# Distributed Computing GCOS 7/DA 7

# DA 7 (NGEN) on TCP/IP for XTA

# Distributed Computing GCOS 7/DA 7

# DA 7 (NGEN) on TCP/IP for XTA

Subject:	This manual describes how to install and configure the product <b>Data Access 7 (NGEN) on TCP/IP for XTA.</b>
Special Instructions:	
Software Supported:	Data Access 7 (NGEN) Version 7 or greater.
	N17GW V1.0.0.1 or greater GCOS 7-V7 TS >= 7560, or GCOS 7-V8 TS >= 8560, or GCOS 7-V9 TS >= 9662
Software/Hardware required:	<ul> <li>PC with Windows 95, 98, or NT 4.0 or NT server. SOCKG7 (XTA versions)</li> <li>SRVCAM</li> </ul>
	<ul> <li>the NTGW7 gateway (release 1.0.0.1 or greater)</li> <li>GCOS 7-V780 or greater</li> <li>V7000</li> </ul>
Date:	June 2001
Bull S.A. CEDOC	Bull HN Information Systems Inc. Publication Order Entry
Atelier de reprographie	FAX: (800) 611-6030

Atelier de reprographie 357, Avenue Patton BP 20845 49008 ANGERS Cedex 01 FRANCE Publication Order Entry FAX: (800) 611-6030 MA30/415 300 Concord Rd. Billerica, MA 01821 U.S.A.

Copyright © Bull S.A., 2001

Bull acknowledges the rights of proprietors of trademarks mentioned herein.

Your suggestions and criticisms concerning the form, contents and presentation of this manual are invited. A form is provided at the end of this manual for this purpose.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical or otherwise without the prior written permission of the publisher.

Bull disclaims the implied warranties of merchantability and fitness for a particular purpose and makes no express warranties except as may be stated in its written agreement with and for its customer. In no event is Bull liable to anyone for any indirect, special, or consequential damages. The information and specifications in this document are subject to change without notice.

Consult your Bull Marketing Representative for product or service availability.

# Preface

Scope and Objectives	This manua <b>TCP/IP for</b> to refer to t	al describes how to install and configure the <b>Data Access 7 (NGEN) on</b> r <b>XTA</b> product. In this manual, the abbreviation <b>DA 7 (NGEN)</b> is used his product. NGEN stands for "New Generation".
Intended Readers	This manua DA 7 (NGI	al is intended for persons who need to install, configure, and/or use the EN) product.
Prerequisites	None.	
Structure	This manua	al contains 6 chapters:
	Chapter 1	gives an overview of DA 7 (NGEN).
	Chapter 2	describes the installation of DA 7 (NGEN).
	Chapter 3	describes the functions and architecture of DA 7 (NGEN).
	Chapter 4	describes the configuration of Bull MultiDBMS driver.
	Chapter 5	describes the configuration of DA 7 (NGEN).
	Chapter 6	describes the migration of DA7NGEN-based applications from ODBC to OLE DB.
Bibliography	GCOS 7 D. GCOS 7 D.	DA Manager SQL Supplement47 A2 52UR DA Manager Administrator's Guide47 A2 51UR
Syntax Notation	The abbrev Data Acces	iation <b>DA 7</b> ( <b>NGEN</b> ) is used for the name of the product ss 7 ( <b>NGEN</b> ) on <b>TCP/IP for XTA</b> .

No



Delivery Conditions

# **Table of Contents**

# 1. Data Access 7 (NGEN) Overview

	1.1	Purpose	1-1
	1.2	Hardware and Software Requirements	1-2
		1.2.1 Hardware Requirements on PC	1-2
		1.2.2 Software Requirements	1-2
		1.2.2.1 PC Software	
		1.2.2.2 Maintrame Soπware	1-2
	1.3	Benefits of Data Access 7	1-3
2.	Data	a Access 7 (NGEN) Installation	
	2.1	Prerequisites for Installation	2-1
	2.2	The Installation Program	2-2
	2.3	Installation Procedure	2-3
3.	Data	a Access 7 (NGEN) Functions & Architecture	
3.	<b>Data</b> 3.1	a Access 7 (NGEN) Functions & Architecture Functional Description	3-1
3.	<b>Data</b> 3.1 3.2	a Access 7 (NGEN) Functions & Architecture Functional Description Data Access 7 Architecture in two-tier context	3-1
3.	<b>Data</b> 3.1 3.2 3.3	a Access 7 (NGEN) Functions & Architecture Functional Description Data Access 7 Architecture in two-tier context Data Access 7 Architecture in three-tier context	3-1 3-2 3-4
3. 4.	Data 3.1 3.2 3.3 Bull	a Access 7 (NGEN) Functions & Architecture Functional Description Data Access 7 Architecture in two-tier context Data Access 7 Architecture in three-tier context I MultiDBMS ODBC Driver	3-1 3-2 3-4
3.	<b>Data</b> 3.1 3.2 3.3 <b>Bull</b> 4.1	a Access 7 (NGEN) Functions & Architecture Functional Description Data Access 7 Architecture in two-tier context Data Access 7 Architecture in three-tier context I MultiDBMS ODBC Driver Prerequisites for Installation	3-1 3-2 3-4 4-1
3.	Data 3.1 3.2 3.3 Bull 4.1 4.2	a Access 7 (NGEN) Functions & Architecture Functional Description Data Access 7 Architecture in two-tier context Data Access 7 Architecture in three-tier context I MultiDBMS ODBC Driver Prerequisites for Installation Setting up a Data Source	3-1 3-2 3-4 4-1 4-2
3.	Data 3.1 3.2 3.3 Bull 4.1 4.2 4.3	Access 7 (NGEN) Functions & Architecture Functional Description Data Access 7 Architecture in two-tier context Data Access 7 Architecture in three-tier context I MultiDBMS ODBC Driver Prerequisites for Installation Setting up a Data Source Modifying an existing Data Source	3-1 3-2 3-4 4-1 4-2 4-5
3.	Data 3.1 3.2 3.3 Bull 4.1 4.2 4.3 4.4	Access 7 (NGEN) Functions & Architecture Functional Description Data Access 7 Architecture in two-tier context Data Access 7 Architecture in three-tier context I MultiDBMS ODBC Driver Prerequisites for Installation Setting up a Data Source Modifying an existing Data Source Deleting a Data Source	3-1 3-2 3-4 4-1 4-2 4-5 4-7



# 5. Data Access 7 (NGEN) Configuration

5.1	Config	uring TCP/IP	5-1
5.2	Config	uring the Initialization File	5-2
5.3	Config	uring the DSN Connection Parameters	5-5
5.4	Tests,	Trace Files, and Return Codes	5-8
	5.4.1	Connection Error and Return Codes	5-8
	5.4.2	Activation of Traces	5-9
5.5	Securi	ty	5-12

# 6. Migrating DA7NGEN-based applications from ODBC to OLE DB

6.1	What i	is OLE DB?	6-1
	6.1.1	OLE DB relies on COM	6-1
	6.1.2	Consumers and Providers	6-1
	6.1.3	COM objects defined in OLE DB	6-2
	6.1.4	The ODBC Provider	6-4
	6.1.5	OLE DB AND ADO	6-4
	6.1.6	ADO Object models	6-5
		6.1.6.1 ADO: Data manipulation	6-5
		6.1.6.2 ADOX: Data definition and security	6-6
		6.1.6.3 JRO: Replication	6-7
6.2	Migrat	ion guidelines	6-7
	6.2.1	Adding references to ADO / ADOX type libraries	6-7
	6.2.2	Opening a Database	6-8
	6.2.3	Retrieving Data	6-10
		6.2.3.1 Opening a Recordset	6-10
		6.2.3.2 Navigating within a Recordset	6-13
		6.2.3.3 ADO shortcuts to display and open Recordsets	6-15
		6.2.3.4 Using ActiveX controls	6-16
		6.2.3.5 Error handling (in Visual Basic)	6-16

Index



# **Table of Graphics**

## Figures

Figure 3-1	Data Access 7 Architecture in two-tier context	3-2
Figure 3-2	Data Access 7 Architecture in three-tier context	3-4





# 1. Data Access 7 (NGEN) Overview

#### 1.1 Purpose

Distributed Computing Environments are being deployed by enterprises more and more frequently. The role of Bull DPS 7000/GCOS 7 platforms is reinforced as Enterprise Servers provide the mainframe disciplines for hosting corporate databases and the server parts of business-critical applications, based on various Client/Server technologies.

**Data Access 7 (NGEN)** is part of Bull's offer aimed at implementing PC-based Decision Support Systems through a seamless integration of ODBC-compliant PC tools with GCOS 7 UFAS & IDS/II databases, based on either a two-tier PC->V7000 architecture or a three-tier PC->NT server->V 7000 architecture.

For DA 7 (NGEN), this document focuses on the installation, administration, and programming rules of the PC client side.



## **1.2 Hardware and Software Requirements**

The following paragraphs describe the hardware and software required by DA 7 (NGEN).

#### 1.2.1 Hardware Requirements on PC

DA 7 (NGEN) requires the following hardware:

- PC supporting Windows 95 or Windows 98 or Windows NT.
- DA 7 (NGEN) components use about 400 Kbytes of memory. However, to use Data Access 7 (NGEN), several other software components are required, such as MS-Windows, the Bull MultiDBMS ODBC driver, the ODBC Driver Manager and an ODBC compliant application. Because all of these components may potentially be required to use DA 7 (NGEN), a minimum of 16 Mbytes is recommended.
- A hard disk drive and approximately 2 Mbytes of hard disk space for the software described above. Efficient execution requires 4 Mbytes of free space.
- A 3-1/2 inch floppy drive for installation purposes. Note that this is not required if installation is effected by downloading.
- An Ethernet Card.

#### 1.2.2 Software Requirements

#### 1.2.2.1 PC Software

DA 7 (NGEN) requires the following PC software:

- Microsoft Windows NT 4.0 or Windows 95 or greater.
- Bull MultiDBMS ODBC Driver release 02.10.30 or greater (for ODBC compliant tools).
- DA 7 (NGEN) PC Gateway, MkgIni, DsnCnfg, and PcClean (included on the installation diskette or available by downloading).

#### 1.2.2.2 Mainframe Software

DA 7 (NGEN) requires the following mainframe software:

- GCOS 7-V780 or greater
- V7000
- SRVCAM
- SOCKG7 (Diane versions)
- the NT7GW gateway (release 1.2.0 or greater)
- In addition, DA 7 requires GCOS 7-DDA Manager V32.01or greater.



## 1.3 Benefits of Data Access 7

Data Access 7 (DA 7) provides easy and ad-hoc access to enterprise information. It is a powerful means for enterprise players to improve their decision processes and carry out their day-to-day business activities with increased productivity.

The openness of ODBC enables the DA 7 solution to meet your needs and retains the benefits of your existing investments:

- PC's, MS-Windows software products and related end-user skills,
- GCOS 7 UFAS & IDS/II databases (no need to change either data structures or existing applications),
- Bull DPS 7000 power capability for processing of large databases,
- Simple two-tier architecture facilitating implementation and deployment.

The capability for end-users to manage their information needs by themselves frees the MIS staff from the continuous task of building customized user queries.

Security and integrity of data is guaranteed through centralized management and standard database security features.

DA 7 enables the integration of individual productivity tools with the Enterprise Client/Server applications on the end-user's desktop within a standard Windows environment.





# 2. Data Access 7 (NGEN) Installation



#### WARNING:

Some screen images may depend on the installation machine operating system.

The version numbers seen on the screen images may vary for future installations.

#### 2.1 Prerequisites for Installation

The DA 7 (NGEN) Gateway diskette contains the key components for the DA 7 solutions.

Other software requirements for its use are as follows:

- V7000 must be installed on the XTA machine and GCOS7 must be executing
- SRVCAM must be executing under GCOS7
- NT7GW and SOCKG7 must be executing on the XTA machine
- The DDW/ODBC2 driver must be installed on the PC or NT server.
- On the DPS 7000, the GCOS 7 DDA Manager must be installed and set up to access a GCOS 7 database before the Data Access 7 application on the PC or NT server can be used.



## 2.2 The Installation Program

To install DA 7 (NGEN) on the PC, you use the installation program SETUP.EXE as described in the paragraph "Installation Procedure" below.

The installation program copies the following:

- 1. for BullMultiDBMS driver:
  - BULL Dynamic Link Library interface application (BULLODBC.DLL).
  - BULL Dynamic Link Library (BULLSTP.DLL).
  - BULL Help file (BULLODBC.HLP).
- 2. for DA7NGEN driver:
  - DA 7 GW Dynamic Link Library (NDDAOP7.DLL).
  - DA 7 interface with Bull MultiDBMS ODBC Driver (ITFDA.DLL).
  - Error message file (GWSQL7ER.ENG).
  - Tools for connection configuration (DSNCNFG.EXE).
  - Tools for configuraton of .ini file Mkg Ini (ESPINI.EXE) which creates the parameter file (OP7GTW.INI).
  - Clean-connection program (PCCLEAN.EXE).

The installation program lets you specify a directory other than the directory **DA7NGEN**. If you choose a different directory, that directory must be declared in your path environment.



## 2.3 Installation Procedure

- 1. Start Windows.
- If you have a previous version of DA 7, you must first uninstall it.
   DA7NGEN can be installed from a PC or NT server either using a browser.
- 3. To install by downloading, first carry out the following steps:





In the page that appears, click the "DA7NGEN Info" link:



Note that other products may be added to this page from time to time.



The following page allows for downloading of the DA7NGEN product including DA7 driver and Bull MultiDBMS ODBC driver. To select the product to be downloaded, simply click on the appropriate link.







91% of da7ngen.e	exe Completed	_ 🗆 🗵
<b>80</b> 0		
Opening:		
da7ngen.exe from l	rcls3201	
Estimated time left: Download to: Transfer rate:	23 sec (92,0 KB of 1,26 MB copied) Temporary Folder 52,0 KB/Sec	
🔽 Close this dialo	g box when download completes	
	Open Open Folder	Cancel





A self-extracting executable file (named DA7NGEN\_ODBC.exe) is downloaded to the PC or NT server. Once executed (either by the browser, or by the user if he chose to save it on his PC or NT server), the installation begins.

4. Run the installation program (install/remove software) or run the Setup.exe.

The "Welcome" window appears, as shown below.





5. Advance in the installation process by clicking on "Next". The "User Information" window then appears, as shown below.

User Information	X
	Please enter your name and the name of the company for whom you work.
	Ngme: Jacobs Company: BULL S.A
	<∄ack Nest> Cancel

6. Click on "Next" in this window to obtain the "Choose Destination Location" window.



<b>Choose Destination Loca</b>	ition E	đ
	Setup will install DA7NGEN in the following directory.	
	To install to this directory, click Next.	
	To install to a different deectory, click Browse and select another directory.	
	You can choose not to install DA7NGEN by clicking Cancel to exit Setup.	
- 🍕	Destination Directory D:\DA7NGEN	
	< <u>B</u> ack Next> Cancel	

7. Choose the destination directory and click on "Next". The following window appears, requesting the setup type. Choose Custom installation.



8. Click on "Next" to obtain the "Select Components" window.



Select Components		×
	Select the components you want to install, you do not want to install.	clear the components
	Companents	
	Program Files	1001 K
	Description	
<b>1</b>		Conge
	Space Required:	1001 K
	Space Available:	855780 K
	< <u>B</u> ack <u>N</u> ext>	Cancel

9. Click on "Next" to obtain the "Select Program Folder" window.

Select Program Folder	×
	Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing Folders list. Click Next to continue.
	Program Folders:
And the second second	Existing Folders:
20	Adobe Acrobat 4.0 ATI Bull MuhDBMS 0DBC
	Démanage ESP7NGEN InstallShield 5.5 Professional Edition
	Microsoft Developer Network.
	<back next=""> Cancel</back>

- 10. Select DA7NGEN and click on "Next".
- 11. When the following window appears, choose your installation type (PC or NT server version)





12. When the following window appears, choose your configuration parameters (see Chapter 4).

PATH	D-DA7NDEN	GUSER IN	YES
DEBUG	NODERUG	THEOUTINI	20
LEVEL INI	0	TIMEOUTZINI	600
DATA M/F (N)	00	VERSION (N)	1
URVPT (1)	YES	HUFFER (V)	40285
*1 YES a N	a		
N   Names			

13. Click on "OK" to continue the installation process. The "Setup Complete" window appears. Click on "Finish" to complete the installation process.





# 3. Data Access 7 (NGEN) Functions & Architecture

#### 3.1 Functional Description

On the end-user's PC side, the Bull MultiDBMS ODBC Level 2 driver and DA 7 PC Gateway components enable ODBC-interface PC tools to access GCOS 7 UFAS & IDS/II data in read-only OPEN SQL mode.

With an SQL query, the PC tool can access a relational model that represents one IDS/II database or a set of UFAS files.

A wide range of PC tools is supported such as Access, Excel, Visual Basic, Andyne's GQL, LOTUS 1-2-3, Vision Pro, Windows 4GL, Gupta's Quest, SQL Windows, Powerbuilder, Cognos, Impromptu, PC/FOCUS for Windows, and so on.

On the V 7000 side, the Database Administrator must have previously defined the relational models (according to the end-users' needs).

The GCOS 7 DDA Manager component processes read-only SQL queries on non-relational UFAS & IDS/II databases, using the defined relational models.

The GCOS 7 DDA Manager can share access to data with other user applications.



## 3.2 Data Access 7 Architecture in two-tier context

The following figure shows how Data Access 7 enables PC end-users to access data on a V7000 in a two-tier context.



Figure 3-1 Data Access 7 Architecture in two-tier context



On the PC Client side:

- The Bull MultiDBMS ODBC Level 2 driver provides the Application Programming Interface (API) for the PC tool or Client application to issue SQL requests towards the connected "ODBC source".
- The DA 7 PC Gateway (DA7NGEN) dialogs with the NT7GW Gateway on the NT side, which is in charge of converting data sent to or received from DA7NGEN.

On the Bull DPS 7000 Server side:

- The NTGW 7 Gateway dialogs with the DA 7 Gateway on the PC side, and is in charge of forwarding SQL requests towards the GCOS 7 DDA Manager as well as returning data on completion of request execution.
- The GCOS 7 DDA Manager is a Bull-defined dedicated TDS application in charge of handling the SQL requests issued from several users and having them executed concurrently.

The PC-Bull DPS 7000 link is based on TCP/IP, with standard Windows TCP/IP sockets on the PC side and standard OPEN 7 TCP/IP sockets on the Bull V 7000 side.Note that there are two ways of communicating with SQL7 via NT7GW:

- 1. either using standard SockG7 and the VCAM server to effect DSA connections to SQL7
- 2. or using the TDS-TCP/IP specific version of SockG7 to effect TCP/IP connections to SQL7 (in this case SQL7 must be regenerated for use with TCP/IP).



## 3.3 Data Access 7 Architecture in three-tier context

The following figure shows how Data Access 7 enables PC end-users to access data on a V7000 in a three-tier context.



Figure 3-2 Data Access 7 Architecture in three-tier context





#### WARNING:

Some screen images may depend on the installation machine operating system.

The Bull MultiDBMS ODBC Driver is required for use with DA7NGEN. This chapter describes how to instal and configure this product.

## 4.1 Prerequisites for Installation

The pre-requisites for Bull MultiDBMS ODBC Driver installation are outlined below:

- An MS-Windows environment must be operational.
- Local Area Network (LAN) software must be installed and operational.
- ODBC compliant tool(s) must be installed and tested locally.
- Networking for Windows must include a socket interface to the LAN.



## 4.2 Setting up a Data Source

Before you can access data with the Bull MultiDBMS ODBC Driver, you must set up a data source for each of the target databases. The Bull MultiDBMS ODBC Driver uses the information you enter when you add the data source in order to access the data. Note that you can modify or delete a data source at any time (see 4.3 Modifying an existing Data Source and 4.4 Deleting a Data Source).

The Bull MultiDBMS ODBC Driver Data Source facilities are accessed via "Start" + "Programs" + "DA7NGEN" + "ODBC Data Source Administrator".

As a result of this action, the following window appears.

🐖 ODBC D	ata Source Administrator 🔗	×
User DSN	System DSN File DSN Drivers Tracing Connection Pooling About	
System D	Jala Sources: Agd	I
Name	Driver	L
b4 1s	Bull MultiDBMS Driver Eemove	L
b4 2s	Bull MultiDBMS Driver	E.
b4_9s	Bull MultiDBMS Driver Configure	L
b6_1s	Bull MultiDBMS Driver	L
b6_2s	Bull MultiDBMS Driver	E.
b6_6s	Bull MultiDBMS Driver	L
b6_bd	Bull MultiDBMS Driver	L
b6_bm	Bull MultiDBMS Driver	L
b6_bp	Bull MultiDBMS Driver	L
b6_bs	Bull MultiDBMS Driver	L
b6 bu	Bull MultiDBMS Driver	E.
3	An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.	
	OK Annuler Appfouer Aide	

This window may be used to add, modify or remove data source entries.

It is highly recommended to choose the option "System DSN" when adding data source entries, as only system DSN entries allow for tracing .

After choosing "System DSN", we choose "Add" to create a new data source entry.

The "Create New Data Source" window appears, as shown below.

We select "Bull MultiDBMS Driver" and then click on "Terminate".



At this point, the "Bull MultiDBMS ODBC Driver Setup" window appears (see below). We may now provide all the required information concerning the database to be accessed.

Create New Data Source			? ×
	Select a driver for which you want i Name Buil MultiDBMS Driver Microsoft Access Driver (*.mdb) Microsoft Bacel Driver (*.dbf) Microsoft Excel Driver (*.dbf) Microsoft FoxPoo Driver (*.dbf) Microsoft Parados Driver (*.dbf) Microsoft Text Driver (*.bst: *.csv) SQL Server	v set up a data Version 2.10.26.00 3.51.1713.00 3.51.1713.00 3.51.1713.00 3.51.1713.00 3.51.1713.00 3.51.1713.00 3.51.1713.00 3.70.06.23	Bounce. Bull HM Micros Micros Micros Micros Micros Micros Micros Micros
		miner /	• ernuler

Bull MultiDBMS OD	BC Driver Setup	×		
Data Source <u>N</u> ame	nyTest	ОК		
Description	DDW Server	Cancel		
Server		<u>H</u> elp		
⊻irtual Node	myTest			
<u>D</u> atabase	modtest			
<u>S</u> erver Type	DDW/SQL7	Options>>		
Options	DDW/RMS			
Use Select Loop				
Use <u>T</u> able for SC	L Statistics			
Se with Access	or Visual Basic			
Show System Ta	ibles			
Initial Autocommit State				
Name of Logon INI F	file (for			



Data Source Name	a unique name that identifies the parameters used to access the target database
Description	a description of the data in the data source
Virtual Node	the name used by DA7NGEN to obtain the connection parameters
Database	the name of the database
Server Type	the server type (DDW/SQL7 must be chosen)
Use Select Loop	Tuple data is ordinarily returned from a database server using a cursor fetch loop, one row per fetch. This can generate a high level of network traffic. The use of Select Loop allows an optimization, whereby multiple tuples are batched in each server data message. The number of tuples returned depends on the size of the tuple.
Use Table for SQL Statist	ics Many ODBC compliant applications (for example, MS Access, Visual Basic, PowerBuilder) use SQLStatistics ODBC API calls to provide index and table volume information. SQLStatistics may be optimized by selecting this option. A table is created and maintained to hold the SQLStatistics information. This table must be refreshed periodically.
Use with Access or Visual	<b>Basic</b> This option must be selected for use from an. Access or Visual Basic application.
Show System Tables	This option must be selected if system tables are to be displayed
Initial Autocommit State	These buttons are used to alter the initial Auto Commit state of the ODBC connection. The default state is ON and should remain ON.
Name of the LOGON INI	file name of .ini file for LOGON

The following options are displayed in the window:

Click on "OK" in the "Bull MultiDBMS ODBC Driver Setup" window and then on "OK" in the "ODBC Data Source Administrator" window to complete the data source creation.





## 4.3 Modifying an existing Data Source

The Bull MultiDBMS ODBC Driver Data Source facilities are accessed via "Start" + "Programs" + "DA7NGEN" + "ODBC Data Source Administrator".

As a result of this action, the following window appears.

🕼 ODBC D	ata Source Administrator	? ×		
User DSN	System DSN File DSN Drivers Tracing Co	nnection Pooling About		
<u>S</u> ystem D	lata Sources:	Add		
Name	Driver			
b4 1s	Bull MultiDBMS Driver	Bemove		
b4 2s	Bull MultiDBMS Driver			
b4 9s	Bull MultiDBMS Driver	Configure		
b6 1s	Bull MultiDBMS Driver			
b6_2s	Bull MultiDBMS Driver			
b6_6s	Bull MultiDBMS Driver			
b6_bd	Bull MultiDBMS Driver			
b6_bm	Bull MultiDBMS Driver			
b6_bp	Bull MultiDBMS Driver			
b6_bs	Bull MultiDBMS Driver			
b6 bu	Bull MultiDBMS Driver	<u> </u>		
An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.				
	OK. Annuler	Aide Aide		

This window may be used to add, modify or remove data source entries.

It is highly recommended to choose the option "System DSN" when adding data source entries, as only system DSN entries allow for tracing.

After choosing "System DSN", we choose "Configure" to modify an existing data source entry.

At this point, the "Bull MultiDBMS ODBC Driver Setup" window appears (see below). We may now modify all the required information concerning the database to be accessed.

Click on "OK" in the "Bull MultiDBMS ODBC Driver Setup" window and then on "OK" in the "ODBC Data Source Administrator" window to complete the data source modification.



Bull MultiDBMS (	DBC Driver Setup	×
Data Source <u>N</u> ame	my Test	OK.
Description	DDW Server	Cancel
Server		Help
⊻irtual Node	myTest	
Database	modtest	-
Server Type	DDW/SQL7	
Options Use Select La Use Lable for Use with Acc Show System Initial Autocome	DDW/RMS DDW/SSUL7 DDW/SYBASE DDW/SYBASE DDW/VSAM SQL Statistics ess or Visual Basic Tables nk State	×
C Dn Dff Name of Logon II	NI File (for	

The above fields are described in Section 2.4 Creating a Data Source Entry.



## 4.4 Deleting a Data Source

The Bull MultiDBMS ODBC Driver Data Source facilities are accessed via "Start" + "Programs" + "DA7NGEN" + "ODBC Data Source Administrator".

As a result of this action, the following window appears.

🐠 ODBC D	ata Source Administrator		? ×	
User DSN	System DSN File DSN Drivers Tracing	Connection Poo	aling About	
<u>S</u> ystem D	ata Sources:		Add.	
Name	Driver	-		
b4_1s	Bull MultiDBMS Driver		Bemove	
b4_2s	Bull MultiDBMS Driver			
b4_9s	Bull MultiDBMS Driver		Configure	
b6_1s	Bull MultiDBMS Driver			
b6_2s	Bull MultiDBMS Driver			
b6_6s	Bull MultiDBMS Driver			
b6_bd	Bull MultiDBMS Driver			
b6_bm	Bull MultiDBMS Driver			
b6_bp	Bull MultiDBMS Driver			
b6_bs	Bull MultiDBMS Driver	-1		
1 b6 bu	Bull MultiDBMS Driver	-		
An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.				
	OK. Annuler	Apptiquer	Aide	

This window may be used to add, modify or remove data source entries.

It is highly recommended to choose the option "System DSN" when adding data source entries, as only system DSN entries allow for tracing.

After choosing "System DSN", we select the data source to be removed (b4\_1s in our example) and click on "Remove" to delete the data source entry. We must provide confirmation by clicking on "OUI" in the following window.

ODBC Ad	dministrator 🛛 🕅
?	Are you sure you want to remove the testpw data source?
	Qui



# 4.5 Data Types

The Bull MultiDBMS ODBC driver maps OPEN SQL data types to ODBC SQL data types. The equivalents are shown in the following table.

OPEN SQL data type	ODBC SQL data type	Description
CHAR	SQL_CHAR	Character string of fixed length n (1 <=n<= 254)
FLOAT	SQL_FLOAT	Signed, approximate, numeric vale with a mantissa precision of 15 (0 or absolute value 10 to the power -308 to 10 to the power 308)
INTEGER	SQL_INTEGER	Exact numeric vale with precision 10 and scale 0 (signed: -2 to the power $31 \le n \le 2$ to the power 31 - 1, Unsigned: $0 \le n \le 2$ to the power $32 - 1$ )
SMALLINT	SQL_SMALINT	Exact numeric vale with precision 10 and scale 0 (signed: -32768 <=n<=32767, Unsigned: 0<=n<=65535)
VARCHAR	SQL_VARCHAR	Variable length character string with a maximum length of n, where 1 <= n <= 254

# 5. Data Access 7 (NGEN) Configuration



#### WARNING:

Some screen images may depend on the installation machine operating system.

Configuring Data Access consists of the following steps:

- Configuring TCP/IP.
- Configuring the Initialization file.
- Configuring ODBC.
- Configuring the DSN Connection Parameters.

Each of these steps is described below, except for ODBC configuration, which is described in Chapter 2.

#### 5.1 Configuring TCP/IP

Edit the "services" file in the <System Disk>:\<Windows Directory> directory by adding lines of the following form:

- Srvcam 9003/TCP
- Srvntgw 9002/TCP

Edit the "hosts" file in the <System Disk>:\<Windows Directory> directory and add (if not already included) the address of your PC and the host (NTaddress).

#### **EXAMPLE:**

129.182.197.25 FRCLS3201 DPS7 #address of NT machine

where:

FRCLS3201 is the NT machine name DPS7 is the alias (4 characters maximum) for the NT machine name)



# 5.2 Configuring the Initialization File

The initialization file name is OP7GTW.INI located in the directory:

<SystemDisk>:\<Windows directory >

#### NOTE:

This tool is used by other products.

1. Click on the MkgIni icon.



2. Choose the Product type DA7NGEN.

Prod	uct type		×
			1
	DA7NGEN DA7NGEN		
	ESP7NGEN NATOP7		
	NATOP7		
	1		
	Cancel	ОК	

- 3. As a result, the "Make Ini File" window appears, as shown below.
- 4. Enter/modify the parameters.

#### Section: [DA7NGEN]

Field "PATH"	Entry: PATH=
C:\DA7NGEN	(default). Set during the product installation.
Field "DEBUG"	Entry: DEBUG=
NODEBUG	(default)



FILE\_ALL Internal trace for support team. Records the internal procedures in the PC Gateway in the "g7a\_xx.log" file where xx is the connection number.

Mak	e Ini File					
	PATH	D:\DA7NGEN	GU	SER (*)	YES	
	DEBUG	NODEBUG	TIN	IEOUT1 (N)	20	
	LEVEL (N)	0	TIN	IEOUT2 (N)	600	
	DATA M/F (N)	00	VE	RSION (N)	1	
	CRYPT (*)	YES	BU	FFER (V)	4096	
	(*) YES or N	1				
	( )					
	(N) Numeric					
	(V) 1024 or 20	148 or 4096 or 8192				
		Cancel	[	OK		

Field "LEVEL (N)"	Entry: LEVEL=
0:	(default). No trace between ODBC and DA 7 gateway.
1:	Trace in c:\TrcDA_xx.log
Field "DATA M/F (N)"	Entry: DATA=
00:	(default) returned data will be stored in temporary memory. Note that if the returned data $> 32K$ , you will obtain a memory violation message.
10:	returned data will be stored in a temporary file (data > 32K).



Field "CRYPT (*)"	Entry: CRYPT=			
YES:	(default) password is crypted.			
NO:	The password is not crypted.			
Field "GUSER (*)"	Entry: GUSER=			
YES:	(default). Authorize dynamic input of the user name and password for a connection if the password = <b>DA7ESP7UR</b> .			
NO:	This functionality is not authorized.			
Field "TIMEOUT1 (N)"	Entry: TIMEOUT1=			
20:	(default). This is the timeout value applied for connection (in seconds).			
nnn:	up to 3 digits.			
Field "TIMEOUT2 (N)"	Entry: TIMEOUT2=			
600:	(default). This is the timeout value applied for reception of data (in seconds).			
nnn:	up to 3 digit			
Field "VERSION (N)"	Entry: VERSION= 1			
1:	(default).			
n:	A single digit. <b>Do not modify this field.</b> It is reserved for communication with NT7GW).			
Field "BUFFER (V)"	Entry: BUFFER=			
4096:	(default)			
2048 or 8192	depending on the expected data row size. If any other value is supplied, the default value (4096) is used.			
5 Validata your modifica	tions by alighting on the bytton "OV"			

5. Validate your modifications by clicking on the button "OK".



#### **Configuring the DSN Connection Parameters** 5.3

DSN stands for Data Source Name (Virtual Node for ODBC).

Note that this tool is used by other products.

1. Click on the DSNCnfg icon.



2. Choose Product type DA7NGEN and click on OK.

Prod	luct type	×
	DA7NGEN	
	ESP7NGEN NATOP7	
	1	
	Cancel	OK

3. Create a new Virtual Node by entering its name and clicking on New.

Virtual Node	×
b6_1s	New
b6_1s b6_2s espf	Modify
	Delete
	View
Cancel	OK



4. Enter the parameter values.

Virtual Node		×
Virtual Node	b6_1s	
UserName	tsqlhst1	
Password	×××	
Password confirm	×××	
Host name	BC06-42	
Port num	9002	
GCOS	BC06	
Mailbox	TSQL	
Project		
Billing		
Terminal ID		
Trace		Cancel
Other		(OK)

#### NOTE THAT THE ABOVE SCREEN IMAGE WILL BE MODIFIED LATER

User name:	name used for the TDS connection.
Password:	Password used for the TDS connection.
Host name:	name of the NT.
Port num:	Port used by the NT7GW gateway (see the documentation of the NTGW 7 gateway). This should correspond to the services file entry for srvntgw.
GCOS:	GCOS name.
Mail box:	Name of the Mail box where the TDS is connected.
Project name (optional):	requested by your GCOS 7 administrator.



Billing name (optional):	requested by your GCOS 7 administrator.
Terminal ID:	(not used by DA 7).
Trace:	YES or NO.
	THE TRACE FILE NAME IS NOT YET DEFINED
Other:	(reserved for other applications).

5. Validate this grid (return to the previous grid) by clicking on "OK".

Validate the virtual node grid (to record all the modifications) by clicking on "OK", or cancel (to disregard all modifications) b clicking on "Cancel".



#### 5.4 Tests, Trace Files, and Return Codes

#### 5.4.1 Connection Error and Return Codes

The connection codes are returned in the file  $c: \G7A.LOG$ 

THE FOLLOWING LIST MAY HAVE TO BE MODIFIED - 201 = "Gateway not initialized"; - 202 = "Memory allocation error"; - 203 = "Too many users"; - 301 = "Gateway not initialized"; - 302 = "String connection too long"; - 303 = "Invalid DSN"; see ODBC Administrator or DSN configurator - 304 = "Invalid server type"; see ODBC Administrator (SQL7) - 305 = "Memory allocation error"; - 306 = "Connect cancel by user"; - 400 = "Comm error: Other error"; - 401 = "Comm error: Invalid port number"; see DSN configurator - 402 = "Comm error: Invalid host name"; see DSN configurator - 403 = "Comm error: Socket creation error"; - 404 = "Comm error: Socket connection error"; Op7Gateway not started or different port number - 405 = "Comm error: Send/Receive TCP/IP error"; - 406 = "Comm error: Timeout connect expired"; - 407 = "Comm error: Too many connections"; - 408 = "Comm error: Bad session ID or too many users"; - 409 = "Comm error: Receive TCP/IP error"; - 410 = "Bad terminate last transaction ; Retry "; - 500 = "Server error: Other error"; - 501 = "Server error: No fork process"; - 503 = "Server error: Bad string connection"; - 504 = "Server error: GetDSA failed"; - 506 = "Server error: All connections used"; - 507 = "Server error: MI not avaible"; - 508 = "Server error: Bad password"; - 509 = "Server error: Select error"; - 510 = "Server error: Administrator already connected"; - 511 = "Server error: Open DSA failed"; - 512 = "Server error: gethostbyaddr error"; - 513 = "Server error: No fork process"; - 600 = "Server error: error read PC"; - 700 = "Server error: error read MB"; - 800 = "Server error: Internal error with VCAM"; - 900 = "Server error: Internal error";



-	10500	=	"VCAM	error:	Other error";
-	10501	=	"VCAM	error:	Abnormal end";
-	10504	=	"VCAM	error:	Unknown mail box";
-	10524	=	"VCAM	error:	User/Password bad access right";
-	10566	=	"VCAM	error:	User already connected";
(2	cxxxx) =	= E	lrror r	eturneo	d by SQL7 TDS server ;

#### NOTE:

When a fatal error is detected, this is recorded in the file  $C:\G7A.log$ 

#### 5.4.2 Activation of Traces

The different trace files will be analyzed by the technical support when requested.

Note that the activation of traces significantly decreases performance.

Traces can be activated at different levels as follows:

#### Between Your Application and the ODBC API

1. In the ODBC administrator tool, select the "Tracing" page.

💞 ODBC Data Source Administrator	? ×
User DSN System DSN File DSN Dr	ivers Tracing Connection Pooling About
When to trace	
Start <u>Tracing Now</u>	Start <u>V</u> isual Studio Analyzer
_Log file Path	
D:\SQL.LOG	D:\WINNT\System32\odbctrac.dll
Browse	<u>S</u> elect DLL
ODBC tracing allows you to o use by support personnel or t Visual studio tracing enables	create logs of the calls to ODBC drivers for o aid you in debugging your applications. Microsoft Visual studio tracing for ODBC.
ОК	Annuler Aide



- 2. In the "when to trace" field, choose trace mode (by clicking on the field "Start tracing now").
- 3. In "log file path", choose the file name (default SQL.LOG).

Remember to stop tracing later (by clicking on the field "Stop tracing now").

#### Inside the ODBC Multidbms Driver

- 1. Run "regedit.exe".
- 2. Select HKEY\_LOCAL\_MACHINE\Software\ODBC\ODBC.INI
- 3. For the DSN to trace: set the new value "UseLog" and set the string to "True".
- 4. The trace file is BullTr.log located in: <System Disk>:\<Windows Directory>



#### Between the ODBC Multidbms Driver and DA 7 Gateway

- 1. Click on the MkgIni icon.
- 2. Choose the product DA7NGEN.
- 3. Modify the field "Level " by setting its value = 1.
- 4. The trace file is C:\TrcDa\_xx.log, where xx is the session number.



#### Inside the DA 7 Gateway

- 1. Click on the MkgIni icon.
- 2. Choose the product DA7NGEN.
- 3. Modify the field "Debug " by setting its value = "FILE\_ALL".
- 4. The trace file is  $C: G7A_xx.log$ , where xx is the session number.

#### Inside the NT7GWGateway

- 1. Click on the DSN Cnfg icon.
- 2. Choose the product DA7NGEN.
- 3. Enter YES in the Trace field.



# 5.5 Security

An encrypted password facility is available.

Dynamic User/Password is available if the password is DA7ESP7UR. The gateway opens a dialog box asking you to enter the new User and the new password.

This does not activate SECUR'ACCESS for the TDS SQL7 server.



# 6. Migrating DA7NGEN-based applications from ODBC to OLE DB

This chapter is a guide to changing code that uses the ODBC API implemented by DA7NGEN to a code that uses the OLE DB API, through the Microsoft ODBC provider. As it is not possible to describe all code modifications for each language and/or development tool, migration guidelines will be given only for client applications that use an object-oriented data access interface (that is, either DAO or RDO). Visual Basic code samples will be given to illustrate these guidelines.

#### 6.1 What is OLE DB?

#### 6.1.1 OLE DB relies on COM

OLE DB is a low-level API that enables access to any data source - either relational or not - through Microsoft Component Object Model (COM). COM modelizes a software application (in our case, data source access) by a set of reusable software components (in the same way basic hardware components are used to build various complex chips). Each COM object specifies the functionality it supports by exposing interfaces – a set of semantically related methods (i.e. procedures or functions) that can be called by the client application. Each COM object must implement the **IUnknown** interface, containing the **QueryInterface** method that can be called by the client application to get a pointer on any other interface of the COM object.

#### 6.1.2 Consumers and Providers

There are two types of OLE DB applications: consumers and providers. A consumer can be any application that uses or consumes OLE DB interfaces. For example, a Microsoft Visual C++ $\circledast$  application that uses OLE DB interfaces to connect to a database server is an OLE DB consumer.



An OLE DB provider implements OLE DB interfaces; therefore, an OLE DB provider allows consumers to access data in a uniform way through a known set of documented interfaces. In a sense, an OLE DB provider is similar to an ODBC driver that provides a uniform mechanism for accessing relational data. OLE DB providers not only provide a mechanism for relational data but also for non-relational data types. Furthermore, OLE DB providers are built on top of Component Object Model (COM) interfaces that allow more flexibility; whereas ODBC drivers build on top of a C API specification.

#### 6.1.3 COM objects defined in OLE DB



The following illustration shows the core object model of OLE DB:

• Data Source Object

The data source object is the initial object returned from an enumerator (see "Enumerator Object," below) or instantiated by calling the OLE function **CoCreateInstance** with a given OLE DB data provider's unique class identifier (CLSID). It encapsulates the functionality of the ODBC environment as well as the connection and informational properties of the ODBC connection.

• Session Object

A session object defines the scope of a transaction and generates rowsets from the data source. If the provider supports commands, the session also acts as a command factory. The data source object can also support interfaces for describing schema information and for creating tables and indexes for providers





that support that functionality. Along with the data source object, the session encapsulates the functionality of the ODBC connection. Calling **IDBCreateSession::CreateSession** creates a session from the data source object. There can be multiple sessions associated with a data source object.

• Command Object

If a provider supports building and executing queries, it exposes a command object. A command object is generated from a session object. It is used to specify, prepare, and execute a DML query or DDL definition and associated properties. The command encapsulates the general functionality of an ODBC statement in an unexecuted state. There may be multiple commands associated with a single session.

• Rowset Object

A rowset object is a shared data object that represents tabular data, such as a result set returned by executing a query. Minimally, rowsets can be generated by calling **IOpenRowset::OpenRowset** on the session. All providers are required to support this minimal functionality. If the provider supports commands, rowsets are used to represent the results of row-returning queries. There are a number of other methods in OLE DB, such as the schema functions, that return information in the form of a rowset. A rowset encapsulates the general functionality of an ODBC statement in the executed state. There may be multiple rowsets associated with a single session or command.

The following objects are also defined in OLE DB. They provide recursive data source enumeration, enhanced transaction control, and extended error retrieval.

• Enumerator Object

Enumerator objects list the data sources and enumerators visible to that enumerator. This is similar to the information provided by **SQLDataSources**, except that the information can be recursive.

• Transaction Object

In addition to supporting **ITransactionLocal** on the session, providers that support transactions can optionally support the creation of a transaction object. Transaction objects provide more advanced transaction functionality, such as the registration of transaction notifications.

Error Object

In addition to the return codes and status information returned by each method in OLE DB, providers can optionally expose an OLE DB error object for extended error information, such as a description of the error or the appropriate SQLSTATE. This is similar to the information returned by **SQLError** or **SQLGetDiagRec**.



Each of these OLE DB objects has a number of attributes called *properties*. For example, a Rowset object has properties that describe the maximum number of rows that can be opened at one time.

Consumers set property values to request specific object behavior. For example, Consumers use properties to specify which interfaces they want a rowset to expose.

Consumers get property values to determine the capabilities of an object. For example, consumers use properties to determine the level of support for SQL.

#### 6.1.4 The ODBC Provider

The ODBC Provider maps OLE DB interfaces to ODBC APIs. With the ODBC Provider, OLE DB consumers can connect to a database server through the existing ODBC drivers in the following process: A consumer calls an OLE DB interface on the ODBC Provider. The ODBC Provider invokes corresponding ODBC APIs and sends the requests to an ODBC driver.

Because the ODBC Provider allows OLE DB consumers to use existing ODBC drivers, there may be some performance concern about the additional layer of the ODBC Provider on top of the existing ODBC driver manager. The design goal of the ODBC Provider is to implement all the functionality of the ODBC driver manager; therefore, the ODBC driver manager is not needed. However, the ODBC Provider still requires the ODBC Driver Manager to support connection pooling with ODBC applications.

#### 6.1.5 OLE DB AND ADO

Like the ODBC API, the OLE DB API is designed to provide optimal functionality in a wide variety of applications, but does not meet the requirement for simplicity. Microsoft ActiveX Data Objects (ADO) solves this problem by defining a programming model - the sequence of activities necessary to gain access to and update a data source – through three sets of objects which possess *methods* (actions that the object is capable of performing) and *properties* (attributes that define the object characteristics, or its behaviour when calling one of its methods). These three object models - ADO, Microsoft ADO Extensions for DDL and Security (ADOX), and Microsoft Jet and Replication Objects (JRO) provide the functionality found in DAO/RDO. Client applications can use any of these subsets of functionality to fulfil their needs.



#### 6.1.6 ADO Object models

#### 6.1.6.1 ADO: Data manipulation

Connection
Errors Error
Properties Property
Command
Parameters Parameter
Properties Property
Recordset
Fields Field
Properties Property
Record
Fields Field
Stream

The ADO object model contains objects for connecting to a data source and reading, adding, updating or deleting data.

The ADO **Connection** object defines a session for a user for a data source. This differs from DAO (RDO respectively) where the **Workspace** (**rdoEnvironment** resp.) object defines the session for a user and the datasource is defined by the **Database** (**rdoConnection** resp.) object.

The ADO **Command** object is similar to the DAO **QueryDef** and RDO **rdoQuery** objects, in that all objects can be used to execute an SQL statement against a data source. Likewise, the ADO **Recordset** object in both ADO and DAO, as well as the **rdoResultset** object in RDO, can be used to view the contents of a table or the results from executing an SQL statement.



#### 6.1.6.2 ADOX: Data definition and security



The ADOX object model contains objects for data definition (such as tables, views, and indexes) and creating and modifying users and groups. With ADOX, an administrator can control database schema and grant and revoke permissions on objects to users and groups.

The **Catalog** object is the container for the data definition collections (**Tables**, **Procedures** and **Views**) and the security collections (**Users** and **Groups**). This differs from DAO, in which the **Database** object is the container for the data definition collections and the **Workspace** object contains the security collections. The **Table**, **Index** and **Column** objects in ADO are roughly equivalent to the **TableDef**, **IndexDef** and **Field** objects in DAO.



RDO does not support any type of security or database schema modification because they are fully supported in the tools and utilities provided with the server systems (although it is possible to execute SQL queries which modify databases and tables).

Each **Catalog** object is associated with only one **Connection**, whereas a DAO **Workspace** (RDO **rdoEnvironment** resp.) may contain multiple **Databases** (**rdoConnections** resp.).

#### 6.1.6.3 JRO: Replication

It will not be detailed, as it can only be used with Microsoft Jet databases.

#### 6.2 Migration guidelines

ADO isn't automatically code-compatible with existing data access applications. While ADO encapsulates the *functionality* of DAO and RDO, many of the language elements must be converted over to ADO syntax. In some cases, this will mean only a simple conversion of some functions of the existing code. In other cases, it might be best to rewrite the application using ADO's new features.

Much of the functionality contained in the DAO and RDO models was consolidated into single objects, making for a much simpler object model. Because of this, however, it might be initially difficult to find the appropriate ADO object, collection, property, method, or event. Unlike DAO and RDO, although ADO objects are hierarchical, they are also creatable outside the scope of the hierarchy.

#### 6.2.1 Adding references to ADO / ADOX type libraries

#### NOTE:

It is assumed that the Microsoft Data Access Components (MDAC) 2.1 - which includes OLE DB components, ADO components and ODBC provider - has been installed on the client application development computer (it is shipped with Microsoft Visual Basic 6 or Microsoft Office 2000). If not, it can be downloaded from Microsoft web site (see <u>http://www.microsoft.com/data</u> for further information).

To run the Visual Basic code examples, the ADO/ADOX type libraries must be referenced in the client application by selecting "Microsoft ActiveX Data Objects 2.1 Library" (for ADO) and (if needed) "Microsoft ADO Ext. 2.1 for DDL and Security" in the Project->References menu.



If both DAO and ADO libraries are referenced, then the library to use must be explicitly specified when declaring object, as DAO and ADO include several objects with the same names. For example, both models include a Recordset object so the following code is ambiguous:

#### Dim rst as Recordset

To specify which object model is used, a qualifier must be included:

#### Dim rst as ADODB.Recordset

#### Dim rst as DAO.Recordset

#### 6.2.2 Opening a Database

Generally, one of the first steps in writing an application to access data is to open the data source. The following code demonstrates how to do it using a DSN in DAO, RDO and ADO.

#### DAO

```
Sub DAOOpenDatabase()
    Dim db as DAO.Database
    ' Use ODBCDirect with BULL ODBC Driver
    DBEngine.DefaultType = dbUseODBC
    Set db = DBEngine.OpenDatabase("", False, True,
"ODBC;DSN=dsnName")
    db.Close
End Sub
```

#### RDO

```
Sub RDOOpenDatabase()
    Dim cn as New RDO.rdoConnection
    cn.Connect = "DSN=dsnName"
    cn.EstablishConnection rdDriverNoPrompt
    cn.Close
End Sub
```



#### ADO

```
Sub ADOOpenDatabase()
Dim cnn as New ADODB.Connection
cnn.Open "Provider=MSDASQL;DSN=dsnName"
cnn.Close
End Sub
```

These two code listings for opening a database look somewhat different, but are not all that dissimilar. Aside from the fact that the objects have different names, the major difference is the format of the string passed to the method that opens the database.

Because Microsoft ODBC Provider is the default provider for ADO, if the Provider= parameter is omitted from the connection string, ADO will attempt to establish a connection to this provider.

The provider does not support any specific connection parameters in addition to those defined by ADO. However, the provider will pass any non-ADO connection parameters to the ODBC driver manager.

Because the Provider parameter can be omitted, it is therefore possible to compose an ADO connection string that is identical to an ODBC connection string for the same data source, using the same parameter names (DRIVER=, DATABASE=, DSN=, and so on), values, and syntax as in an ODBC connection string. A connection can be set with or without a predefined data source name (DSN).

Syntax with a DSN:

# "[**Provider=MSDASQL**;] **DSN**=name; [**DATABASE**=database;] **UID**=user; **PWD**=password"

Syntax without a DSN (DSN-less connection):

# "[**Provider=MSDASQL;**] **DRIVER=***driver*; **SERVER=***server*; **DATABASE=***database*; **UID=***user*; **PWD=***password*"

If a **DSN** is used, it must be defined through the ODBC Administrator in the Windows Control Panel. As an alternative to setting a **DSN**, it is possible to specify the ODBC driver (**DRIVER**=), the server name (**SERVER**=), and the database name (**DATABASE**=).

It is also possible to specify a user account name (**UID**=), and the password for the user account (**PWD**=) in the ODBC-specific parameters or in the standard ADO-defined *User ID* and *Password* parameters. If both the ADO and the ODBC-specific parameters are included for these values, the ADO parameters take precedence.



Although a **DSN** definition already specifies a database, a *DATABASE* parameter may be specified in addition to a **DSN** to connect to a different database. This also changes the **DSN** definition to include the specified database.

#### 6.2.3 Retrieving Data

Both ADO and DAO include a **Recordset** object (RDO includes an equivalent **rdoResult** object) that is the primary object used for retrieving data. A **Recordset** object represents a set of records in a table or a set of records that are a result of a query.

The **Recordset** object contains a **Fields** collection that contains **Field** objects, each of which represent a single column of data within a **Recordset**. The RDO equivalent object to the **Field** object is the **rdoColumn** object.

#### 6.2.3.1 Opening a Recordset

Like DAO and RDO, ADO **Recordset** objects can be opened from several different objects. In ADO, a **Recordset** object can be opened with the **Execute** method of the **Connection** Object, the **Execute** method of a **Command** object, or the **Open** method of a **Recordset** object. ADO **Recordset** objects cannot be opened directly from **Table**, **Procedure** or **View** objects. ADO **Recordset** objects opened with the **Execute** method are always forward-only, read-only **Recordsets**. To scroll data within a **Recordset** object, the **Recordset** object **Open** method must be used.

Whatever the interface used, a recordset type is specified by 3 parameters: cursor type, concurrency lock type and another interface-specific option.

• Selecting cursor type

Although the BULL ODBC driver does not support setting the cursor type, it is possible to have support for forward/backward scrolling and static (aka snapshots) cursors through (in DAO and RDO) the ODBC cursor library. This can be performed in DAO by setting the *DefaultCursorDriver* property of an ODBCDirect **Workspace** Object to dbUseODBCCursor, and in RDO by setting the *rdoDefaultCursorDriver* property of an **rdoEngine** object to rdUseODBC.

In ADO, using Microsoft Cursor Service for OLE DB performs this facility. To invoke that component, the *CursorLocation* property of **Connection** or **Recordset** object must be set to adUseClient.

• Selecting concurrency lock type

Setting the recordset lock type to any other value than default (read-only) is not supported by the BULL ODBC driver.



• Selecting option

The only common option (running a query asynchronously) is not supported by the BULL ODBC driver.

The following listings demonstrate how to open a forward-only, read-only **Recordset**, and then print the values of each field:

#### DAO

```
Sub DAOOpenRecordset()
   Dim db as DAO.Database
   Dim rst as DAO.Recordset
   Dim fld as DAO.Field
    ' Use ODBCDirect with BULL ODBC Driver
   DBEngine.DefaultType = dbUseODBC
   Set db = DBEngine.OpenDatabase("", False, True,
"ODBC;DSN=dsnName")
   ' Open the Recordset
   Set rst = db.OpenRecordset("SELECT * FROM CUSTOMERS",
dbOpenForwardOnly)
   ' Print the values for the fields in
   ' the first record in the debug window
   For Each fld In rst.Fields
       Debug.Print fld.Value & ";"
   Next
   Debug.Print
    ' Close the Recordset
   rst.Close
   db.Close
End Sub
```



#### RDO

Sub	RDOOpenRecordset()
	Dim cn as New RDO.rdoConnection
	Dim rst as RDO.Resultset
	Dim cl as RDO.rdoColumn
	cn.Connect = "DSN= <i>dsnName"</i>
	cn.EstablishConnection rdDriverNoPrompt
	' Open the Recordset
	<pre>Set rst = cn.OpenResultset("SELECT * FROM CUSTOMERS",</pre>
	rdOpenForwardOnly, rdConcurReadOnly)
	' Print the values for the columns in
	' the first result in the debug window
	For Each cl In rst.rdoColumns
	Debug.Print cl.Value & ";"
	Next
	Debug.Print
	' Close the Resultset
	rst.Close
	cn.Close
End	Sub

#### ADO

Sub	AD00penRecordset()
	Dim cnn as New ADODB.Connection
	Dim rst as New ADODB.Recordset
	Dim fld as ADODB.Field
	cnn.Open "DSN= <i>dsnName"</i>
	' Open the Recordset
	rst.Open "SELECT * FROM CUSTOMERS", cnn, _
	adOpenForwardOnly, adLockReadOnly
	' Print the values for the fields in
	' the first record in the debug window
	For Each fld In rst.Fields
	Debug.Print fld.Value & ";"
	Next
	Debug.Print
	' Close the Recordset
	rst.Close
	cnn.Close
End	Sub



#### 6.2.3.2 Navigating within a Recordset

A **Recordset** object has a current position. The position may be before the first record (**BOF**), after the last record (**EOF**), or on a specific record within the **RecordSet**. When retrieving information with the **Field** object, the information always pertains to the record at the current position.

DAO, RDO and ADO contain several methods for moving from one record to another. These methods are **Move,MoveFirst**, **MoveLast**, **MoveNext** and **MovePrevious**.

The following listings demonstrate how to use the **MoveNext** method to iterate through all of the records in the Recordset/rdoResultset object.

#### DAO

```
Sub DAOOpenRecordset()
   Dim db as DAO.Database
   Dim rst as DAO.Recordset
   Dim fld as DAO.Field
    ' Use ODBCDirect with BULL ODBC Driver
   DBEngine.DefaultType = dbUseODBC
   Set db = DBEngine.OpenDatabase("", False, True,
"ODBC;DSN=dsnName")
    ' Open the Recordset
   Set rst = db.OpenRecordset("SELECT * FROM CUSTOMERS",
dbOpenForwardOnly)
    ' Loop on each record of the Recordset
   While not rst.EOF
        ' Print the values for the fields in
        ' the current record in the debug window
        For Each fld In rst.Fields
            Debug.Print fld.Value & ";"
       Next
       Debug.Print
       rst.MoveNext
   Wend
    ' Close the Recordset
   rst.Close
    db.Close
End Sub
```



#### RDO

```
Sub RDOOpenRecordset()
   Dim cn as New RDO.rdoConnection
   Dim rst as RDO.Resultset
   Dim cl as RDO.rdoColumn
   cn.Connect = "DSN=dsnName"
   cn.EstablishConnection rdDriverNoPrompt
    ' Open the Recordset
   Set rst = cn.OpenResultset("SELECT * FROM CUSTOMERS", _
            rdOpenForwardOnly, rdConcurReadOnly)
    ' Loop on each result of the Resultset
   While not rst.EOF
        ' Print the values for the columns in
        ' the current result in the debug window
        For Each cl In rst.rdoColumns
           Debug.Print cl.Value & ";"
        Next
        Debug.Print
       rst.MoveNext
   Wend
    ' Close the Resultset
   rst.Close
   cn.Close
End Sub
```



#### ADO

Sub	ADOOpenRecordset()
	Dim cnn as New ADODB.Connection
	Dim rst as New ADODB.Recordset
	Dim fld as ADODB.Field
	cnn.Open "DSN= <i>dsnName"</i>
	' Open the Recordset
	rst.Open "SELECT * FROM CUSTOMERS", cnn, _
	adOpenForwardOnly, adLockReadOnly
	'Loop on each record of the Recordset
	While not rst.EOF
	' Print the values for the fields in
	' the current record in the debug window
	For Each fld In rst.Fields
	Debug.Print fld.Value & ";"
	Next
	Debug.Print
	rst.MoveNext
	Wend
	' Close the Recordset
	rst.Close
	cnn.Close
End	Sub

The code for iterating through the Recordset/Resultset in DAO, RDO and ADO is identical.

#### 6.2.3.3 ADO shortcuts to display and open Recordsets

In the previous example, the ADO code could be rewritten to use the **Recordset** object's **GetString** method to print the data to the Debug Window. This method returns a formatted string containing data from the records in the **Recordset**. Using **GetString**, the While loop in the previous ADO example could be replaced with the single line:

Debug.Print rst.GetString(adClipString, , ";")

This method is handy for debugging as well as populating grids and other controls that allow passing in a formatted string representing the data. **GetString** is also faster than looping through the **Recordset** and generating the string with Visual Basic code.



The ADO example could also have been rewritten more concisely by using the **Recordset** object's **Open** method's *ActiveConnection* parameter to specify the connection string rather than first opening a **Connection** object and then passing that object in as the *ActiveConnection*. The **Recordset** object's **Open** method call would look like this:

rst.Open "SELECT \* FROM CUSTOMERS", "DSN=dsnName", \_ adOpenForwardOnly, adLockReadOnly

Internally, these two mechanisms are essentially the same. When a connection string is passed to the **Recordset**'s object's **Open** method (rather than assigning a **Connection** object to the **Recordset** object's *ActiveConnection* property), ADO creates a new, internal **Connection** object. If the client application opens more than one **Recordset** from a given data source, or opens **Command** or **Catalog** objects, it is advised to create a **Connection** object and to use that object for the *ActiveConnection*: it will reduce the amount of resources consumed and increase performances.

#### 6.2.3.4 Using ActiveX controls

ADO includes ActiveX controls that offer the same functionality as DAO/RDO controls. For example, the ADO Data Control is the equivalent of the intrinsic Data Control and of the Remote Data Control. Likewise, the DataGrid control is code-compatible with the DBGrid control (with one exception: the DataGrid control doesn't support the DBGrid notion of "unbound mode").

#### 6.2.3.5 Error handling (in Visual Basic)

If ADO encounters an error, it does not populate the **Errors** collection. Instead, it uses the relevant programming language's native error mechanism to catch and display the error – in this case the Visual Basic Err Object. However, if the provider or underlying components generate errors, these will be populated in the ADO Errors collection. Therefore, both the Visual Basic Err object and the ADO **Errors** collection must be checked. Also, the Err object property values should be preserved because the process of examining and displaying the **Errors** collection could reset the Err object if the error handler is complex.

The Errors collection is available only from the Connection object.



# A

Access product, 3-1 architecture, 3-2, 3-4

# С

Cognos product,3-1

# D

DA 7 (NGEN) overview,1-1 Data Access 7 (NGEN) architecture,3-2, 3-4 functional description,3-1 gateway,3-3 databases accessing IDS/II,1-1 DDA Manager,3-3 DDW/ODBC API provided,3-3 DDW/ODBC2,1-2 installation required,2-1, 4-1 disk space requirements,1-2

# Ε

Excel product,3-1

# F

files

accessing UFAS files,1-1 functional description,3-1

## G

Gateway,1-2 GQL product,3-1

# Η

hardware requirements,1-2

# I

IDS/II databases access,1-1 read-only access,3-1 Impromptu product,3-1 installation prerequisites,2-1, 4-1 program,2-2, 2-3

# L

LOTUS 1-2-3 product,3-1

# Μ

memory requirements,1-2 Microsoft Windows 95,1-2 Microsoft Windows NT,1-2



# 0

OPEN 7 gateway,3-3 installation required,2-1, 4-1 OPEN SQL,3-1

# Ρ

PC/FOCUS product,3-1 Powerbuilder product,3-1 prerequisites installation,2-1, 4-1 procedures installation,2-3 purpose of DA 7 (NGEN),1-1

# Q

Quest product,3-1

# R

read-only access IDS/II databases,3-1 UFAS files,3-1 requirements disk space,1-2 hardware,1-2 memory,1-2 software,1-2

# S

software requirements,1-2

# Т

TCP/IP installation required,2-1, 4-1

# U

UFAS files access,1-1 read-only access,3-1

# V

Vision Pro product,3-1 Visual Basic product,3-1

# W

Windows 4GL product,3-1 Windows 95,1-2 Windows 98,1-2 Windows NT,1-2

#### Vos remarques sur ce document / Technical publications remarks form

Titre / Title : DA 7 (NGEN) on TCP/IP for XTA

N° Référence / Reference No. : 47 A2 89US Rev01

Date / Dated : June 2001

#### ERREURS DETECTEES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Vos remarques et suggestions seront attentivement examinées. Si vous désirez une réponse écrite, veuillez indiquer ci-après votre adresse postale complète.

Your comments will be promptly investigated by qualified personnel and action will be taken as required. If you require a written reply, furnish your complete mailing address below.

Remettez cet imprimé à un responsable Bull S.A. ou envoyez-le directement à : Please give this technical publications remarks form to your Bull S.A. representative or mail to:

Bull S.A. CEDOC Atelier de reprographie 357, Avenue Patton BP 20845 49008 ANGERS Cedex 01 FRANCE Bull HN Information Systems Inc. Publication Order Entry FAX: (800) 611-6030 MA30/415 300 Concord Rd. Billerica, MA 01821 U.S.A.