Module 14: Appendix A - Troubleshooting Common Administrative Issues

Contents:

Module Overview

Database administrators working with Microsoft® SQL Server® need to adopt the important role of troubleshooter when issues arise, particularly if users are being prevented from working. It is important to have a solid methodology for resolving issues in general, and to be familiar with the most common issues that can arise when working with SQL Server systems.

Objectives

After completing this module, you will be able to:

• Explain the SQL Server troubleshooting methodology.

• Resolve service-related issues.

• Resolve login and connectivity issues.

# Lesson 1: SQL Server Troubleshooting Methodology

Before starting to try to resolve any issue, it is important to apply a logical troubleshooting methodology in a consistent manner. Troubleshooting is often regarded as an art as much as a science. However, there are a number of characteristics common to all good troubleshooters. You should aim to develop or emulate those characteristics to ensure that you can successfully troubleshoot issues in your organization.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the characteristics of a good troubleshooter.

- Apply a troubleshooting methodology.

Discussion: Characteristics of Good Troubleshooters
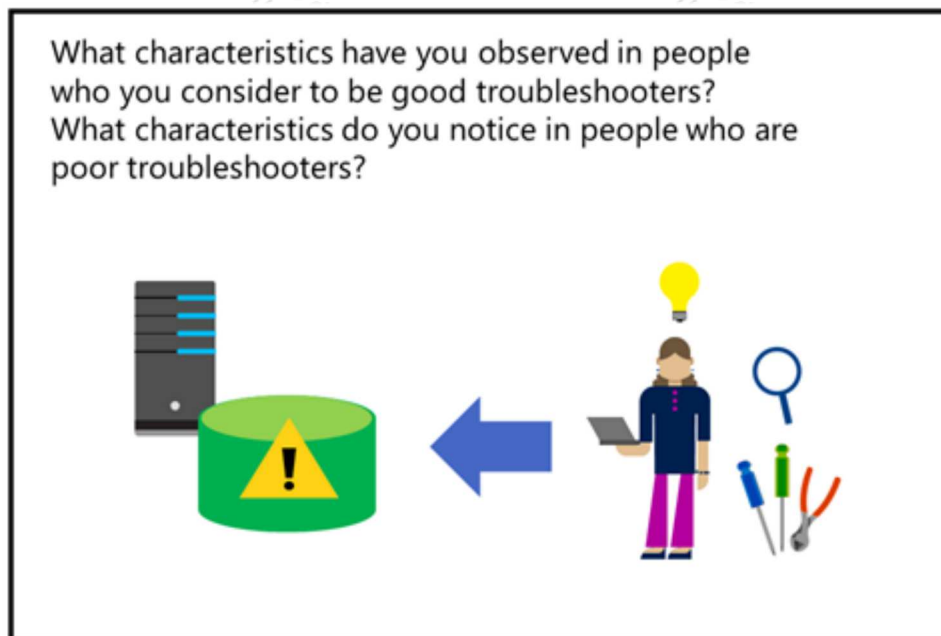
Consider the following questions:

- What characteristics have you observed in people who you consider to be good troubleshooters?

- What characteristics do you notice in people who are poor troubleshooters?

Characteristics of a good troubleshooter may include:

- Follows a repeatable and logical methodology.

- Stays calm, even in stressful situations.

- Is able to continually subdivide problems to move towards a solution.

- Knows about available tools and how to use them.

- Ensures the problem really is resolved.

Characteristics of a poor troubleshooter may include:

- Adopts a haphazard approach that does not follow a logical path

- Becomes more illogical the more that stress is applied.

- Makes assumptions that are not supportable.

- Quickly decides upon a cause for problems – without justification – and then tries to justify the selection of the cause.

- Assumes problems are resolved without justification.

Applying a Troubleshooting Methodology

- Investigate
  - Clearly define the issue as perceived by the user. How would you know if it is resolved?
  - What works? What doesn't work?
- Analyze
  - Discuss all potential causes. Which potential causes are likely? How could they be tested for and eliminated?
- Implement
  - Eliminate potential causes in descending order of likelihood
- Validate
  - Ensure that the issue really is resolved and document for future learning

A key characteristic of good troubleshooters is that they follow a clear methodology in a logical manner. There are many different methodologies you can use, but the following list describes four phases that are common to most of them.

Investigation Phase

This is a critical phase. Too many people shortcut this step and jump directly into finding solutions. However, before you can solve any problem, you need to be very clear in your understanding of what it is.

One very important concept in this phase is that the issue needs to be defined from the affected user's point of view, not from the assumed perspective of an IT professional. For example, there is no point telling a user that a system is working if they cannot use it for any reason, regardless of how your IT-based perspective might tell you that the system is working.

You need to understand what does and doesn't work. In this phase, a common mistake is to assume that, when a user complains that something doesn't work, that it ever did work. Make sure that there was a time when the issue didn't exist and find out when that was. Also find out about anything, no matter how unrelated it might seem at this point, that has changed since that time.

Finally, you need to know how the user would decide that the issue is resolved. A common troubleshooting error is to find a problem, assume that it is the cause of the issue, resolve that problem, and then conclude that the original issue is now settled.

Analysis Phase

In the analysis phase, you need to determine all the possible causes of the issue that you are trying to resolve. At this point, it is important to avoid excluding any potential causes, no matter how unlikely you consider them to be.

A discussion with someone else is often useful in this phase, particularly if that person is capable of constantly providing alternative viewpoints. The analysis phase often benefits from two types of people – one with an excellent technical knowledge of the product and another that constantly requires the first person to justify their thoughts and to think both logically and laterally.

Implementation Phase

In the implementation phase, you need to eliminate each potential cause. This process of elimination usually returns the best results when the potential causes are ruled out in order, from the most likely to the least likely cause.

The critical aspect of the implementation phase is to make sure that your reasons for eliminating potential causes are logically valid. If you reach the end of your list of potential causes and have not yet found a solution to the issue, you need to return to the analysis phase and recheck your thinking. If you cannot find a problem in your analysis, you might even need to go back to recheck your initial assumptions in the investigation phase.

Validation Phase

Too many people, particularly those who are new to troubleshooting, assume that problems are resolved when they are not. Do not assume that, because you have found and resolved a problem, this was the original one that you were aiming to solve.

In the investigation phase, you should have determined how the user would decide if the issue is resolved. In the validation phase, you need to apply that test to see if the issue really is resolved.

Documentation

After the problem is resolved, it is very important to make sure that everyone in the organization learns from what happened by creating relevant documentation.

Lesson 2: Resolving Service-related Issues

In the remainder of this module, you will see how to approach common types of issues that can arise when working with SQL Server systems.

SQL Server comprises a series of Windows® services. While troubleshooting issues with these services have much in common with those of other Windows services, there are some considerations specific to SQL Server. This lesson covers the types of issue involving problems with SQL Server services.

Lesson Objectives

After completing this lesson, you will be able to:

• Troubleshoot service-related issues.

• Use the SQL Server error log.

• Use Windows event logs.

Troubleshooting Service-related Issues

• Check Windows and SQL error logs

• If SQL Server can be started but not accessed:
  • Check for network-related issues
  • Try to access SQL Server via the DAC

• If SQL Server will not start:
  • Check the Windows system log
  • Check master and model databases for corruption
  • Check that the paths to tempdb files are accessible
  • Try to start the service from the command prompt

The most common service-related problem is when one of the SQL Server services will not start or cannot be accessed. As with all Windows services, the SQL Server services are not permitted to interact directly with the system console. This limits the options that are available to the service to advise you of problems. Also, it is uncommon for a user to be logged on at the console to receive any notifications, even if they were permitted.

Windows and SQL Server Logs

The most common cause of a service not being able to start is that the service account logon is failing for some reason. This may be due to an incorrect or expired password, or the account could be locked out. Logon failures for services typically appear in the Windows system event log. More complex issues, such as missing files, appear in the SQL Server logs. It is recommended, therefore, that you check both the Windows and SQL Server logs as the first

step in resolving service startup issues.

Other Service-related Issues

If SQL Server does start but cannot be accessed, the problem is likely to relate to network issues. In rare cases, the SQL Server scheduler can hang and stop accepting new connections. In this and similar situations, you should also try to connect to SQL Server by using the Dedicated Administration Connection (DAC) to access the instance and analyze the problem. Occasionally, you might need to kill individual sessions that are causing the problem.

If SQL Server will not start but the issue is not logon-related, check the following:

- Whether or not the SQL Server log files indicate that the **master** or **model** database is corrupt. If either is corrupt, follow the procedures to recover the databases.

- Whether or not the file paths to the **tempdb** database files are accessible. SQL Server recreates the **tempdb** database each time the server instance starts, but the path to the database files (as configured in the **master** database) must exist and be accessible.

- Whether or not you can start the instance by using the command prompt. If starting SQL Server from a command prompt does work, check the configuration of the service and make sure that the permission requirements are met.

SQL Server Error Log
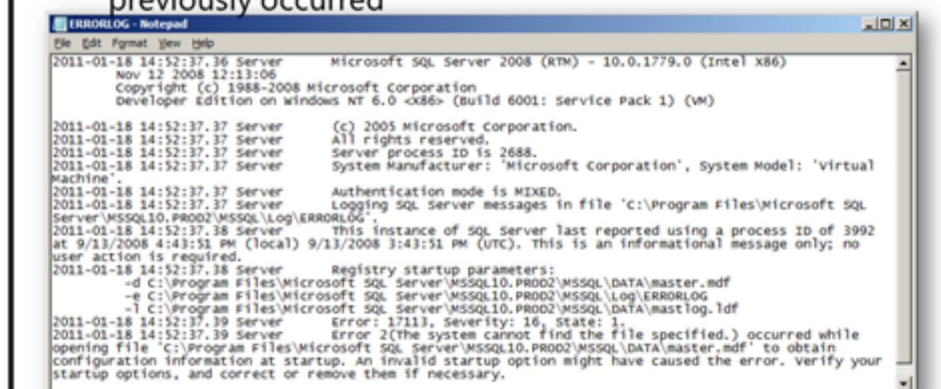
The SQL Server log will often provide detailed information about startup issues. You can view the log by using the Log File Viewer as part of SQL Server Management Studio. However, if you cannot start SQL Server, you are unable to use the log viewer to see the contents of the SQL Server logs. You can, however, open these logs in an editor such as Notepad.

The current error log usually provides the most useful information, though SQL Server keeps a set of archive log files which you can also review, as problems may have been occurring for some time.

Windows Event Logs
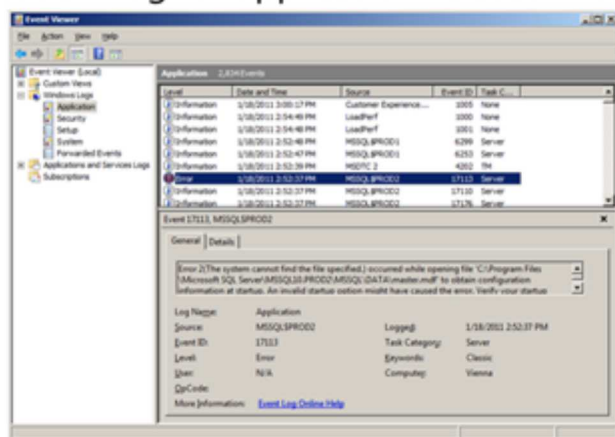
Windows services cannot interact directly with the console, so the key place to find information about issues is the Windows system event log. As with many other applications, over time SQL Server writes a substantial amount of information to the Windows application log. Therefore, you should check both these logs periodically, in case of errors that are logged but not producing symptoms apparent to users.

Demonstration: Troubleshooting Service-related Issues

In this demonstration, you will see how to troubleshoot a service-related issue.

Demonstration Steps

1.   Ensure that the MSL-TMG, MIA-DC, and MIA-SQL virtual machines are running, and log on to MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.

2.   On the taskbar, click the **SQL Server Management Studio** shortcut.

3.   In the **Connect to Server** dialog box, click **Connect**.

4.   On the **File** menu, point to **Open**, and then click **File**.

5.   In the **Open File** dialog box, navigate to **D:\Demofiles\AppendixA**, click **TroubleshootingServices.sql**, and then click **Open**.

6.   Follow the instructions contained in the comments of the script file.

7.   Close SQL Server Management Studio without saving changes.

Lesson 3: Resolving Login and Connectivity Issues

Connectivity and login issues are the final types that database administrators who work with SQL Server are commonly called upon to resolve. Users may have problems making connections to SQL Server, in relation to network names, protocols, and ports, or they may have issues with passwords that you need to resolve. In general, you should try to create logins that are based on Windows group membership. This removes the need for you, as a database administrator, to deal with most password and Windows account-related issues.

Lesson Objectives

After completing this lesson, you will be able to:

- Troubleshoot connectivity issues.

- Troubleshoot login failures.

Troubleshooting Connectivity Issues

- **Try to access using Shared Memory on the server:**
  - If no access via Shared Memory, troubleshoot the login and the service

- **Test the network connectivity**
  - Can the server name be resolved?
  - Can the network and the server be reached?
  - Is the Browser Service Running for Named Instances that are not using fixed ports?
  - Is the client configured to use the right protocol and settings?
  - Is a firewall blocking connectivity?

The most common issue of this type that you might need to resolve is that users cannot connect to SQL Server at all.

Eliminate the Network

The first step in trying to resolve these issues is to determine whether or not they are related to network connectivity. You can often eliminate the network from consideration by attempting to connect via a shared memory connection while local to the server. If you cannot access SQL Server via a shared memory connection, you will need to troubleshoot the login or the service. Login troubleshooting is discussed in the next topic. Service troubleshooting was discussed in Lesson 2.

Network-related Issues

If the login using shared memory succeeds, the problem is almost always network-related. In rare cases, you may see a problem with incompatible network libraries on the client and the server, but most problems are much less subtle.

Test	Actions
Can the server name be resolved?	The client system needs to be able to find the address of the server. In TCP/IP networks, this will usually involve a DNS lookup. Make sure you can ping the server that SQL Server is installed on by name and that you see an IP address returned by the ping command.

Test	Actions
Can the network and the server be reached?	While a server name might be successfully resolved to a network address, no connectivity may be possible between the client and server systems. On TCP/IP networks, make sure that you can successfully ping the server that SQL Server is installed on, assuming that it does not have the ability to echo ping commands disabled. In many cases, problems in this area will relate to incorrect subnet or default gateway/router addresses.
Is the browser service running for named instances that are not using fixed ports?	If you are using named instances of SQL Server, the client systems need to be able to resolve the names of the instances to a port number. By default, this is achieved by connecting to the SQL Browser service on UDP port 1434. If the SQL Browser service is disabled, the named instances need to be configured to use fixed ports and the client systems configured to access those ports, potentially by the creation of aliases. Make sure that the aliases are created for the correct network library, such as 32 bit versus 64 bit.
Is the client configured to use the right protocol and settings?	Check if the protocol that the client is attempting to use to connect to the server is actually enabled on the server and that the protocol settings are appropriate.
Is a firewall blocking connectivity?	Check to make sure that there is no firewall blocking the ports that you are trying to connect over. If a firewall is blocking your access, an exception or rule will likely need to be configured in the firewall.

Troubleshooting Login Failures

- Windows logins:
  - Is the Domain Controller available?
  - Can SQL Server communicate with the Domain Controller?
- SQL Server logins:
  - Is SQL Server configured for mixed mode authentication?
  - Is the password correct?
  - Is the login locked or is there a requirement to change password?
- General considerations:
  - Is the login enabled and does it have CONNECT permission?
  - Is the default/requested database available and access

The typical issue with this type of problem is that a user can establish a network connection to SQL Server but cannot log on. The troubleshooting actions that you need to perform depend upon the type of login being used:

- For Windows logins, make sure that SQL Server can connect to a Domain Controller to process them. To review potential issues, inspect the Windows logs.

- For SQL Server logins, make sure that SQL Server is configured for SQL Server Authentication. SQL Server logins can be created and enabled, even when SQL Server is configured for Windows Authentication only. The most common error returned in this situation says that a trusted connection is not available.

Also, when working with SQL Server logins, make sure that the supplied password is correct and that the login has not been locked out by account policy. Another common problem for SQL Server logins is that they might expire but the application trying to connect to SQL Server does not understand password expiry. This situation is common with applications that were designed for previous versions of SQL Server that did not implement account policy for SQL Server logins.

For both types, make sure that the login has permission to connect to SQL Server, as well as being able to access the database it is attempting to connect to. This check should include the default database for the login.

If a login problem is happening with a large number of different users, check to make sure that there isn't a logon trigger that is failing. When a logon trigger prevents users from connecting, the error message that is returned to interactive users indicates that a logon trigger prevented the connection.

Lab: Troubleshooting Common Issues

Scenario

You need to be able to resolve common issues with SQL Server processes and services at the time they are occurring. There are five exercises that create issues – you should attempt to troubleshoot and resolve as many of them as possible.

Objectives

After completing this lab, you will be able to:

- Troubleshoot and resolve SQL login issues.

- Troubleshoot and resolve backup issues.

- Troubleshoot and resolve job execution issues.

- Troubleshoot and resolve performance issues.

- Troubleshoot and resolve connection time issues.

Lab Setup

Estimated Time: 45 minutes

Virtual machine: **20462C-MIA-SQL**

User name: **AdventureWorks\Student**

Password: **Pa$$w0rd**

Exercise 1: Troubleshoot and Resolve a SQL Login Issue

Scenario

Users of the Promote application are complaining that it can no longer connect to the server. The application connects using the SQL login PromoteApp.

The main tasks for this exercise are as follows:

1.   Read the exercise scenario

2.   Troubleshoot and resolve the issue

Task 1: Read the exercise scenario

• Read the scenario for the exercise.

Task 2: Troubleshoot and resolve the issue

1.   Troubleshoot the issue.

2.   Resolve the issue.

3.   If you have difficulty, check the solution in the LAK.

Result: After this exercise, you will have investigated and resolved a SQL login issue.

Exercise 2: Troubleshoot and Resolve a Backup Issue

Scenario

A junior DBA has created a backup of the production AdminDB database to send to the development team for testing. Now users of that database are complaining they can no longer connect to it.

The main tasks for this exercise are as follows:

1. Read the exercise scenario

2. Troubleshoot and resolve the issue

Task 1: Read the exercise scenario

- Read the supporting documentation for the exercise.

Task 2: Troubleshoot and resolve the issue

1. Troubleshoot the issue.
2. Resolve the issue.
3. If you have difficulty, check the solution in the LAK.

Result: After this exercise, you will have investigated and resolved a backup issue.

Exercise 3: Troubleshoot and Resolve a Job Execution Issue

Scenario

Users are complaining that the Get File List job in SQL Server Agent is failing to execute.

The main tasks for this exercise are as follows:

1. Read the supporting documentation for the exercise

2. Troubleshoot and resolve the issue

Task 1: Read the supporting documentation for the exercise

• Read the exercise scenario.

Task 2: Troubleshoot and resolve the issue

1.   Troubleshoot the issue.

2.   Resolve the issue.

3.   If you have difficulty, check the solution in the LAK.

Result: After this exercise, you will have investigated and resolved a job execution issue.

Exercise 4: Troubleshoot and Resolve a Performance Issue

Scenario

The performance of the AdminDB database has been steadily reducing since the database was deployed. Queries that once took seconds are now taking minutes.

The main tasks for this exercise are as follows:

1.   Read the supporting documentation for the exercise

2.   Troubleshoot and resolve the issue

Task 1: Read the supporting documentation for the exercise

• Read the exercise scenario.

Task 2: Troubleshoot and resolve the issue

1.   Troubleshoot the issue.

2.   Resolve the issue.

3.   If you have difficulty, check the solution in the LAK.

Result: After this exercise, you will have investigated and resolved a performance issue.

Exercise 5: Troubleshooting and Resolving a Connection Issue

Scenario

A strange situation is occurring with the CityDetails database. Most databases work slower as more users are added. However, the CityDetails database performs the worst when only a single user connects to it. The connection time for just this database can be very long and sometimes it times out – other databases are okay.

The main tasks for this exercise are as follows:

1.   Read the supporting documentation for the exercise

2.   Troubleshoot and resolve the issue

Task 1: Read the supporting documentation for the exercise

• Read the exercise scenario.

Task 2: Troubleshoot and resolve the issue

1.   Troubleshoot the issue.

2.   Resolve the issue.

3.   If you have difficulty, check the solution in the LAK.

Result: After this exercise, you will have investigated and resolved a connection issue.

Review Information

Review Question(s)

Module Review and Takeaways

Best Practice:

1. Monitor your system and store historical data using Data Collector, for example, for easier troubleshooting.

2. Clearly identify underlying problems rather than fighting symptoms.

3. Apply deadlock monitoring using SQL Trace.

Review Question(s)