

Advanced Tracker Technologies Inc.

SQL Server Administration

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 command-line utility (see appendix A), as this version of SQL-Server does not come with graphical administration utilities.

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.
- SQL Server 2008 (Enterprise, Standard, Workgroup, Express and Developer) version 10.0.1600.22.

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

Client

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

Collation

All applications from Advanced Tracker 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 default setting and ensure applications from Advanced Tracker maintain their required collation settings. See Appendix B of this document for information on Forcing Collation if your installation of SQL Server requires a collating setting other than the one mentioned above.

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".

Command Line Utilities

Using the command line utilities, 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, 2005 or 2008 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 Or 2008 Express Edition

Much like the SQL Server 2000 Desktop Edition, the light-weight SQL Server 2005 And 2008 Express Editions install with all network access support disabled. Go to Start menu->Run and enter

For SQL Server 2005:

`SQLServerManager.msc`

For SQL Server 2008:

`SQLServerManager10.msc`

to execute the SQL Server Configuration Manager. Ensure that your intended communication protocols are enabled for both the SQL Server 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 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 user's 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 in the 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** or **SQLNCLI10**

The SQL Native Client provider (SQLNCLI and SQLNCLI10) is the updated provider for SQL Server 2005/2008 and will connect to SQL Server 2000, 2005 and 2008.

The SQL Provider for OLE (SQLOLEDB) will connect to SQL Server 2000, 2005 and 2008, 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" or "sqlcmd -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:

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

MARS CONNECTION

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

MARS (Multiple Active Result Sets) is a new feature in SQL Server 2005 and 2008 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 or 2008.

- 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 [etingm]
Inquiry Employee [etingm], [etinge], [etp], [etpdscan] {, [stp]}
Record Export to Excel [etp]
Schedule Export to Excel [etp]
Access Tracker [atr], [etp]* (will action on the first entry in
that block)
```

- The asterisk (*) 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 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.

The last option is “Restrictive Pathing”, which forces connection details to be selected via the Start In folder for each application and entry in the ATTIDATA.INI file. This is required to be turned ON for installations that are to access individual prefixes of data within the Advanced Tracker database.

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 the command line utilities (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:\Backup.bak'"
```

****Note* This command line statement can be setup in a Scheduled Task to automatically create backups on a regular basis for disaster recovery and is encouraged by Advanced Tracker.***

Appendix A - The oSQL Utility

Note SQL Server 2005 and 2008 replaced this command-line utility with one named SQLCMD. The features and syntax presented in this document are available in both applications.

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 the command line utilities, 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.

Appendix C – Database Management

Although SQL Server does a great job at keeping everything tidy and inline, it is still an administrative task to manage certain aspects of the database to ensure top efficiency and hard drive usage.

“Shrinking” The Database

All flavors of SQL Server allow for shrinking of the database which removes unneeded space on the hard drive and de-fragments the data pieces. This will ensure your SQL Server is running at his highest potential.

SQL Server 2000

1. Open Enterprise Manager
2. Expand the 'Microsoft SQL Servers' item to the [Advanced Tracker] database
3. Right-click on the [Advanced Tracker] database, ->All Tasks->Shrink Database
4. Perform the database shrink with the default values.

SQL Server 2005 And 2008

1. Open the Microsoft SQL Server Management Studio
2. Expand the Object Explorer item to the [Advanced Tracker] database
3. Right-click on the [Advanced Tracker] database, ->Tasks->Shrink->Database
4. Perform the database shrink with the default values.

SQL Server 2000 MSDE

1. Get into a command prompt and execute OSQL
2. Perform the following command:

```
USE [Advanced Tracker]
GO
DBCC SHRINKDATABASE(N'Advanced Tracker' )
GO
```

SQL Server 2005 and 2008 Express

1. Get into a command prompt and execute SQLCMD
2. Perform the following command:

```
USE [Advanced Tracker]
GO
DBCC SHRINKDATABASE(N'Advanced Tracker' )
GO
```

Ensure Database Element Integrity

Although the served-data nature of SQL Server has many advantages and limits the possibilities for data corruption, it can still occur from time to time.

SQL Server 2000

1. Open Enterprise Manager and click on the server which is housing the Advanced Tracker database.
2. Under the menu Tools, select 'SQL Query Analyzer'
3. In the Query window, type in:

```
DBCC CHECKDB ([Advanced Tracker])
```
4. Click on the menu Query->Execute

SQL Server 2005 And 2008

1. Open the Microsoft SQL Server Management Studio
2. In the Query window, type in:
DBCC CHECKDB ([Advanced Tracker])
3. Click on the menu Query->Execute

SQL Server 2000 MSDE

1. Get into a command prompt and execute OSQL
2. Perform the following command:
DBCC CHECKDB ([Advanced Tracker])
GO

SQL Server 2005 and 2008 Express

1. Get into a command prompt and execute SQLCMD
2. Perform the following command:
DBCC CHECKDB ([Advanced Tracker])
GO

The `DBCC CHECKDB` command will output numerous lines whichever method is used to execute the command. The most important part to pay attention to will be one of the final lines which will look something like the following:

```
CHECKDB found 0 allocation errors and 0 consistency errors in
database 'Advanced Tracker'.
```

If your query has found any allocation or consistency errors, you may have yourself an SQL Server which has encountered database corruption. Refer to **Appendix F: SQL Server Data Corruption** for how to correct this.

Appendix D - Server Migrations

The migration of an existing SQL Server to a different one is a fairly straight forward process. There are a couple of things to keep in mind:

1. This process outlined here will not work if the destination SQL Server is an earlier release. For example, data hosted on SQL Server 2005 cannot be mounted to SQL Server 2000, only another SQL Server 2005 and beyond.
2. There will be downtime as the migration occurs. Scheduling of when the migration will occur should take into consideration payroll cut off dates and any other time sensitive operations that Advanced Tracker software is responsible for.
3. SQL Server 2000 MSDE and the Express versions of SQL Server 2005 and 2008 do not allow for network connections by default. See page 5 for the necessary steps to allow this required behaviour.
4. Ensure the client software is compatible with the version of SQL Server you are moving too. Older software will not be able to connect to SQL Server 2008. Contact Advanced Tracker technologies if you have any questions or concerns about the version of software you are running and its compatibility with your target SQL Server.

Step 1. Disable automatic or always-running software

If you have any of the following pieces of Advanced Tracker software running, be sure to disable them during the migration process:

- All The Goodies (ATG)
- Symcod
- Retrieve
- Synel
- ProcessAlerts

Step 2. Inform users and request they stop work in Advanced Tracker software

Once everything is migrated, the users can continue to work on approvals or scheduling, and if all goes according to plan they should not experience any differences.

Step 3. Backup the original database

Create a FULL database backup, via command line utility (oSQL or SQLCMD), Enterprise Manager (SQL Server 2000) or SQL Server Management Studio (SQL Server 2005 and SQL Server 2008).

Step 4. Detach the original database

As with step 3, various tools are available to achieve this. Once this step is complete, the old server is no longer accessible. If things do not go

according to plan and the database needs to continue being hosted on the original server, a re-attach can be performed and no data should be lost.

Step 5. Restore the backup in the destination database

Take the backup produced in step 3 and move it to the new SQL Server. Using the tool of your choice, restore the backup to the new server. The name of the database **must** be **Advanced Tracker** without exception.

Step 6. Install software on the server

Even if you do not expect to run any Advanced Tracker software on the SQL Server, it's always a good idea to install and configure on that computer to ensure things are working as expected, before rolling out to the client computers.

Step 8. Install, configure and test the client software

Now that we have verified the database exists and works fine from the server itself, it's time to go around to all the client computers and install any necessary software, and configure the ATTIDATA.INI files to point at the new SQL Server. Each client should be tested after installation and configuration to ensure any issues are handled immediately and not found by the client computers' user at a critical point in time.

Step 10. Enable automatic or always-running software

Restart any applications disable during step 1.

Appendix E: Multiple Prefixes Through One Application

Overview

This appendix will outline the steps necessary to run to completely different sets of data, herein referred to as a prefixed data source, or prefix. This can be done entirely through one application, using distinct shortcut properties.

These setup steps do apply to all Advanced Tracker software, but for this example we will focus on Employee Tracker.

Setup

Given an existing SQL Server database containing two independent sets of data, each with it's own unique prefix, we would start by creating two shortcuts, both pointing to an existing Etp.Exe application.

First, you must enable Restrictive Pathing. Information on where this is set can be found earlier in this document under the section titled 'Client Administration Application'.

The properties for each shortcut must now be changed. For example, suppose our first prefix was "Company1" and the second prefix was "Company2". First we would make two empty folders. In this example, we place these empty folders off the installation folder for Employee Tracker:

```
C:\Program Files\ETP\Company1
C:\Program Files\ETP\Company2
```

Now we alter the "Start In" property for each of the two shortcuts to have one start in the first new folder, and the second shortcut starts in the second folder.

Now an adjustment to the attidata.ini file to something like:

```
[etp]
1a. "C:\Program Files\etp\company1"
1b. "Provider=SQLOLEDB; server=Auximines; user id=sa; password=0112rrdg;
trusted_connection=no;"
1c. "Company1"

2a. "C:\Program Files\etp\company2"
2b. "Provider=SQLOLEDB; server=Auximines; user id=sa; password=0112rrdg;
trusted_connection=no;"
2c. "Company2"
```

The "1a." and "2a." lines are used to reference which instance of Employee Tracker data to connect to via the "1c." and "2c." prefix lines!

Appendix F: SQL Server Data Corruption

A corruption with the SQL Server database can cause applications to fail or act in unpredictable ways, and can also cause severe restrictions in the efficiency of the database.

Information on how to check your SQL Server database for corruption issues can be found in **Appendix C – Database Management** under the section titled *Ensure Database Element Integrity*. You may also receive messages from Advanced Tracker software if it encounters corruption within the database.

1. Backup

Because corruption in the database may incur data loss, it's always a good idea to make a backup prior to attempting any corrections. Locate the section titled *Sending Data to Advanced Tracker* earlier in this document for the steps necessary to backup an SQL Server database.

2. Enter Single User Mode

Repairing of an SQL Server database requires the database to be in Single User Mode.

SQL Server 2000

1. Open Enterprise Manager and click on the server which is housing the Advanced Tracker database.
2. Locate the Advanced Tracker database, right-click on it and select Properties from the drop-down menu.
3. On the Options tab, check Restrict Access and select Single user.
4. Click [Ok] to commit the change.

SQL Server 2005 And 2008

1. Open the Microsoft SQL Server Management Studio
2. Locate the Advanced Tracker database, right-click on it and select Properties from the drop-down menu.
3. Select Options from the categories on the left.
4. Change the Restrict Access option to SINGLE_USER
5. Click [Ok] to commit the change.

SQL Server 2000 MSDE, SQL Server 2005 and 2008 Express

1. Get into a command prompt and execute OSQL for SQL Server 2000 MSDE or SQLCMD for SQL Server 2005 and 2008 Express.
2. Perform the following command:

```
ALTER DATABASE [Advanced Tracker] SET SINGLE_USER WITH  
ROLLBACK IMMEDIATE  
GO
```

3. Repair

Now that the database is in Single User mode, we can issue the repair command. Following the steps in **Appendix C – Database Management** under the section titled *Ensure Database Element Integrity*, replace the DBCC command in that section with the following one:

```
DBCC CHECKDB ([Advanced Tracker], REPAIR_ALLOW_DATA_LOSS)
```

4. Repeat

When the repair command is complete, you will get a report of the items found and corrected. Sometimes, you will see in the report messages stating that items could not be repaired until other repairs are performed first. This will require a repeat of the repair command in Step 3. It is advisable that you re-run the repair command until you see the following:

```
CHECKDB found 0 allocation errors and 0 consistency errors in
database 'Advanced Tracker'.
```

5. Multi User Mode

Now that the repair is complete, we need to set the database back to Multi User Mode.

SQL Server 2000

1. Open Enterprise Manager and click on the server which is housing the Advanced Tracker database.
2. Locate the Advanced Tracker database, right-click on it and select Properties from the drop-down menu.
3. On the Options tab, un-check Restrict Access.
4. Click [Ok] to commit the change.

SQL Server 2005 And 2008

1. Open the Microsoft SQL Server Management Studio
2. Locate the Advanced Tracker database, right-click on it and select Properties from the drop-down menu.
3. Select Options from the categories on the left.
4. Change the Restrict Access option to MULTI_USER
5. Click [Ok] to commit the change.

SQL Server 2000 MSDE, SQL Server 2005 and 2008 Express

1. Get into a command prompt and execute `OSQL` (for SQL Server 2000 MSDE) or `SQLCMD` (for SQL Server 2005 and 2008 Express).
2. Perform the following command:

```
ALTER DATABASE [Advanced Tracker] SET MULTI_USER WITH ROLLBACK
IMMEDIATE
GO
```