Advanced Tracker

SQL Server Administration Documentation

Note

Service packs and updates for MDAC are available for download direct from www.Microsoft.com

Customers who prefer to use the Desktop version of SQL-Server 2000 must be prepared to perform all database maintenance using the **oSQL** command-line utility, as this version of SQL-Server does not come with graphical administration utilities. This document covers limited use of the **oSQL** utility.

Requirements

Server

- SQL-Server 2000 (Enterprise, Standard or Desktop) version 8.00.760 (Service Pack 3).
- SQL Server 2005 (Enterprise, Standard, Workgroup Express and Developer) version 9.0.1399.

- Microsoft Data Access Components (MDAC) 2.7 Service Pack 1 Refresh

Client

- Microsoft Data Access Components (MDAC) 2.7 Service Pack 1 Refresh

All applications will create their tables within a default database name of [Advanced Tracker] and this cannot be changed. The [Advanced Tracker] database must have a Collation setting of ‘`SQL_Latin1_General_CP1_CI_AS`’. If this is not the default collating setting for the server, see Appendix B which discusses how to override the server setting and ensure applications from Advanced Tracker maintain their required collation settings.

There may be further operating system component updates needed for any of the base requirements of Advanced Tracker software listed above.

SQL Server Version Check

Enterprise Manager or Management Studio
Connect to and locate the server and do a right-click->Properties. Version is identified as “Product Version”.

oSQL
Using oSQL (see Appendix A), execute the following query on the computer running the database server:

```
“Select @@Version"
```

Version will be displayed on the line containing the following:

```
Microsoft SQL Server 200? – ?.??.??? (Intel X86)
```

Where “200?” could be 2000 or 2005 and “?.??.???” will denote the SQL Server version. Don’t be misled; the version identified on the last line of the output denotes the version of the operating system.

MDAC Version Check

Microsoft has a free utility which will check the version of data components on your computer. Go to WWW.Microsoft.Com and search for “check for mdac version”. Your search results should include a HOW TO article where this utility can be downloaded.

The Server

SQL Server 2000 Desktop Edition

Base installation of SQL Server 2000 Desktop requires a couple of extra steps to ensure the database is accessible throughout the network.

Once installation has completed, restart the computer if required. Open Windows Explorer and browse to the folder “C:\Program Files\Microsoft SQL Server\80\Tools\Binn”. Within this folder are a couple of applications we will want to run.

SQLMANGR.EXE
This application is the SQL Server service manager. It puts an icon in the system tray which indicates the status of the SQL Server. Users are also able to start and stop the server, as well as tell Windows to auto-start the SQL Server service when the OS starts.

SVRNETCN.EXE
This application is the SQL Server Network Utility. The following protocols should be enabled for the server to accept client connections:

- Named Pipes
- TCP/IP
- Multiprotocol

SQL Server 2000 Standard and Enterprise Editions

From within Enterprise Manager, right click on the server->Properties. Ensure the following minimums:

- General tab, “Autostart SQL Server” must be on to ensure the server is always accessible, even after a restart of the computer.
- Security tab, authentication must be set to SQL Server and Windows. With update SQL Server components (see the *Client Administration Application* section below), the SQL Server can be set for Windows only authentication. In this scenario, the user is validated based on the Windows user and group memberships. The permissions for each Windows user and group are maintained within the SQL Server user list.

SQL Server 2005 Express Edition

Much like the SQL Server 2000 Desktop Edition, the light-weight SQL Server 2005 Express Edition installs with all network access support disabled. Go to Start menu->Run and enter

```
SQLServerManager.msc
```

to execute the SQL Server 2005 Configuration Manager. Ensure that your intended communication protocols are enabled for both the SQL Server 2005 Network Configuration and the SQL Server Native Client Configuration. If you’re not sure which protocols to use, the minimum of TCP/IP will allow Advanced Tracker software to work successfully.

You should also have the SQL Server service set to Automatic from the SQL Server 2005 Services node of the Configuration Manager to ensure the server is always accessible, even after a restart of the computer.

Users

On a day to day basis, a typical user of Advanced Tracker software will only require ‘`db_datareader’` and ‘`db_datawriter’` permissions to the `[Advanced Tracker]` database. There will be instances when elevated permissions to the database are required. The following sections detail the database Client Access File which specifies a users login.

Applications

Client Access File (attiData.Ini)

The attidata.ini file is a configuration file needed for the Advanced Tracker applications to communicate with the SQL server. This file needs to reside on computers running Advanced Tracker applications, typically the in system folder. The example below shows a configuration file that would be used on a computer running Employee Tracker, All the Goodies, Symcod Retrieval and Labor Tracker.

Further information pertaining to this file can be located below, under the heading “Client Administration Application”.

-- Start of file --

```
[Atg]
1a. “C:\Program Files\Atg"
1b. "Provider=SQLOLEDB; Server=Aux; User ID=SA; Password=t4evr; Trusted_Connection=No;"
1c. "Soap”
1d. “dbo”

[Etp]
1a. “C:\Program Files\Etp"
1b. "Provider=SQLOLEDB; Server=Aux; User ID=SA; Password=t4evr; Trusted_Connection=No;"
1c. "Soap"
1d. “dbo”

[Symcod]
1a. “C:\Program Files\Symcod"
1b. "Provider=SQLOLEDB; Server=Aux; User ID=SA; Password=t4evr; Trusted_Connection=No;"
1c. "Soap”
1d. “dbo”

[STP]
1a. "C:\Program Files\Stp”
1b. "Provider=SQLOLEDB; Server=Aux; User ID=SA; Password=t4evr; Trusted_Connection=No;"
1c. "Soap"
1d. “dbo”
```

-- End of file --

Definition of the file contents:

i. `[atg]` – specifies the Advanced Tracker application, where
   - `[etp]` = Employee Tracker,
   - `[atg]` = All the Goodies,
   - `[Symcod]` = Symcod Contention Collection application,
   - `[stp]` = Shop or Labor tracker.

   The square brackets constitute the beginning of an application connection block.

ii. 1a. “C:\program Files\ETP” – specifies the path to the application executable file.

iii. 1b. "Provider=SQLOLEDB; Server=Aux; User ID=*xxxxxx*; Password=*xxxxxx*; Trusted_Connection=No;"

PROVIDER

Values: **`SQLOLEDB`** or **`SQLNCLI`**

The SQL Native Client provider (SQLNCLI) is the updated provider for SQL Server 2005 and will connect to both SQL Server 2000 and SQL Server 2005.

The SQL Provider for OLE (SQLOLEDB) will connect to both SQL Server 2000 and SQL Server 2005, although a computer which previously used this provider to connect will no longer be able to use it once the SQL Native Client installed.

Whichever provider you specify within the connections file, Advanced Tracker software will first test the usability of what is entered and the connection details may be modified internally to use the correct provider. This may not always prove to be the best scenario, so the correct provider should be used in the connection file as much as possible.

SERVER

Values: *`Server.Name`*, *`Server.Alias`* or *`Server.IP`*

This value specifies the SQL Server to connect to. Use the “OSQL –L” command to query a network for available SQL Servers.

USER ID

Values: *`SQL Server login name`*

This property signifies the SQL Server login to use for authentication in a server setup to use mixed-mode authentication. If using Windows authentication, this property can be ignored and not included in the connection details.

PASSWORD

Values: *`SQL Server login password`*

This property signifies the SQL Server login password to use for authentication in a server setup to use mixed-mode authentication. If using Windows authentication, this property can be ignored and not included in the connection details.

TRUSTED_CONNECTION

Values: **`TRUE`**, **`FALSE`**, **`YES`** or **`NO`**.

This setting specifies whether to use Windows or SQL Server login authentication. A value of TRUE or YES will login using the current Windows login credentials. A value of FALSE or NO will login using the specified USER ID and PASSWORD.

The SQLOLEDB provider will accept TRUE, FALSE, YES or NO. The SQLNCLI will only accept YES or NO.

NETWORK LIBRARY

Values: **dbnmpntw, dbmssocn, dbmsspxn, dbmsvinn, dbmsrpcn, dbmsadsn, dbmsgnet, dbmsipcn**

The client has several possible ways it can connect to the SQL Server. Usually the default method is acceptable, but some instances may require modifying the networking library being used on the client. If the connection fails using the forced network library properties, there is potential that the client, the server or the network does not support the requested protocol. Your system administrator should be able to assist.

The most probable selections would be Named Pipes, TCP/IP or Shared Memory. TCP/IP would be the best choice in a slow LAN, WAN or dial-up network. Named pipes may perform better when network speed is not an issue, and Shared Memory can be used when both the SQL Server and application reside on the same computer.

Here's the breakdown of accepted values:

dbnmpntw	Named Pipes
dbmssocn	Winsock TCP/IP
dbmsgnet	VIA
dbmsipcn	Shared Memory
dbmsspxn	SPX/IPX *(SQL Server 2000 only)*
dbmsvinn	Banyan Vines *(SQL Server 2000 only)*
dbmsrpcn	Multi-protocol *(SQL Server 2000 only)*
dbmsadsn	Apple Talk *(SQL Server 2000 only)*

MARS CONNECTION

Values: **YES** or **NO**

MARS (Multiple Active Result Sets) is a new feature in SQL Server 2005 which attempts to reduce server load by reusing existing data connections. This may prove beneficial in some circumstances. This feature can only be used when explicitly using the SQL Native Client connection provider against SQL Server 2005.

iv. 1c. “Soap” – prefix for table entries within the Advanced Tracker database.

v. 1d. “dbo” – explicitly specifies the owner to use for all data activities. This line will expedite ownership resolution at the server, thereby ensuring duplicate tables are not possible, and further, potentially speeding up database element access.

Each application from Advanced Tracker has different requirements within the ATTIDATA.INI file. Here are the more probable entry requirements:

Employee Tracker [etp]
Shop Tracker [stp]
Labour Tracker [stp], [etp]
All the Goodies [atg], [etp]
Symcod: [symcod] *{, [etp]*, [stp]*, [atr]*}*
Shop Standard Retrieval [retrieve], [stp] *{, [etp]}*

Inquiry Master [etinqm]
Inquiry Employee [etinqm], [etinqe], [etp], [etpdscan] *{, [stp]}*

Record Export to Excel [etp]* (will action on the first entry in that block)
Access Tracker [atr], [etp]* (will action on the first entry in that block)

- The asterix (*) mark denotes that the application will action on the first entry in that block only.
- Items within {..} brackets are mandatory only for specific application.

First run or Updated Software

When new software is installed for the first time, or when updates for software are delivered, there may be a requirement for additional tables and/or fields within the Advanced Tracker database. If we have set out that the standard client access user has no permission to adjust the database schema, we will need to use System Administrator privileges to get the job done.

Given a computer which has been installed or updated with Advanced Tracker software, we would want to either:

- Edit the attidata.ini file and adjust the user id and password settings for each block within the file to reflect the sa or system administrator login if using SQL Server authentication.
- Log into a computer with a Windows account privileged to alter the database schema.

Then we can then first-time run the software, get the databases updated, and take the back out action necessary to reduce the level of permission.

Client Administration Application

File Locations

Once installed, go to the Start menu, Run and enter:

```
     AtCLAdmin clientfolders
```

This will bring up the “Advanced Tracker File Locations” dialog.

Here each client workstation can be adjusted for a “Data Template File (*.DTF) Repository”. By default (and with previous versions) the location for storage of these .DTF files is the Windows system folder.

Also, the location for the “Client Connection File (attiData.INI)” can be adjusted per client. Again, by default and in previous versions, the location for storage of the attiData.INI file is the Windows system folder.

Client Connection File (attidata.ini)

There are two states for the client connections file; secure or not. Old software maintained the client connection file as a plain ASCII file within the Windows system folder. This would present a security risk for the SQL Server as user id and password information was readily available. Enhanced steps have been taken to ensure this file can be secured.

Go to the Start menu, Run and enter:

```
          AtCLAdmin
```

This will bring up the “Client Connection File” dialog. On a first run against an old-style file, it will immediately display the contents of the file. Upon saving, you will be informed that the file has previously been stored in a less secure fashion. Here you can decide to encrypt the file and secure its contents.

Once secured, the user will be prompted to enter the password to view or alter its contents. When changes are being committed to disk, the user must supply the same password to succeed.

There is an Edit menu available when operating on an encrypted file. This Edit menu has one option which allows for altering of the file’s password. To accomplish this, the previous password must be confirmed, and then the user is able to enter the new password.

Sending Data to Advanced Tracker

Occasionally, we may require your Microsoft SQL Server data be sent to Advanced Tracker for support or testing purposes. This presents a challenge in that the underlying files containing the data can become rather large.

Here is our preferred method for achieving this:

Open Enterprise Manager or Management Studio and expand the tree to expose the “Advanced Tracker” database.

1. Right-click on the “Advanced Tracker” database, go to “All Tasks” in Enterprise Manager or “Tasks” in Management Studio and select “Backup Database…”
2. Verify the destination for the backup and ensure the switch labeled “Overwrite existing media” is turned on. Click the Ok button to begin the backup.
3. The resulting backup file will be large, most probably over 100Mb. Using a compression application (WinZIP, WinRAR, etc.) will shrink this file considerably, and make it more portable.
4. Contact your Advanced Tracker technical representative to work out the best possible delivery method and be ready with the size of the file resulting from steps 1-4, as this will be imperative to the delivery strategy.

Using oSQL (see appendix A), execute the following query:

```
“Backup Database [Advanced Tracker] To Disk =
'path/file'"
```

Path/file would be a drive:\folder\file.ext combination, for example C:\Temp\Backup.txt.

Understand that the Path/file combination must be a drive local to the server. The backup cannot be stored on a network drive.

For example, if I was to request a backup using the sa user, password being password and dropping the file to C:\Backup.bak I would type:

```
OSQL -U sa -P password -Q "Backup Database [Advanced
Tracker] To Disk = 'c:\Back.bak'"
```

Appendix A - The oSQL Utility

Throughout this document, there are instances indicating queries to run through the oSQL command-line utility to get or set various pieces of information. Users without access to GUI administration tools (typically would be using the Desktop version of SQL Server 2000, also called MSDE) could perform database operations via this command line utility.

This utility requires the user to be sitting at the computer where the data is served from, as it is a component installed with the database server.

The standard syntax for the utility is:

```
OSQL -U username -P password -Q "query"
```

username would be the user to log into the SQL Server with.
password would be the password for the user above.
query would be the stated query to run

Note that the –U, -P and –Q are all required to be capitalized.

Further note that oSQL which comes with SQL Server 2005 requires an additional parameter in the syntax, even if run from the intended server. This extended syntax is:

```
OSQL –S servername -U username -P password -Q "query"
```

servername being the name of the server to connect to.

All versions of oSQL can query the network for the published server name using the following:

```
OSQL –L
```

Appendix B – Forcing Collation

Advanced Tracker software will only run predictably when the database collation order is set to ‘`SQL_Latin1_General_CP1_CI_AS`’.

To force a different collation setting from the default of SQL Server, before any piece of Advanced Tracker software is run to generate the database schema, manually add the [Advanced Tracker] database from oSQL, Enterprise Manager or Management Studio. The initial creation of the database from these utilities allows for overriding the server default collation value and specifying one to use for all tables and fields within.