
OpenAccess SDK for .NET - Installation Guide

Part # OA-SDK-NET-V560-DOC-INST

© 1996-2007 OpenAccess Software, Inc., a wholly-owned subsidiary of Progress Software Corporation

This manual may not, in whole or in part, be copied, translated, or reduced to any electronic medium or machine-readable form without prior written consent, in writing, from DataDirect Technologies. The information in this manual is subject to change without notice, and DataDirect Technologies assumes no responsibility for any error that may appear in this document. The references in this manual to specific platforms are subject to change.

DataDirect and OpenAccess are trademarks or registered trademarks of Progress Software Corporation or one of its subsidiaries or affiliates in the United States and other countries. Any other trademarks or tradenames contained herein are the property of their respective owners.

OpenAccess ODBC client includes:

ICU Copyright © 1995-2003 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

Preface

About this Manual

This manual describes the use, installation, and configuration of the OpenAccess SDK for .NET products that include Local and Client/Server versions of the ODBC SDK, OLE DB SDK, .NET SDK, & JDBC SDK.

Specifically, the topics covered include:

- Installation and configuration of the SDK for .NET products
- Testing of the installed components
- Building and testing the sample IP

Notation Convention

The courier font is used for system-dependent keywords and commands to be typed by the user or for code samples and program output. The pipe character '|' is used as a shortcut notation and separator between menus, sub-menus, and items. As an example: 'File | Open' means the Open option from the File menu.

Technical Support

Technical support is available over the telephone, fax, and Internet. Please send e-mail and files to DataDirect Technologies. Please provide the OpenAccess product name, version, and the platform you are running on when calling/writing for technical support.

Telephone	UNITED STATES: 888-DDANSWR (888-332-6797) Others: see http://www.datadirect.com/support/contactus/phone/index.ssp
Email	supportlink@datadirect.com

Table Of Contents

INTRODUCTION	1-1
INSTALLING THE OPENACCESS SDK FOR .NET	2-1
WHAT IS INCLUDED IN THE OPENACCESS SDK FOR .NET	2-1
WHAT YOU NEED TO BUILD AN IP	2-2
INSTALLING THE CLIENT/SERVER SDK	2-2
INSTALLING THE LOCAL SDK	2-7
WHAT'S NEXT	2-9
LIST OF TABLES	
TABLE 2-1: SUPPORTED PLATFORMS	2-2
TABLE 2-2: PLATFORM SYMBOLS	2-2
TABLE 2-3: INSTALLING FROM CD-ROM	2-2
TABLE 2-4: OPENACCESS DIRECTORY STRUCTURE	2-3
TABLE 2-5: OPENACCESS DIRECTORY STRUCTURE	2-7

Chapter 1

Introduction

OpenAccess™ SDK for .NET is a suite of software components to allow you to develop client/server or local ODBC, OLE DB, .NET, & JDBC compliant interfaces to your data sources by programming in a .NET language (C#, C++, VB.NET). It includes everything you need to develop an interface and to test it from Microsoft Windows and Unix based client applications. The rest of the document refers to this product as the OpenAccess SDK for .NET.

OpenAccess SDK for .NET is available as a client/server solution or a local solution. In the client/server solution, the OpenAccess ODBC driver/OLE DB provider/.NET provider/JDBC driver runs on one system and the server portion runs on the system containing the database. To make your data source accessible through ODBC, OLE DB, .NET or JDBC over a network, all you have to do is develop a driver (IP) for your data source. All client/server communication issues are handled by the OpenAccess client/server components. The OpenAccess server component can be run standalone or embedded within your application.

In the local solution, the ODBC driver/OLE DB provider/.NET provider/JDBC driver directly links to the database interface. This solution is suitable for cases where the application and the data reside on the same system or where you want to implement a client/server solution using your network API for accessing the database. The code you write is 100% portable between the OpenAccess SDK for .NET versions. You can start with one version and easily move to the other.

The OpenAccess SDK for .NET can be used to implement interfaces to a SQL or non-SQL compliant data source. If your backend is SQL capable or you want to expose your existing query language through a standard API, then you will implement a NSQL IP class that exposes the oanet.sql.sqlip interface. If you will be using the OpenAccess SQL engine then you will implement a .NET IP class that exposes the oanet.sql.ip interface. In this document we refer to this as the .NET IP. The .NET IP uses our NDAM (.NET based DAM) to parse & execute SQL commands.

For Client/Server SDK for .NET, please go to *Chapter 2: Installing the Client/Server SDK for .NET*.

For Local SDK for .NET, please go to *Chapter 2: Installing the Local SDK for .NET*.

For writing your .NET IP, please go to *Chapter 3: Writing Your Own .NET IP*.

For .NET IP configuration, please go to *Chapter 4: .NET IP Configuration*.

Chapter 2

Installing the OpenAccess SDK for .NET

This chapter walks you through the steps to install the SDK and to set up to develop your IP. Always refer to the Release Notes shipped with your SDK for any updates to this document.

Installation of the OpenAccess SDK for .NET software system involves the following steps:

1. Installing the SDK on your development platform.
2. Testing the operations of the installed software.
3. If using the client/server version then installing the OpenAccess ODBC/OLE DB/JDBC/.NET Driver for Windows on your PC.
4. Building a sample IP and testing it.
5. Testing ODBC/OLE DB/JDBC/.NET driver for a sample database from a client application.

What Is Included in the OpenAccess SDK for .NET

Components included:

- OpenAccess ISQL - an interactive SQL application to allow you to query OpenAccess database servers.
- OpenAccess Server for .NET (client/server version)- a pre-built server to access the .NET IP object. The client communicates with this server process that in turn interacts with the .NET IP you implement. Also included is a oaserver.dll for embedding the OpenAccess Server functionality within another process.
- OpenAccess ODBC Driver for .NET (local version) - a pre-built ODBC driver DLL to access the .NET IP object.
- Example IPs - source code for example IPs to be used as templates.

What You Need to Build an IP

You will need .NET framework and at least a C# compiler. If you need to debug then Visual Studio .NET or other .NET debugger is required. Table 2-1 details the supported platforms. Please contact us if you have any questions.

Table 2-1: Supported Platforms

Platform
x86 - Microsoft Windows NT, 2000, XP, 2003

Installing the Client/Server SDK

Installation of the OpenAccess SDK for .NET requires a set of files to be copied to your hard disk and configuration files to be modified. The mechanism for installing files onto your system and performing the configuration is platform dependent.

1. Gather the following information:

- The client and server key on the License Certificate form
- The name you want to call your OpenAccess Server (must be a single word limited to 32 alphanumeric characters, may include "_").
- IP address/host name of your system (assuming you are on a TCP/IP network)
- The TCP/IP port you want the OpenAccess Server to listen on (1706 for example)
- Description of this server

2. Install the files from the media

Follow the instructions in Table 2-3 to start the installation. Refer to Table 2-2 for platform identifiers.

Table 2-2: Platform Symbols

Symbol	Substitution
{platform}	iwinnt x86/Microsoft Windows (WIN32)

Table 2-3: Installing from CD-ROM

Platform	Instructions
Windows 2000, XP, 2003	<p>From CD:</p> <ol style="list-style-type: none"> 1. Insert the CD-ROM and change to directory: {CD-ROM drive}\sdknet directory. 2. Run winnsrvr.exe 3. Follow the instructions provided by the installation program. <p>From compressed file:</p> <ol style="list-style-type: none"> 1. Copy the compressed file iwinnt_nsrvr.exe to a temporary location 2. Execute it to start the installation process 3. Follow the instructions provided by the installation program.

At this point, all files have been copied to the {root} directory (directory specified during the installation) and the basic setup is complete. Files on your system are organized as shown in Table 2-4. All directories labeled {platform} will change to the name of the platform the software is being installed on. Following ready to run components are provided:

- OpenAccess Server for .NET
- OpenAccess Interactive SQL for interactively submitting queries to the server
- Example 1 in C#, VB.NET, and Managed C++. Example 3 in C#.

Table 2-4: OpenAccess Directory Structure

Directory	Description
{root}	The root directory for OpenAccess products. Includes make files for any component that can be built
--bin --{platform}	Executables for {PLATFORM} oaserver.exe, oanet.sql.ip.dll
--config --{platform}	Files to setup the make environment for the various platforms. Also contains the INI files and the install scripts. Configuration and setup files
--doc	Contains ZIP (ndoc.zip) of Programmers Reference for .NET in Web Help and HTML help formats. Unzip ndoc.zip
--oanet --ip	Classes directory Sample IP libraries example1.dll, example3.dll etc.
--Example1	Sample C# IP with Static Schema
--Example3	Sample C# IP with Dynamic Schema
--memory	Memory C# IP
--mcp_Example1	Example 1 implemented in Managed C++ IP
--vb_Example1	Example 1 implemented in VB.NET
--template	Template .NET project to start writing IP in C#
--mcp_template	Template .NET project to start writing IP in Managed C++
--vb_template	Template .NET project to start writing IP in VB.NET
--schema	Files required for configuration of the OpenAccess components
--DYNAMIC	Schema files to be used as template for the dynamic schema IP. These provide support for Views.
--TEMPLATE	Schema files to be used as template for the static schema IP.
--NET	Schema files for example1 for testing the installation

3. Test the sample .NET IP that implements the oanet.sql.ip interface

An OpenAccess Server for .NET with access to the test database through the example.cs is installed in step 2. Also installed is a client program, OpenAccess Interactive SQL, to use for testing the server. For testing the client/server setup, we will run the OpenAccess Server and connect to it from the OpenAccess Interactive SQL running on the same machine. This tests the full client/server configuration.

On Windows platforms:

- a. Start the sample server:

Execute the OpenAccess Server for .NET by using the program group item OpenAccess Server to start the `oaserver.exe` program located in `{root}/bin/{platform}`.

- b. Start the interactive SQL program:

Execute the Interactive SQL `oaisql.exe` program located in `{root}/bin/{platform}`.

- c. Connect to your example1 database by typing:

```
ISQL> connect {server name}
```

Where `{server name}` is replaced by the name of the OpenAccess Server you specified during installation.

- d. Type:

```
ISQL> select * from curvalue;
```

to get a list of values.

- e. Exit from OAISQL by typing `exit` at the ISQL prompt.

- f. If above steps are successful, then the basic setup is complete.

- g. If you want to test connectivity from ODBC driver then continue at step 5. If you want to test connectivity from OLE DB provider then continue at step 6. If you want to test connectivity from JDBC driver then continue at step 7. If you want to test connectivity from .NET Data Provider then continue at step 8. Otherwise stop the OpenAccess Server by going in the window where it is running and hitting Ctrl+C (delete key on some platforms) and continue at step 9.

4. Test VB .NET client/server connectivity

As part of the SDK we include a fully functional IP written in VB.NET to access the sample database. The installation sets up the VB_NET entry in the OpenAccess configuration file `oadrd.ini` to connect to a sample IP `vb_example1.dll`. If you changed the default installation path while installing the SDK you will need to change the path to the sample `vb_example1.dll` by modifying the `CONNECT_STRING` property. Please run the `{root}/bin/{platform}/wrdadm` and make sure `CONNECT_STRING` of `vb_net1` database entry is correct and pointing to `{oa_root}\oanet\ip\vb_example1.dll`.

- a. Start the sample server:

Execute the .NET server by using the program group item OpenAccess Server to start the `oaserver.exe` program located in `{root}/bin/{platform}`.

If the server does not start up, then go to the troubleshooting chapter in *OpenAccess SDK Installation and Configuration Guide* to identify the problem.

- b. Start the interactive SQL program:

Execute the Interactive SQL `oaisql.exe` program located in `{root}/bin/{platform}`.

- c. Connect to your .NET database by typing:

```
ISQL> connect VB_net
```

```
select * from curvalue;
```

Exit from OAISQL by typing `exit` at the ISQL prompt.

- d. If above steps are successful, then the basic setup is complete.
- e. If you want to test connectivity from ODBC driver then continue at step 5. If you want to test connectivity from OLE DB provider then continue at step 6. If you want to test connectivity from JDBC driver then continue at step 7. If you want to test connectivity from .NET Data Provider then continue at step 8. Otherwise stop the OpenAccess Server by going in the window where it is running and hitting Ctrl+C (delete key on some platforms) and continue at step 9.

5. Test ODBC connectivity to the OpenAccess Server (.NET IP)

You need to install and configure the OpenAccess ODBC driver on one of your Windows systems. You may skip this step for now if you are not immediately interested in accessing the OpenAccess database on your server machine from a Windows client.

Please install OpenAccess ODBC Driver. Please refer to "*Installing The OpenAccess ODBC Driver*" chapter in *OpenAccess SDK Installation and Configuration Guide*.

Next we will run ODBC Interactive SQL, which is an ODBC compliant application, to test the ODBC configuration.

- a. Execute the ODBC version of the Interactive SQL program by using the program group item `Interactive SQL (ODBC)` to start the `odbcisql.exe` program located in `{root}/bin/{platform}`.
- b. Connect to the OpenAccess .NET SDK Server

```
ISQL> connect {server}
```
- c. Exit by typing `exit` at the ISQL prompt.

6. Test OLE DB connectivity to the OpenAccess Server (NET/NSQL IP)

Please install OpenAccess OLE DB Provider. Please refer "*Installing The OpenAccess OLE DB Provider*" chapter in *OpenAccess SDK Installation and Configuration Guide*.

Next we will run OLE DB Interactive SQL, which is an OLE DB compliant application, to test the OLE DB configuration.

You need to install and configure the OpenAccess OLE DB provider on one of your Windows systems. You may skip this step for now if you are not immediately interested in accessing the OpenAccess database on your server machine from a Windows client.

- a. Execute the OLE DB version of the Interactive SQL program by using the program group item `Interactive SQL (OLE DB)` to start the `oledbisql.exe` program located in `{root}/bin/{platform}`.
- b. To test .NET IP then connect to the OpenAccess database entry configured in above steps by typing:

```
ISQL> connect {server}
```

A list of providers will be displayed. From this list select "Automation Technology, inc. OLE DB Provider". Then select Connection tab and type `test_NET` in the Data Source field and press OK button.

```
ISQL> select * from curvalue;
```

- c. Exit by typing `exit` at the ISQL prompt.

7. Test JDBC connectivity to the OpenAccess Server (.NET)

You need to install and configure the OpenAccess JDBC driver on one of your Windows or UNIX systems. You may skip this step for now if you are not immediately interested in accessing the OpenAccess database on your server machine from a Windows client.

Please install OpenAccess JDBC Driver. Please refer "*Installing The OpenAccess JDBC Driver*" chapter in *OpenAccess SDK Installation and Configuration Guide*. If you want to test JDBC connectivity from a non-Windows platform then please follow the instructions in the ADK Installation Guide.

Next we will run JDBC Interactive SQL, which is a JDBC compliant application, to test the JDBC configuration.

- a. Execute the JDBC version of the Interactive SQL program located in `{root}/jdbc`.

```
{root}\jdbc> java jdbcisql
```
- b. To test .NET IP then connect to the OpenAccess database entry configured in above steps by typing:

```
ISQL> connect {server}
ISQL> select * from curvalue;
```
- c. Exit by typing `exit` at the ISQL prompt.

8. Test ADO.NET connectivity to the OpenAccess Server (.NET)

You need to install and configure the OpenAccess .NET data provider on one of your Windows systems. You may skip this step for now if you are not immediately interested in accessing the OpenAccess database on your server machine from a Windows client.

Please install OpenAccess .NET Provider. Please refer "*Installing The OpenAccess .NET Provider*" chapter in *OpenAccess SDK Installation and Configuration Guide*. When installing the client,

Next we will run .NET connect sample, which is a .NET compliant application, to test the .NET configuration.

- a. Execute the sample connect.exe program by using program located in `{root}/net`
- b. To test .NET IP then connect to the OpenAccess database entry configured in above steps by typing:

```
{root}\net> connect {server}
```

9. What's Next

Congratulations!!! The SDK for .NET software is correctly installed and operational.

Installing the Local SDK

Installation of the OpenAccess SDK for .NET requires a set of files to be copied to your hard disk and configuration files to be modified. The mechanism for installing files onto your system and performing the configuration is platform dependent.

1. Gather the following information:

- The client key on the License Certificate form

2. Install the files from the media

Execute the installation file `iwinnt_odbcnsdk_local.exe` and follow the instructions.

At this point, all files have been copied to the `{root}` directory (directory specified during the installation) and the basic setup is complete. Files on your system are organized as shown in Table 2-4. All directories labeled `{platform}` will change to the name of the platform the software is being installed on. A program group OpenAccess Local ODBC SDK for .NET is created and contains shortcuts for the Interactive SQL(ODBC) program, the OpenAccess Administration program.

Table 2-5: OpenAccess Directory Structure

Directory	Description
<code>{root}</code>	The root directory for OpenAccess products. Includes make files for any component that can be built
<ul style="list-style-type: none"> --bin <ul style="list-style-type: none"> --{platform} 	Executables for {PLATFORM} oadbcl.dll, oanet.sql.ip.dll
<ul style="list-style-type: none"> --config <ul style="list-style-type: none"> --{platform} 	Sample IP libraries example1_odbcl.dll, example3_odbcl.dll etc. Files to setup the make environment for the various platforms. Also contains the INI files and the install scripts.
<ul style="list-style-type: none"> --doc 	Configuration and setup files Contains ZIP (ndoc.zip) of Programmers Reference for .NET in Web Help and HTML help formats. Unzip ndoc.zip
<ul style="list-style-type: none"> --oanet <ul style="list-style-type: none"> --ip <ul style="list-style-type: none"> --Example1 --Example3 --Memory --mcp_Example1 --vb_Example1 --template --mcp_template --vb_template 	Classes directory
<ul style="list-style-type: none"> --schema <ul style="list-style-type: none"> --DYNAMIC --TEMPLATE --NET 	Sample C# IP with Static Schema Sample C# IP with Dynamic Schema Memory C# IP Example 1 implemented in Managed C++ IP Example 1 implemented in VB.NET Template .NET project to start writing IP in .NET Template .NET project to start writing IP in Managed C++ Template .NET project to start writing IP in VB.NET Files required for configuration of the OpenAccess components Schema files to be used as template for the dynamic schema IP. These provide support for Views. Schema files to be used as template for the static schema IP. Schema files for example1 for testing the installation

3. Install oanet.sql.ip.dll to the global assembly cache

If you will be using the ODBC driver from applications other than the Interactive SQL (ODBC) then you need to install the oanet.sql.ip.dll assembly in the global assembly cache.

Please run the following commands from Visual Studio .NET Command Window

```
> cd {oa_root}\bin\iwinnt
> gacutil /i oanet.sql.ip.dll
```

4. Test the sample .NET IP that implements the .NET Object interface

An example .NET IP that implements the .NET object interface is installed in step 2. Step 2 configures an ODBC DSN named `net1` that is mapped to the DataDirect OpenAccess Local for .NET driver. We will use this entry to test the .NET SDK from OpenAccess Interactive SQL.

- a. Start the interactive SQL program:

Execute the ODBC version of the Interactive SQL program by using the program group item `Interactive SQL (ODBC)` to start the `odbcisql.exe` program located in `{root}/bin/{platform}`.

- b. Connect to the `example1` database by typing:

```
ISQL> connect net1
```

- c. Type:

```
ISQL> select * from curvalue;
```

to get a list of values.

- d. Exit the Interactive SQL program by typing `exit` at the ISQL prompt.
- e. If above steps are successful, then the basic setup is complete and you are ready to develop your own IP to access your data source.
- f. If you want to test a sample Vb.NET IP then continue with the next step. Otherwise continue at step 6.

5. Test the sample VB.NET IP

As part of the SDK we include a fully functional VB .NET IP written in VB .NET to access any .NET compliant data source. By default the installation performed in above steps sets up the VB .NET DSN entry in OpenAccess configuration file `oadrd.ini` to connect to a sample IP.

Run the ODBC administration tool from control panel on your system and select the system DSN tab and click on `ADD` and select "DataDirect OpenAccess Local for .NET" from the drivers list. Give the ODBC Name as `vb_net1` and select `vb_net1` as the database name from the dropdown list.

- a. Start the interactive SQL program:

Execute the ODBC version of the Interactive SQL program by using the program group item `Interactive SQL (ODBC)` to start the `odbcisql.exe` program located in `{root}/bin/{platform}`.

- b. Connect to your VB .NET database by typing:

```
ISQL> connect vb_net1
```

```
select * from curvalue;  
to get the list of values
```

- c. Exit from Interactive SQL by typing `exit` at the ISQL prompt.
- d. If above steps are successful, then the basic setup is complete.

6. What's Next

Congratulations!!! The OpenAccess SDK for .NET software is correctly installed and operational.

What's Next

You have installed the SDK and tested the installation. You can now experiment with the included examples by referring to the accompanying documentation and then move on to developing your own IP.

If you are working in non-SQL IP mode (NET IP), then please refer to the *OpenAccess SDK Programmer's Guide* to design the IP. Refer to the OpenAccess SDK Programmer's Reference for .NET on details of the IP interface, the DAM API, and setting up to implement your own class.