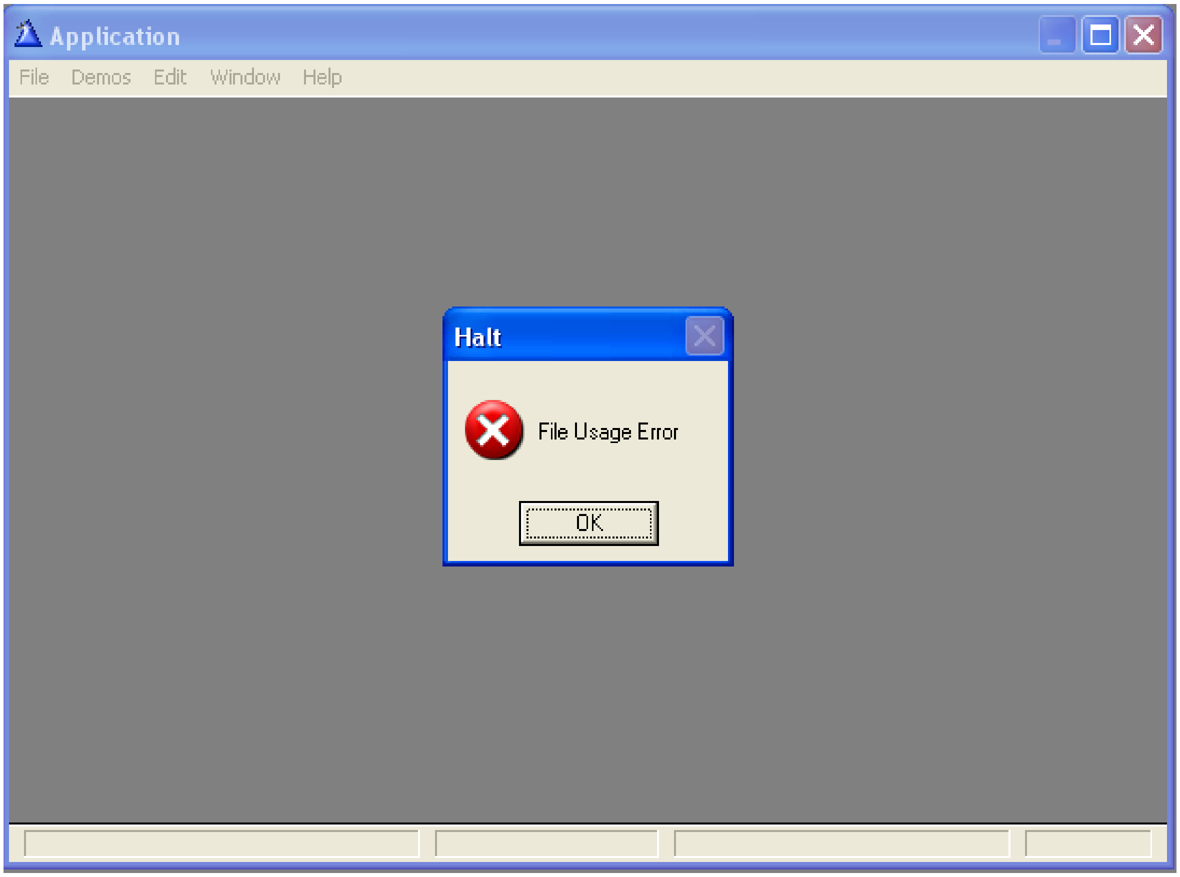
File Usage Error

Attempt to use file (States) before it was opened Press OK to end this application

OK

Figure 2. ABC error on Insert when file is not open

But the error message in Figure 2 is immediately followed by Figure 3, and the application halts. .



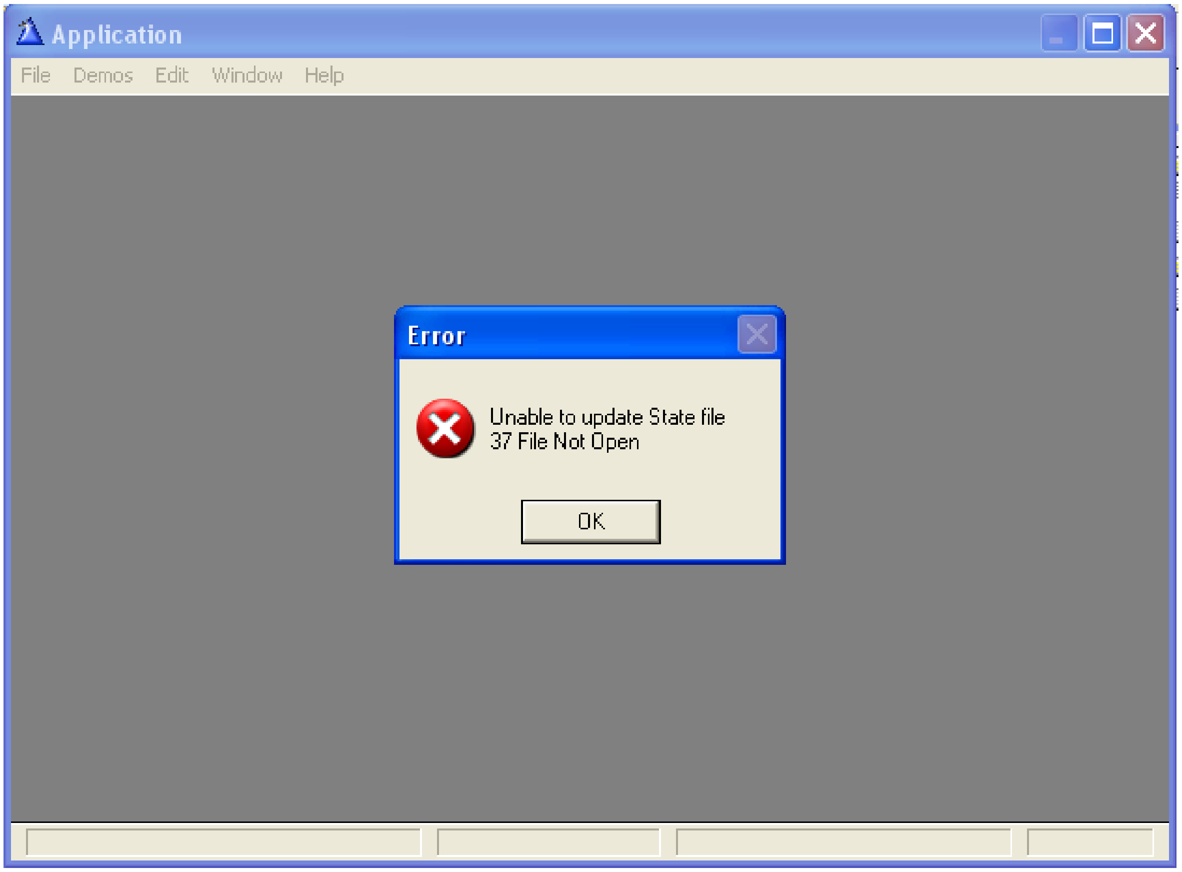
Halt

File Usage Error

OK

Figure 3. Follow on message

Somebody made a "severity decision" that doesn't seem to quite suit me. Among other things, I want to choose what action is taken. All I want is a single understandable message, as in Figure 4. without application termination:.



Unable to update State file
37 File Not Open

Figure 4. Desired message

I want to configure the wording for my target audience and, by using unique wording, make the source of the error easy for me to find. This code produces exactly the effect I want:

```
Add(States)
If Errorcode()
  Message('Unable to update State file <13,10>' &|
    Errorcode() & ' ' & Error(),'Error',ICON:Hand)
  Return
End
```

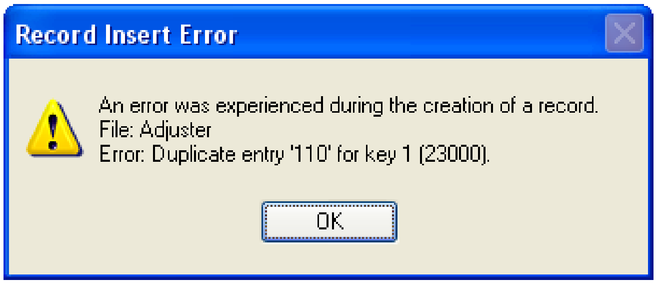
This message window gives a fairly clear indication of the error and it supplies the error code. This provides information I can use to locate where and why the error happened. And it doesn't terminate my app (unless I want it to).

Ideally I could just adapt the codelet above to my ABC code :

```
STA:State = 'AK'
If Access:States.Insert()
  Message('Error adding a state.||The errorcode is ' &|
           ErrorCode() & ' the error is ' & |
          Error(),'Warning Will Robinson',ICON:Hand)
     Return
    End
```

In the demo app, press Demos | "ErrorCode" in ABC (the procedure is Force4) and you will see that the results are exactly the same as if I had put in no message at all (as in the Force3 procedure which produced Figures 2 and 3). In fact, my message never appears (I did not open the file, so one would think that the message would appear).

Using SQL, the default error messages become even less informative:



Record Insert Error

An error was experienced during the creation of a record.
File: Adjuster
Error: Duplicate entry '110' for key 1 (23000).

Figure 5. A SQL duplicate key error

In more complex tables, I have seen the message in Figure 5 run on for 15 or more lines. And, if that is not enough, the key number is not always correct.

This apparent loss of control led me to continue using "Legacy" I/O statements (Add, Put, Delete) in my hand coded and embedded I/O. I was not alone; many did so. The *only* caveat, we were informed, was that while we lost ABC's built in error handling (no great thing, as many of us were not very impressed with it anyway), one must never, never, ever, ever mix ABC and "Legacy" file opening/closing. The code,

```
    Relate:Customer.Open
    Open(Customer,42h)
```

followed, later, by an ABC Close could, would and did lead to very nasty bugs in later versions of Clarion. Over the years, the compiler has been less and less forgiving of shortcuts and mistakes as "Legacy" opens and closes were made to update the internal counters. This killed the comfort of adding an Open(*file*) or a Close(*file*) for "insurance" (all you C5b survivors out there should remember this). Doing this nowadays can cause template generated opens and closes to fail because the internal opened/closed counters are "off." Such problems are almost impossible to debug.

Continuing to rely on Legacy I/O made it possible to keep the old error handling and reporting. It is comfortable, it works, but my new employer insists on ABC consistency. So, here I am, back with the error handling issues highlighted above (and since my new boss is a Clarion Mag subscriber, I certainly would never assert that I use Legacy I/O in *his* apps).

If you've read the docs or lurked in the newsgroups, you should immediately realize that I have made a fundamental error. The methods I have been using combine error handling into the base operation. I should be using the *Try* variants, which do not.

```
    Access:States.Insert()
```

is logically, though not functionally, equivalent to

```
    Add(States)
    If ErrorCode()
     Message('There was an error')
    End
```

(Actually, the messages for duplicate key and file not open, shown in Figures 1 and 2, are moderately specific. Other errors may not be – see abfile.clw around line 653.)

On the other hand,

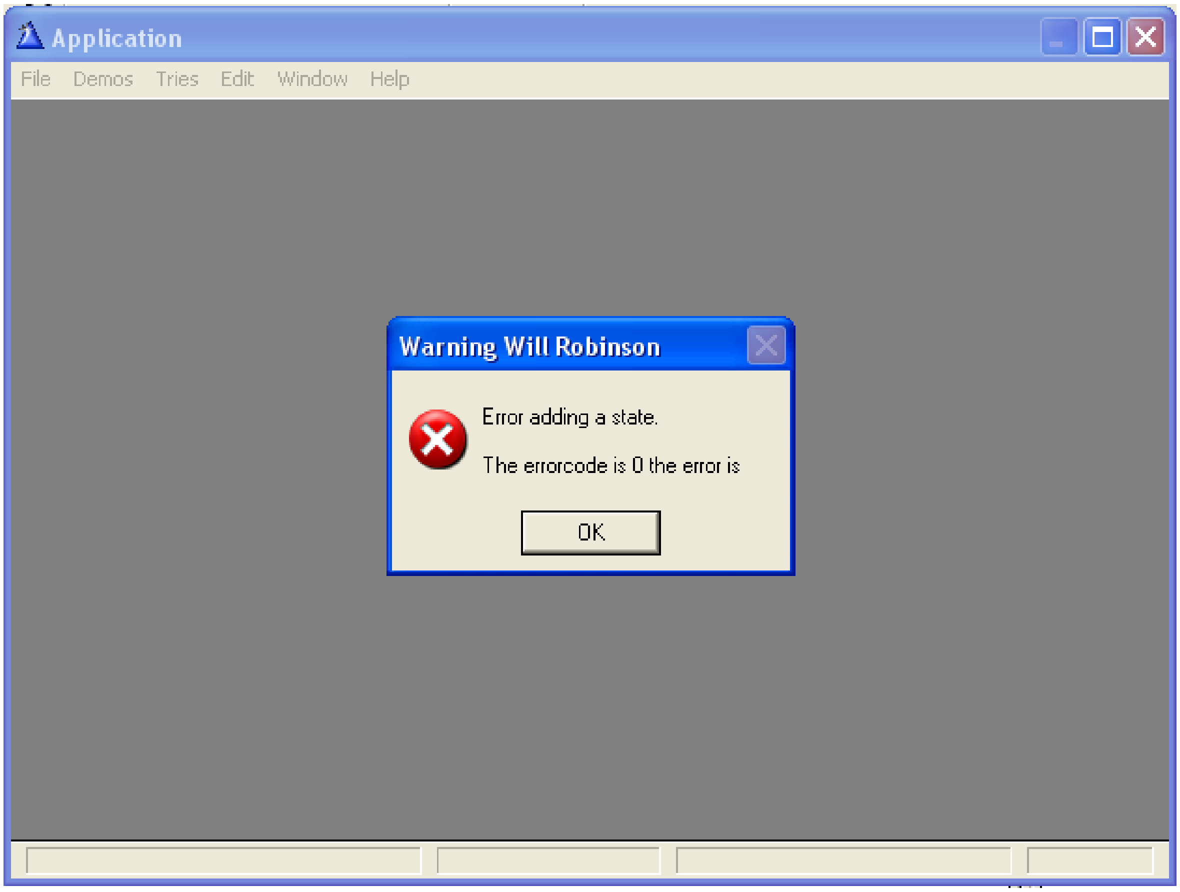
```
    Access:States.TryInsert()
```

is logically equivalent to

```
    Add(States)
```

In the demo app, there is a procedure (Tries | Try Force) which attempts to force a duplicate key error. Try it. It doesn't report anything. And that is precisely what it is supposed to do! The Try variants do not attempt to deal with errors that may occur. That means that I should be able to check for errors myself and write my own error handling code. And so it is.

In the demo app Tries | Try Force Message demonstrates that my message will appear now (the default method, as in Figure 1 or 2 does not show mine). However, error information is still unavailable (as promised earlier).



Error adding a state.

The errorcode is 0 the error is

Figure 6. Message generated in a TryInsert

It is simply inconceivable to me that ABC I/O entirely throws out the ErrorCode. A bit of detective work shows that, in fact, ErrorCode is stored. It is stored but it is buried. Searching through the ABC source files show that the parent Insert and TryInsert methods are in abfile.clw.

TryInsert calls InsertServer. InsertServer calls Usefile, ValidateRecord, HasAutoInc and something called SetError.

Eureka! SetError, is in aberror.clw.

Examining the code in SetError shows that it stores the ErrorCode and Error in a retrievable form. (You can also search for ErrClarion or back trace the InsertServer method's code)

First, I need to use SetErrors() to prime the class' variables. then GetErrorCode() and GetError() to retrieve the desired information:

```
If Access:hRESERVES.Insert()
  ThisWindow.Errors.SetErrors()
  Message('Error getting the matching claim.' & '<13,10>' &|
    ThisWindow.Errors.GetErrorCode(ErrClarion) & ': ' &|
    CLIP(ThisWindow.Errors.GetError(ErrClarion)), |
   'Error',ICON:Hand)
End
```

It is worth noting that SQL back ends often return errorcode 90. These three methods will translate the errorcode 90 into "Clarion" (a human readable form).

But wait!

If you examine the demo app, you will notice that most of my procedures are source procedures. But the one that uses SetErrors, GetErrorCode and GetError is a Window procedure (TryForceFinal).

The reason for this is that the templates provide an ErrorManager for Window procedures. The Source template does not.

The relevant template code is:

```
! Enter procedure scope
GlobalErrors.SetProcedureName('procName')
! [Priority 2500]
! Snap-shot GlobalRequest
SELF.Errors &= GlobalErrors     ! Set this windows
         ErrorManager to the global ErrorManager
! [Priority 5300]
```

And, at the end of the procedure:

```
! Leave procedure scope
GlobalErrors.SetProcedureName
! [Priority 9800]
```

Unfortunately, I couldn't just add this code to a Source procedure. There appear to be a few more steps require to set up an ErrorManager in a Source procedure. At this point, however, I don't feel the need to take this investigation that much further.

Conclusion

Is there anything inherently wrong in using Legacy I/O statements in the middle of an ABC template procedure? No. Is there anything inherently wrong in mixing Legacy I/O statements with ABC file access methods? No. As long as you do not mix open/close methods, there will be no ill effect.

Will you lose anything by mixing ABC and Legacy approaches? No.

But, if you are not familiar with "the old ways" or if, like my boss, you want your code consistent, the biggest roadblock has always been getting good error information.

I could have consulted the newsgroups or my own knowledge base (Scott Ripley, Dennis Evans and John Dunn have FAQs relative to this subject and these gentlemen turned me on to SetErrors) but, when all else fails, I have found that tracing the relevant ABC methods can bear fruit.

Download the source

Steve Parker started his professional life as a Philosopher but now tries to imitate a Clarion developer. He has been attempting to subdue Clarion since version 2007 (DOS, that is). He reports that, so far, Clarion is winning. Steve has been writing about Clarion since 1993.

Reader Comments

Posted on Sunday, July 22, 2007 by John Dunn

Hi Steve,

If you want to get backend error info from a SQL access error you can use the 'ErrFile' flag instead of, or in addition to, the 'ErrClarion' flag.

In the following code snippet, if an error occurs on the fetch, I display the Clarion error code, error description and SQL (backend) error code and error description.

```
IF Access:Document.Fetch(Document:PK_DocumentID) ~= LEVEL:Benign
  ThisWindow.Errors.SetErrors()
  bErrorLevel   = LEVEL:Fatal
  sErrorMessage = 'Document Fetch Error: '           & |
  ThisWindow.Errors.GetErrorCode(ErrClarion) & ' - ' & |
  CLIP(ThisWindow.Errors.GetError(ErrClarion)) & '|' & |
  ThisWindow.Errors.GetErrorCode(ErrFile) & ' - '    & |
  CLIP(ThisWindow.Errors.GetError(ErrFile))
 BREAK
END ! if
```

John

Posted on Sunday, July 22, 2007 by Steven Parker

Good addition. Thanks.

Posted on Friday, August 17, 2007 by Joe Snyder

Steve

You might try this format. Seems to work for me.

```
 if Access:Allocation.TryUpdate() <> Level:Benign
    Message('Allocation Completed Update error=' & Access:Allocation.GetError())
```

Add a comment

Clarion Magazine

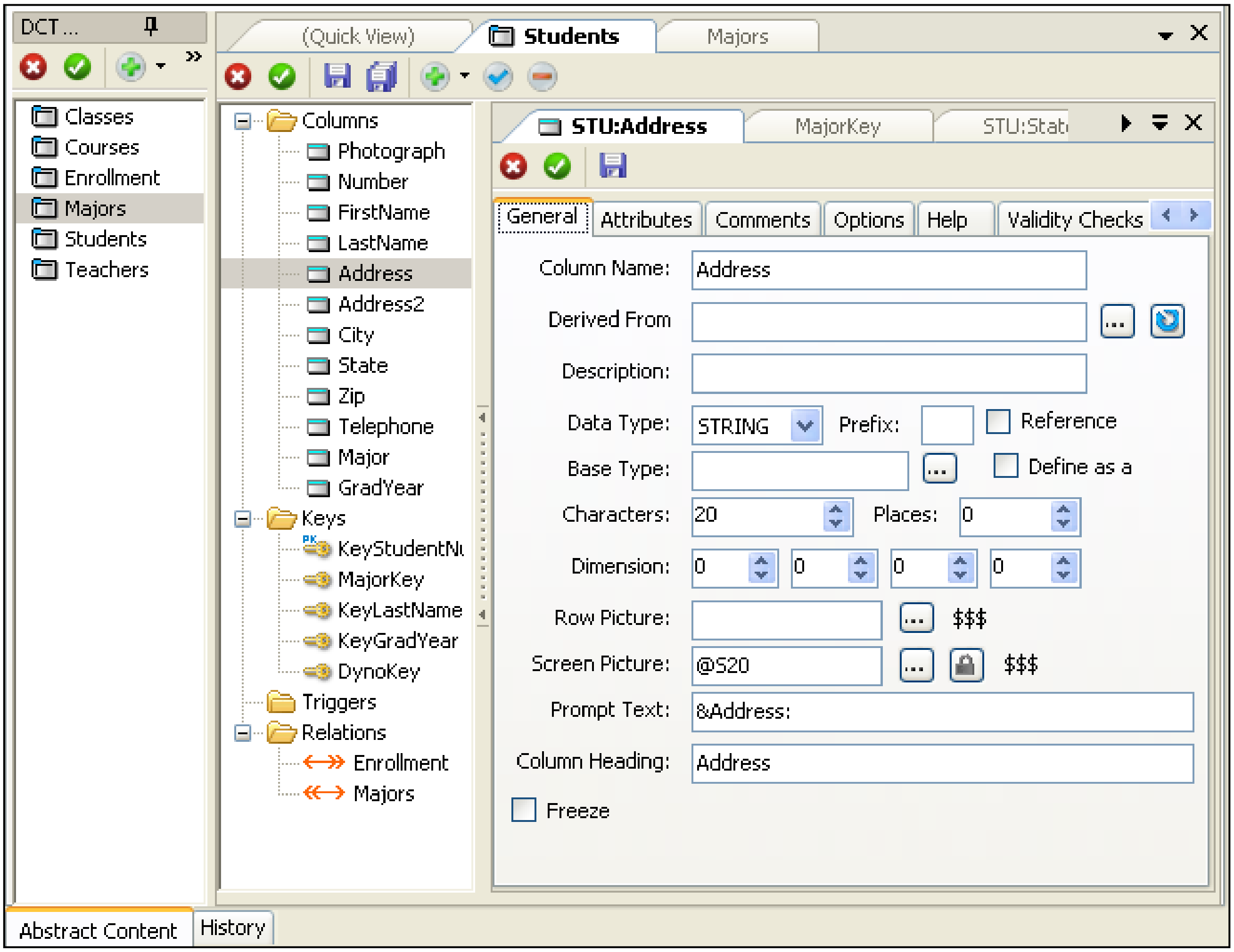
New Alpha 2 Build Improves DCT UI

by Dave Harms

Published 2007-07-13

SoftVelocity has released a new alpha phase two build with significant changes to the dictionary editor user interface. As with the previous phase two build, this is a read-only version of the editor, although older dictionaries opened in the editor are converted and saved in the new format.

The big change in this release has to do with the use of tabs to flatten the IDE. Several alpha testers (okay, it was Lee) complained of too many tabs in the previous release, as shown in Figure 1. You can see three rows of tabs: one for the tables, one for the columns, and a third for each set of prompts.



Column Name: Address

Derived From:

Description:

Data Type: STRING Prefix: □ Reference

Base Type: □ Define as a

Characters: 20 Places: 0

Dimension: 0 0 0 0

Row Picture: $$$

Screen Picture: @S20 $$$

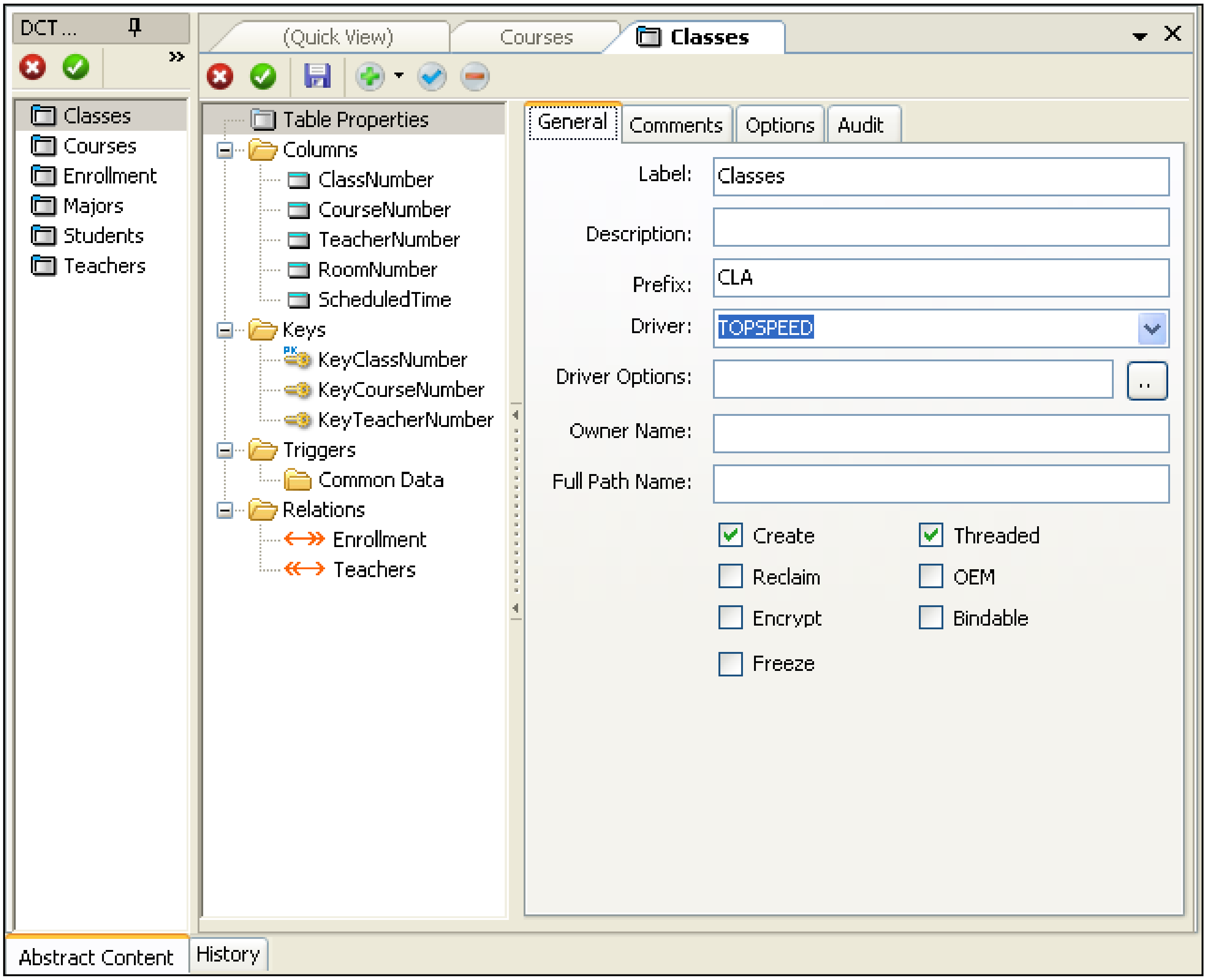
Prompt Text: &Address:

Column Heading: Address

□ Freeze

Figure 1. The Dictionary Editor in the first release (showing column properties)

In the most recent release the second row of tabs has been removed (Figure 2). As you might recall, that second row showed which columns had been opened for editing, so it wasn't simply a duplicate of the column list on the left side.



Label: Classes

Description:

Prefix: CLA

Driver: TOPSPEED

Driver Options:

Owner Name:

Full Path Name:

☑ Create ☑ Threaded

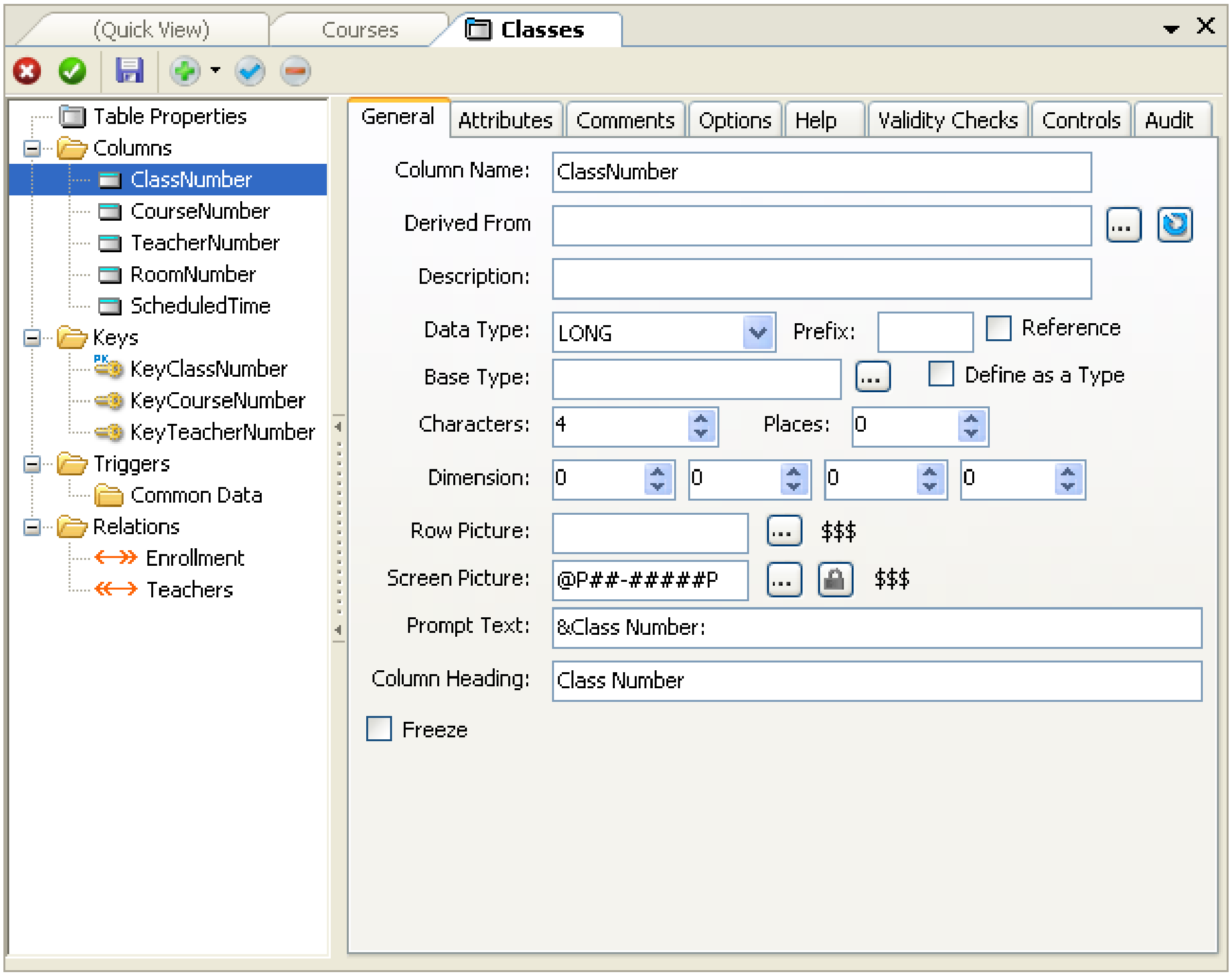
□ Reclaim □ OEM

□ Encrypt □ Bindable

□ Freeze

Figure 2. The Dictionary Editor in the second release (showing table properties)

You might also recall that in the first release you had to double-click on a column to display its tab and edit the various settings. In this release you still have to double-click on a table to select it for editing, but once the table is up (and easily selectable via the row of tabs) all you have to do to edit a column is select it in the column list (see Figure 3 – and note that to save space I've omitted the list of tables down the left side).



Column Name: ClassNumber

Derived From:

Description:

Data Type: LONG Prefix: ☐ Reference

Base Type: ☐ Define as a Type

Characters: 4 Places: 0

Dimension: 0 0 0 0

Row Picture:

Screen Picture: @P##-#####P

Prompt Text: &Class Number:

Column Heading: Class Number

☐ Freeze

Figure 3. Editing columns in alpha two

I suppose one thing has been lost in this change, and that is the ability to easily navigate through a small subset of columns. But I doubt that's going to be an issue for most developers, and the ease of single-click navigation and the cleaner look of the interface make this small loss worthwhile.

There's one other small quirk of the new system, and that's related to the table list. Double-clicking a table makes it active for editing, and single-clicking a table simply selects it in the Quick View tab. If you're not on the Quick View tab then it may appear that single-clicking a table does nothing. But switch to Quick View and you'll see that table's columns, relations and keys displayed.

It's hard to see a way around this apparent inconsistency between selecting tables and columns. On a very small dictionary you could perhaps get away with a single-click acting like a double click on the tables list, but on a large dictionary you'd quickly open a great many more tables than you really wanted to open. I tend to work on just one or several tables at a time, and having each table in its own tab, and no other tables open, makes for easier navigation. And while the double-click for tables may seem inconsistent with the single-click for columns, it's completely consistent with the double-click for opening files/items listed in the project tree.

Besides the UI changes, this release includes a number of bug fixes as well as some changes to the database browser. Interestingly, the IDE's bin directory contains a DLL which appears to be the AppGen (it may have been there in earlier alphas as well), but that functionality is not enabled in this release. Next on the schedule (besides a read/write dictionary editor) is the new data diagrammer, after which we should see the new AppGen.

David Harms is an independent software developer and the editor and publisher of Clarion Magazine. He is also co-author with Ross Santos of *Developing Clarion for Windows Applications*, published by SAMS (1995), and has written or co-written several Java books. David is a member of the American Society of Journalists and Authors (ASJA).

Reader Comments

Posted on Tuesday, July 17, 2007 by Peter Gysegem

Any idea whether there will be cut/copy and paste for tables and fields? I use this functionality often and would hate to see it go away.

Posted on Wednesday, July 18, 2007 by James Lishman

In the current DCT system, if you have a lot of fields (in excess of 60) the system crashes on copy and paste of a table.
I hope this is fixed in the new one!

Posted on Wednesday, July 18, 2007 by Dave Harms

These two builds have been read-only so SV can focus on UI improvements. I expect copy and paste will get a good workout when we get a fully functional build.

Dave

Posted on Tuesday, July 31, 2007 by Desmond Yaxley

I agree that it has a cleaner look and the double click vs single click is purely habit. Will there be a graphical display of the dict showing all relations.

Posted on Tuesday, July 31, 2007 by Dave Harms

Desmond,

I expect the next build to be released to alpha testers will include the new data designer, which will let you view and design databases with a graphical interface.

I don't know when the next build will arrive, but I'll have a report on the new data designer as soon as possible.

Dave

Posted on Sunday, August 19, 2007 by Stu Andrews

Dave,

Thanks very much for keeping us up to date with the releases of C7.

Cheers,

Stu

Add a comment

Clarion Magazine

Profiling Your Applications

by Philip Prohm

Published 2007-07-11

Profiling is the process of finding out how quickly (or slowly) the various parts of your program run. But as important as profiling is, it shouldn’t be your first thought. Premature optimisation is a bad idea; you should definitely concentrate on getting the code right before worrying about how fast it runs.

However, let's assume you have satisfactorily working code and it's just not running as fast as you'd like. Well, then it's time to identify the bottleneck. But how? Basically you need to find a way to plug in some of your own code to determine elapsed time.

Clarion has hooks called EnterProc and LeaveProc, which are compiled into your app subject to a project define called profile. Here’s a MAP statement showing the function declarations:

```
MAP
  EnterProc(UNSIGNED Line, *CSTRING Proc, *CSTRING File), |
        NAME('Profile:EnterProc')
  LeaveProc(), NAME('Profile:LeaveProc')
END
```

If you define these two procedures and link them into your program they will be called by the Clarion runtime library every time a procedure is entered or exited, *if* profile is defined when that procedure is called.

To enable profiling on a global basis, choose Project|Edit, click on Properties, choose the Defines tab, and add the define

```
profile=>on
```

For finer granularity you can turn profiling on and off in source code on a per-procedure basis via pragmas. Add

```
pragma('define(profile=>on)')
```

directly before the procedure’s source code and

```
pragma('define(profile=>off)')
```

directly after the procedure’s source code.

You can place these two pragmas anywhere that suits you; you might even want to profile just one specific procedure. It is very important, however, to turn profiling off before the definition of EnterProc and LeaveProc and not turn it back on until after, or you’ll cause an infinite loop because entering EnterProc will induce a call to EnterProc!

What is profiling?

I've been talking about profiling in a general way, but what does it actually mean? In its simplest form, profiling is counting. For example, you could count the number of times a procedure (or every procedure) is called. This is what the out-of-the-box profiling code does. Clarion ships with a source file called profile.clw – you can find it in your libsrc directory.

To try out the standard profiler go to Project|Edit and add profile.clw under External source files. Then select profile.clw and click on Properties. On the Defines tab add the text profile=>off to disable profiling for this source module. Don’t forget to add profile=>on to the Defines tab of the project as a whole.

Here’s the code in profile.clw:

```
MEMBER
  MAP
    EnterProc(UNSIGNED Line,*CSTRING Proc,*CSTRING File),|
      NAME('Profile:EnterProc')
    LeaveProc(),NAME('Profile:LeaveProc')
  END

Q QUEUE
Line  UNSIGNED
File  STRING(40)
Proc  STRING(40)
Used  LONG
  END


C CLASS
Destruct PROCEDURE
  END

COMPILE('***',profile)
 Don't try and profile this, you get infinite recursion!
 Set profile=>off on this module in the project
***

EnterProc PROCEDURE(unsigned Line,*cstring Proc,*cstring File)
  CODE
    Q.Line = Line
    Q.File = File
    Q.Proc = Proc
    GET(Q,Q.Line,Q.File)
    IF ERRORCODE()
      Q.Used = 1
      ADD(Q,Q.Line,Q.File)
    ELSE
      Q.Used += 1
```

```
   PUT(Q)    ! Does not change sort order
  END

LeaveProc PROCEDURE()
 CODE
  ! Could put timing info in here

C.Destruct PROCEDURE
P FILE,DRIVER('ASCII'),CREATE,NAME('Profile')
R RECORD
B STRING(100)
 END
 END
I   LONG,AUTO
 CODE
  SORT(Q,-Q.Used)
  CREATE(P)
  ASSERT(~ERRORCODE())
  OPEN(P)
  ASSERT(~ERRORCODE())
  P.B = ' TopSpeed Profile Output :' &FORMAT(TODAY(),@D2)
  ADD(P)
  P.B = LEFT('Invoked',10)&LEFT('Procedure',30)|
     &LEFT('File',30)&LEFT('Line',10)
  ADD(P)
  P.B = LEFT('=======',10)&LEFT('=========',30)|
     &LEFT('====',30)&LEFT('====',10)
  ADD(P)
  P.B = ''
  ADD(P)
  LOOP I = 1 TO RECORDS(Q)
   GET(Q,I)
   P.B = LEFT(Q.Used,10)&LEFT(Q.Proc,30)|
      &LEFT(Q.File,30)&LEFT(Q.Line,10)
   ADD(P)
  END
  CLOSE(P)
```

Notice the COMPILE statement which will trigger a compiler error if profiling is enabled for this module. As well, LeaveProc is a stub and only EnterProc is used since the code only tracks the number of procedure calls. This source dumps the queue data to a text file when the program exits.

Here are the first few lines from the example SCHOOL.APP with profile.clw included:

```
TopSpeed Profile Output : 6/18/2007
Invoked   Procedure                File                Line
=======   =========                ====                ====


60      GETFILEID(FILESMANAGER,FILE)  ABFILE.CLW                399
44      KILL(CRITICALPROCEDURE)      CWSYNCHC.CLW                28
24      KILL(FILEMANAGER)            ABFILE.CLW                1137
22      CONSTRUCT(CRITICALPROCEDURE)  CWSYNCHC.CLW                9
22      DESTRUCT(CRITICALPROCEDURE)   CWSYNCHC.CLW               14
22      INIT(CRITICALPROCEDURE,ISYNCOBCWSYNCHC.CLW               19
14      TAKEWINDOWEVENT(WINDOWMANAGER)ABWINDOW.CLW               204
14      TAKEEVENT(WINDOWMANAGER)      ABWINDOW.CLW               703
```

If all procedures had the same execution time, simply counting the number of calls would be ample. However, some procedures consume more CPU cycles than others, particularly the ones that involve disk access. If you want to count the total time spent in each procedure you need to do more than simply count invocations.

Getting procedure timings

You could use CLOCK for timings, but CLOCK will only give you a resolution of 1/100th of a second at best. This is where QueryPerformanceCounter comes in. QueryPerformanceCounter is a Windows function which retrieves the current value of the Windows high-resolution performance counter. If the installed hardware doesn't support a high-resolution counter the function will return zero (the minimum requirement for the function is Windows NT 3.51 or later, or Windows 95 or later). You can call GetLastError() to find out what went wrong.

By taking a snapshot of this counter when entering a Clarion procedure and again when leaving it, you can calculate the time spent in that procedure via the difference between the two values. To get the value of the counter you call the QueryPerformanceCounter API call. To convert this counter value to a time, you call QueryPerformanceFrequency which returns the frequency of the counter in counts per second, and you divide the counter by the frequency.

The code presented herein is based on PROFILE.CLW as supplied by SoftVelocity, meaning constructs in the original work are retained to a greater or lesser extent. You also implement this profiler in exactly the same way, by including it in the project and setting the appropriate defines (see the discussion above).

The first thing you need is a variable to hold the timing value. QueryPerformanceCounter receives a pointer to a LARGE_INTEGER variable. In Clarion, this is declared as:

```
LARGE_INTEGER   GROUP,TYPE
LowUlong         ULONG
HighLong         LONG
                END
```

Note the TYPE attribute on the GROUP. This is how to declare a variable of type LARGE_INTEGER within the class:

```
C               CLASS
Counternow_li      group(LARGE_INTEGER).
                END
```

The use of a class is not overly important per se but is used in the original work to exploit the fact that the destructor is called automatically when the class is destroyed as the program is shutting down. If you want to profile and report more than once without terminating and restarting your program, you will need to reorganize the code appropriately.

In Clarion the API calls are declared as follows:

```
MAP
 module('winapi')
  QueryPerformanceCounter(ulong ptrLARGE_INTEGER),BOOL,|
     PASCAL,name('QueryPerformanceCounter')
  QueryPerformanceFrequency(ulong ptrLARGE_INTEGER),|
     BOOL,PASCAL,name('QueryPerformanceFrequency')
  GetLastError(),DWORD,PASCAL,name('GetLastError')
 end
END
```

This is how to call QueryPerformanceCounter:

```
if not QueryPerformanceCounter(address(C.counternow_li))
 log(GetLastError(),'EnterProc')
end
```

QueryPerformanceCounter sets the LARGE_INTEGER variable appropriately and returns false if a problem was encountered, in which case GetLastError should be called in order to identify the problem. Log() is a vanilla logger which will write messages to a log file. The first argument is the message and the second is some arbitrary text to identify the location in the source code of the particular call to log().

Before you can use QueryPerformanceCounter, however, you have to determine whether it can be used at all. You do this by calling QueryPerformanceFrequency first as a one-off. QueryPerformanceFrequency works similarly to QueryPerformanceCounter except that it returns the frequency (in counts per second) of the high-resolution performance counter, or it returns zero if the installed hardware does not support a high resolution counter. Since a class is already declared, the sensible thing to do is to put the call to QueryPerformanceFrequency in the class's constructor.

```
C          CLASS
freq         group(LARGE_INTEGER).
Construct     PROCEDURE
    end
C.Construct   procedure
 code
 if not QueryPerformanceFrequency(address(c.freq))
  log('* no freq: GetLastError()=' &GetLastError(), 'c.construct')
 else
  c.tickpersec=c.freq.LowUlong
?   assert(c.freq.HighLong=0)
 end
```

The assert will be triggered if no high-resolution performance counter is available, in which case you might as well remove the downloaded file from your project. Alternatively, you can continue your development on a machine which does have a hi-res counter.

Getting the count

A LARGE_INTEGER, being a GROUP, is a little awkward to manipulate so here’s a function to convert it to a 64-bit integer:

```
 map
  ConvPerformanceCounter(LARGE_INTEGER I_li, *decimal O_dec)
 end
type    class
int64     decimal(20)
      end
C              CLASS
Counternow_li     group(LARGE_INTEGER).
counternow_dec    like(type.int64)
bit32             like(type.int64)   !32-bit clarion can't init to 100000000h
                END
C.Construct   procedure
 code
 c.bit32 =10000h
 c.bit32*=10000h   !32-bit clarion can't init to 100000000h
ConvPerformanceCounter    procedure(LARGE_INTEGER I_li, *decimal O_dec)
 code
 if I_li.HighLong=0
  O_dec=I_li.LowUlong
?   assert(O_dec=I_li.LowUlong)
 else
  O_dec =I_li.HighLong
  O_dec*=c.bit32
  O_dec+=I_li.LowUlong

 end
```

ConvPerformanceCounter combines two 32-bit integers into a 64-bit integer. On a 32-bit machine there is no such thing as a 64-bit integer, but you can get the equivalent with a DECIMAL(20). Since on a 32-bit machine you can't directly manipulate genuine integers 33 bits or longer, you need to use a little sleight-of-hand. You can set a 64-bit integer to 100000000h (33 bits) in two steps and use that variable as a multiplier. This achieves the same effect as shifting a 32-bit integer 32 bits to the left, i.e. putting the 32-bit integer into the high part of the 64-bit integer.

You then call ConvPerformanceCounter with the input LARGE_INTEGER as the first argument and the resultant 64-bit integer is returned in the second argument:

```
ConvPerformanceCounter(c.counternow_li, c.counternow_dec)
```

By storing the hi-res counter snapshots in these 64-bit variables you can get the difference very easily by simple subtraction, since the Clarion runtime library uses its decimal arithmetic with DECIMAL variables.

EnterProc and LeaveProc

With these software components in place, the next step is to get on with writing EnterProc and LeaveProc to take the counter snapshots as described earlier. These two procedures are user-defined but they are called by the runtime library and so their prototypes are defined by the vendor as described above.

Getting timings is not as easy as it may seem at first, since almost all procedure/method calls are nested within other calls. You need to keep all these calls in a stack (implemented in this code as a queue) so you can match exits up with entries.

```
Qtype        QUEUE,type      !both stack and array
Line          UNSIGNED
File         STRING(16)
Proc          STRING(64)
         END
stack            queue(qtype)    !push/pop nested procedure calls
counterstart       like(type.int64)
counterstop        like(type.int64)
countsubtotal      like(type.int64)
           end
EnterProc PROCEDURE(unsigned Line,*cstring Proc,*cstring File)
 code
 if not QueryPerformanceCounter(address(C.counternow_li))
  log(GetLastError(), 'EnterProc')
 end
 ConvPerformanceCounter(c.counternow_li, c.counternow_dec)
 !pausetimer(top(stack))
 if records(stack)>0
  get(stack,records(stack))
  stack.counterstop   =c.counternow_dec
  stack.countsubtotal +=(stack.counterstop-stack.counterstart)
  put(stack)
 end
 !push(stack)
 stack.Line         =Line
 stack.File         =File
 stack.Proc         =Proc
 stack.counterstart  =c.counternow_dec
 stack.counterstop   =0
 stack.countsubtotal =0
 ADD(stack)
```

With the messiness being handled at a lower level, EnterProc is actually quite simple. It takes a snapshot of the hi-res counter using QueryPerformanceCounter and pushes this value onto a stack, along with the name of the procedure that was just entered. If there are already items on the stack, the "stopwatch" on the topmost item is paused before the new item is pushed on the top. This way there is an arbitrary number of stopwatches in existence, one for each nested procedure call, but only one stopwatch is "running" at any one time.

```
!associative-indexed access to call counts and execution timings
array            queue(qtype)
Used                LONG
counttotal         like(type.int64)
                end
LeaveProc PROCEDURE()
L_counterstart    like(type.int64),auto
L_counterstop     like(type.int64),auto
L_counttotal      like(type.int64),auto
  code
  if not QueryPerformanceCounter(address(c.counternow_li))
    log(GetLastError(), 'LeaveProc')
  end
  ConvPerformanceCounter(c.counternow_li, c.counternow_dec)
  !pop(stack)
  get(stack,records(stack))
  if errorcode()
    log(errorcode() &'  get(stack,' &records(stack) |
          &')', 'LeaveProc')
  else
    array.line      =stack.line
    array.file      =stack.file
    array.proc      =stack.proc
    L_counterstart  =stack.counterstart
    L_counterstop   =choose(stack.counterstop=0, |
          c.counternow_dec, stack.counterstop)
    L_counttotal    =stack.countsubtotal + |
          (L_counterstop-L_counterstart)
    delete(stack)
  end
  !resumetimer(top(stack))
  if records(stack)>0
    get(stack,records(stack))
    stack.counterstart  =c.counternow_dec
    stack.counterstop   =0
    put(stack)
  end
  !update(array)
  get(array, array.line,array.file)
```

```
    if errorcode()
      array.used        =1
      array.counttotal  =L_counttotal
      add(array, array.line,array.file)

    else
      array.used        +=1
      array.counttotal  +=L_counttotal
      put(array)
    end
```

LeaveProc is no more complicated than EnterProc, really, simply performing the inverse. It performs one additional task, which is to update an array which tracks the amount of time spent inside each procedure.

Finally, the destructor for the local class is called when the program terminates. It reads the array written by LeaveProc and writes a text file with the results of the profiling.

```
C               CLASS
Destruct          PROCEDURE
                END
C.Destruct PROCEDURE
P FILE,DRIVER('ASCII'),CREATE,NAME('profile.txt')
R RECORD
B STRING(256)
  END
  END
I   LONG,AUTO
  CODE



    SORT(array,-array.counttotal)
    CREATE(P)
    ASSERT(~ERRORCODE())
    OPEN(P)
    ASSERT(~ERRORCODE())
    P.B = 'Code Profile  ' &FORMAT(TODAY(),@d17) &' ' |
          &format(clock(),@t7) &'  Counter Frequency=(' |
          &c.freq.HighLong &':' &format(c.freq.LowUlong,@n010) |
          &') counts per second<13,10>'
    ADD(P)
    P.B = right('(clock) count',20) &'  '&right('Invoked',10) |
          &'  '&right('count/Invoked',20) &'  ' |
          &right('seconds',15) &'  '&LEFT('File',12) |
          &'  '&right('Line',5) &'  Procedure'
```

```
ADD(P)
P.B = right('=============',20) &'  '&right('=======',10) |
   &'  '&right('=============',20) &'  ' |
   &right('=======',15) &'  '&LEFT('====',12) |
   &'  '&right('====',5) &'  ========='
ADD(P)
P.B = ''
ADD(P)
LOOP I = 1 TO RECORDS(array)
 GET(array,I)
 P.B = format(array.counttotal,@n-20) &'  '&right(array.Used,10) |
    &'  '&format(array.counttotal/array.used,@n-20.0) |
      &'  '&format(array.counttotal/c.tickpersec,@n-15.6) |
      &'  '&LEFT(array.File,12) &'  '|
      &right(array.Line,5) &'  '&array.Proc
 ADD(P)
END
CLOSE(P)
```

Profiling data

I’ve included some sample profile data; you can view it here. I created this data by running the SCHOOL.APP and choosing various options within the program.

Finding the bottlenecks

From the table written to the file you can quickly establish where the bottlenecks are. First look for total time consumed by a given procedure, then look at how many times it is invoked, then determine whether you wrote it. A procedure that you wrote and consumes significant total time and is called many times is a good candidate for optimization.

When using the profiler remember to:

- Add optprof.clw to your project under External source files
- Turn off profiling for optprof.clw with the project define profile=>off
- Turn on profiling for the app as a whole with the project define profile=>on (or selectively enable profiling as described in the article)

Download the source

Philip Prohm is Director of Optimate Group Pty Ltd, a holding company for numerous ventures including Optimate Computer Systems, Broad Ribbon Designs and The International Camelid Register. The group has customers in five countries. Optimate Computer Systems was established in 1991 in the days of Clarion 2.0 for DOS and these days is Australia's leading vendor of alpaca breeding software, a shrinkwrap application written in Clarion. Away from computers, Philip plays (field) hockey and is partial to science fiction. He and his lovely wife and their two beautiful children are based in Wollongong NSW where they are busily building their global empire.

Reader Comments

Add a comment