Quest® MessageStats® 7.8

Deployment Guide

Contents

MessageStats Architecture Overview

- Introducing the MessageStats Components
- MessageStats Database
- MessageStats Server
- MessageStats Reports
- Communication Between Components
- Communication Requirements for MessageStats Reports

Introducing the MessageStats Components

MessageStats consists of several components, each of which performs a different function when MessageStats gathers information from a messaging environment.

You can install all the MessageStats components on the same computer. However, there are scenarios that might require that you distribute some or all of the MessageStats components to other computers. The following is a description of the MessageStats components and their responsibilities.

For information about prerequisites for installing the components and permissions required to run MessageStats, see the *MessageStats Quick Start Guide*.

MessageStats Database

The MessageStats database is the repository for all the gathered information. It is hosted within a Microsoft SQL Server database. If you have installed report packs, the core database is extended to include tables and columns specific to the report pack.

There is one MessageStats database in a MessageStats installation.

MessageStats Server

The MessageStats Server consists of the following three components:

- MessageStats Task Processors
- MessageStats Scheduler Service
- MessageStats MMC Console

MessageStats Task Processors

In MessageStats, gathering units called Task Processors perform the following functions:

- collect information from the messaging environments.
- push the information to the MessageStats database so the information can be viewed through the MessageStats Reports.

A server on which task processors are installed is called a Task Execution Server. A Task Execution Server connects to the Exchange environment and gathers information from it.

You can install additional Task Execution Servers to distribute the gathering load as the amount of information to be gathered increases.

MessageStats Scheduler Service

The Scheduler Service coordinates the task processors that exist on each Task Execution Server. The Scheduler Service schedules the task processor activities (gathering tasks). The service also updates the MessageStats database with the status and progress of each task processor.

There is only one MessageStats Scheduler Service per MessageStats database.

MessageStats MMC Console

The MMC Console is the user interface that is used to manage the MessageStats gathering components. You create, update, and delete gathering tasks using the console. You also use the console to obtain an overall view of the gathering state of a MessageStats installation.

A MessageStats MMC Console can manage multiple MessageStats implementations (communicating through the MessageStats Scheduler Service). Also, multiple MessageStats consoles can be used to manage one MessageStats implementation.

MessageStats Reports

MessageStats Reports are presented through a web site that is hosted on an IIS (Internet Information Services) server. The web reports retrieve information from the MessageStats database.

You can have multiple MessageStats Reports web sites per MessageStats installation, though not on the same computer.

Communication Between Components

This section describes the type and frequency of communication that occurs between the various MessageStats components. By understanding the communications requirements between the various components, you can determine the topology and deployment options for a MessageStats installation.

Between Database and Reports

Between Task Execution Servers and Database

Between Task Execution Servers and Exchange Servers

Between Scheduler Service and a Task Execution Server

Between Console, Scheduler Service, and Database

Between Database and Reports

MessageStats Reports request and display information from the MessageStats database. This involves a high volume of communication between the two components. It is therefore recommended that the MessageStats database server and MessageStats Reports web server be closely connected.

For optimal performance, Microsoft recommends having SQL Server and IIS on the same computer. However, if the database and reports components are on separate computers, there must be high connectivity to ensure acceptable performance for MessageStats Reports.

Between Task Execution Servers and Database

The task processors on the Task Execution Servers gather data from the Exchange environment and store the data in the database. Since MessageStats provides information in an optimized manner, Task Execution Servers retrieve information from the MessageStats database for use in an aggregation decision. However, most communication is from the Task Execution Server (through the task processors) to the MessageStats database.

Communication between these components is through OLE DB, an RPC-based database communication vehicle. Performance of RPC-based communication is sensitive to network obstructions. Therefore, a clear connection should be maintained between these components.

Between Task Execution Servers and Exchange Servers

Though the Exchange environment is not part of MessageStats, it is important to understand communication between Exchange and the Task Execution Servers. Typically, you locate a Task Execution Server near the MessageStats database. However, Exchange servers are often grouped geographically. You might locate a Task Execution Server closer to the Exchange servers to achieve overall gathering performance.

To determine whether to locate the Task Execution Server closer to the database or the Exchange environment, compare gathering times using both models.

Between Scheduler Service and a Task Execution Server

The MessageStats Scheduler Service and the Task Execution Server operate like a manager and a worker. The Scheduler Service tells the Task Execution Server what to do and when to start. Once the Task Execution Server is active, it reports its current state and progress to the Scheduler Service.

The volume of communication between the Scheduler Service and Task Execution Server is low and these components need not be close. However, since the communication is RPC-based, the communication path should not contain network obstructions. If a firewall is present, you must open ports to enable communication.

You can install the MessageStats Scheduler Service on its own server but it is more efficient to install it with the initial task processor (Task Execution Server). Additional task processors can be installed on additional servers as needed.

Adding More Task Execution Servers

If the amount of allocated work becomes too much for one Task Execution Server to perform, the Scheduler Service requires additional workers. If the daily gathering of information takes close to 24 hours, it is time to add another Task Execution Server.

Additional Task Execution Servers improve the gathering time in two ways:

- They provide additional resources (memory, storage) to perform the work.

- They can be located closer to the Exchange servers from which they gather data, thereby improving communication performance.

There are many variables that affect the time required to collect data. The biggest variable is network usage; performance can vary greatly at different times and days of the week. Close observation of collection times is the best way to determine when additional Task Execution Servers are required.

Between Console, Scheduler Service, and Database

The administration function of the MessageStats MMC console requires it to communicate with the MessageStats Scheduler Service and with the MessageStats Database.

The MCC console communicates with the Scheduler Service to manage gathering tasks and schedule new work. Communication is through DCOM, which is RPC-based.

The MMC console reports the real-time progress of active gatherings using information it retrieves from the MessageStats database. Communication is through the OLE DB driver, which is RPC-based. The greater communication volume between the console and database requires a good connection between these components.

Communication Requirements for MessageStats Reports

The requirement to install MessageStats Reports on a separate computer depends on the amount of report use. Two main factors affect performance:

- The number of users that directly access the MessageStats Reports web site.

- The number of report subscriptions that are set up and executed on a daily basis.

A subscription generates specific reports on a regular schedule. These reports are sent by email or posted on a web site. Subscriptions affect the web site performance in the same manner as a user who directly views a report.

When a large number of MessageStats Reports users are supported, MessageStats Reports should be installed on a separate computer.

Planning for Deployment

Deployment Considerations

Before you deploy MessageStats in any organization, consider the factors that can affect MessageStats data collection, data storage, and reporting:

- From a collection perspective, factors include network latency and the number of Exchange-related objects (such as organizations, routing groups, connectors, servers, distribution groups, mailboxes, and public folders) to be collected.
- From a storage perspective, factors include the amount of historical statistics you want to maintain and the detail level of the information that you want to gather.
- From a reporting site perspective, factors include the number of users that will directly access the site, and the number of report subscriptions that will automatically generate scheduled reports.

The following checklist outlines the decisions that you must make before you install MessageStats.

Pre-Deployment Checklist

Use the following checklist to plan your pre-deployment activities

- Create an Exchange Server List
- Determine Exchange Server Latency
- Estimate the Exchange Tracking Log Sizes

- Identify Data Collection and Storage Requirements
- Determine the Number of Task Execution Servers
- Locate the MessageStats Scheduler Server
- Locate the Task Execution Servers
- Locate the MessageStats Database Server
- Locate the MessageStats Reports Server
- Determine the Tracking Log Compression Requirements
- Plan the Data Collection Schedule
- Determine the Requirements for Gathering

Create an Exchange Server List

To scope out your MessageStats deployment, you need information about your Exchange organization topology, and about the network latency between your Exchange servers and the servers that will host the MessageStats data collectors.

Create a list of your Exchange servers, showing the server functions and physical geography. (To determine the location of the Exchange servers, you can use the Microsoft Exchange Administrator.)

Determine Exchange Server Latency

Once you have a list of Exchange servers, collect ping statistics and server connectivity details from your Exchange servers to the following:

- candidate servers that may host the MessageStats Scheduler Service
- candidate locations of the MessageStats data collectors (Task Execution Servers).

This information will assist in determining the optimal locations of the MessageStats Scheduler Server and of the Task Execution Servers.

Estimate the Exchange Tracking Log Sizes

For each Exchange server in the list, verify that Exchange tracking is turned on and that the tracking log folder is shared. The default location of the folder is as follows for the various releases:

Exchange 2010

```
C:\Program Files\Microsoft\Exchange Server\V14\TransportRoles\Logs\ MessageTracking
```

Exchange 2013/2016/2019

```
C:\Program Files\Microsoft\Exchange Server\V15\TransportRoles\Logs\ MessageTracking
```

To estimate the average size of each tracking log for each Exchange server

1 View each Exchange server tracking log share.

2 Create a file list from each share.

3 Calculate the average size of the tracking logs per server.

Exchange servers usually keep seven days of tracking logs. A week's amount of log files can provide a reliable average per server.

Use this information to obtain insight about the disk space needed on each Task Execution Server to analyze and process all the tracking logs. This information also helps to decide whether to use the MessageStats tracking log compression utility to reduce network traffic.

Identify Data Collection and Storage Requirements

Data requirements are affected by two considerations:

- What information is gathered from what Exchange objects?
- How much historical information must be retained?

This information can assist in estimating the storage requirements. You can use the MessageStats database size estimation tool which is available from the Support portal at https://support.quest.com/.

What information is gathered from what Exchange objects?

MessageStats can collect a number of Microsoft Exchange object characteristics, some of which you can exclude from the gatherings.

In addition to the Exchange core information that is gathered by a Default Gathering task, information that can be gathered includes: distribution group membership, detailed audit information, and public folder information.

How much historical information must be retained?

To estimate the storage capacity that is needed on the SQL Server that hosts the MessageStats database, you need to know how much historical reporting is required. By retaining historical information, you can use MessageStats to view trend and forecast information for your Exchange organization.

Determine the Number of Task Execution Servers

You should distribute computing resources when gathering information to ensure that MessageStats can deliver the information in the quickest and least intrusive way possible. Consider the following:

- The number of required Task Execution Servers is proportional to the number of Exchange objects from which data is collected.
- The Task Execution Servers must have good connectivity to the MessageStats SQL database and to the Exchange servers.
- Collections must complete within a 24-hour cycle so that MessageStats does not get backlogged in Exchange tracking log processing.
- Too many Task Execution Servers trying to connect to the SQL database at the same time can cause deadlocks and timeouts on data storage.

It is recommended that you install one Task Execution Server for each dispersed geographic area in your organization, provided that the servers have reasonable SQL connectivity and sufficient computing resources to handle the work.

Locate the MessageStats Scheduler Server

The MessageStats Scheduler Server coordinates tasks that collect the information and manages the configured task schedules.

Select the location of the scheduler server based on reasonable network connectivity with the Task Execution Servers and the database server. Good connectivity is required because:

- Task Execution Servers regularly communicate their execution status to the scheduler server.

- The scheduler server stores task status information in the database server.

One method to test connectivity to Task Execution Servers is to use ping statistics from a varying number of servers located in different areas of the world to the Task Execution Server locations. Locate the MessageStats Schedule server where you find an acceptable average speed.

Locate the Task Execution Servers

Task Execution Servers collect data as directed by the scheduler server. The type of data collected depends on the task that is executed. Since the Task Execution Server is responsible for storing the collected information into the database, database connectivity is a primary concern.

A Task Execution Server should be located close to the Exchange servers from which it collects data. Since a Task Execution Server can be configured to use an alternate Exchange connection server to collect the information related to Exchange objects, proximity to the alternate Exchange server is also a consideration.

Locate the MessageStats Database Server

The location of MessageStats database server is critical within the deployment because all the other MessageStats components depend heavily on this storage component.

For best results, locate the SQL Server that hosts the MessageStats database within a data center that has the highest connectivity to the Task Execution Servers, the scheduler server, and the IIS reports server.

Locate the MessageStats Reports Server

Ideally, the MessageStats Reports IIS server should reside on the same computer as the SQL server that hosts the MessageStats database.

Co-location avoids moving information multiple times between servers and clients. Multiple hops can cause Windows security problems, as acknowledged by Microsoft. Clients tend to lose their credentials when accessing the reports site, and may reach the SQL Server as anonymous users and be unable to access the report information.

If co-locating the IIS report server and SQL database server is not possible, you must configure the SQL server to use SQL authentication. You must also configure the file used to create the data link for the reports to use a special SQL account so that the reports can access the MessageStats database as valid consumers.

Determine the Tracking Log Compression Requirements

Network administrators have specific tolerances for network traffic. Larger organizations can be more or less tolerant of the type of traffic that flows over the network.

Exchange tracking logs can grow very large, in the range of 10 to 30 gigabytes and may require compression before the log files are transported to the Task Execution Server for processing. Make the decision to compress

tracking logs in consultation with your Exchange administrator. Compressing log files has a very small footprint on the Exchange server.

The MessageStats compression utility (QMSDeployment.exe, QMSCompress.exe) can reduce the size of the tracking log file by up to 80%, which is significant when transporting logs that are gigabytes in size. The compressed logs are not decompressed on the Task Execution Server. They are processed in stream, which means that they do not require extra disk space on the servers for processing.

For information about configuring the MessageStats compression utility on an Exchange server, see the chapter titled "Compressing Tracking Log Files" in the *MessageStats Administrator Guide*.

Plan the Data Collection Schedule

MessageStats depends on the Exchange tracking logs to deliver the activity and volume-based reports. The Default Gathering task, which includes the tracking log gathering, must not be scheduled until the tracking logs are closed. Typically, tracking logs are closed at midnight UTC, so Exchange objects can be collected at any time during the day. However, those objects will be dated on the UTC date they are retrieved.

A worldwide deployment must be tuned in consideration of business activities. To minimize the impact of data collection, tasks must be scheduled to run during off-business hours for the geographic area from which data is collected.

Collection must be planned to complete within a 24 hour UTC cycle. If tracking logs become backlogged and do not complete within 24 hours, collection and subsequent reports will not be accurate.

Consider the time periods when the reports web site will be accessed the most and avoid that time period when scheduling data collection.

Determine the Requirements for Gathering

You must determine the requirements for gathering the Exchange tracking logs and for gathering the Exchange attributes.

Optimal Gathering of Exchange Tracking Logs

Tracking logs are copied to Task Execution Servers to be analyzed and summarized into useful information for the activity-related reports. The analysis can create a large amount of information to be stored in the database, which should be located near the server doing the processing.

A best practice is to remotely compress the tracking logs and to copy the compressed log files to a central location for processing and storage.

The following diagram provides an optimal design for collecting tracking log files, which can include compression, allowing the log file processing to be executed near the database server.
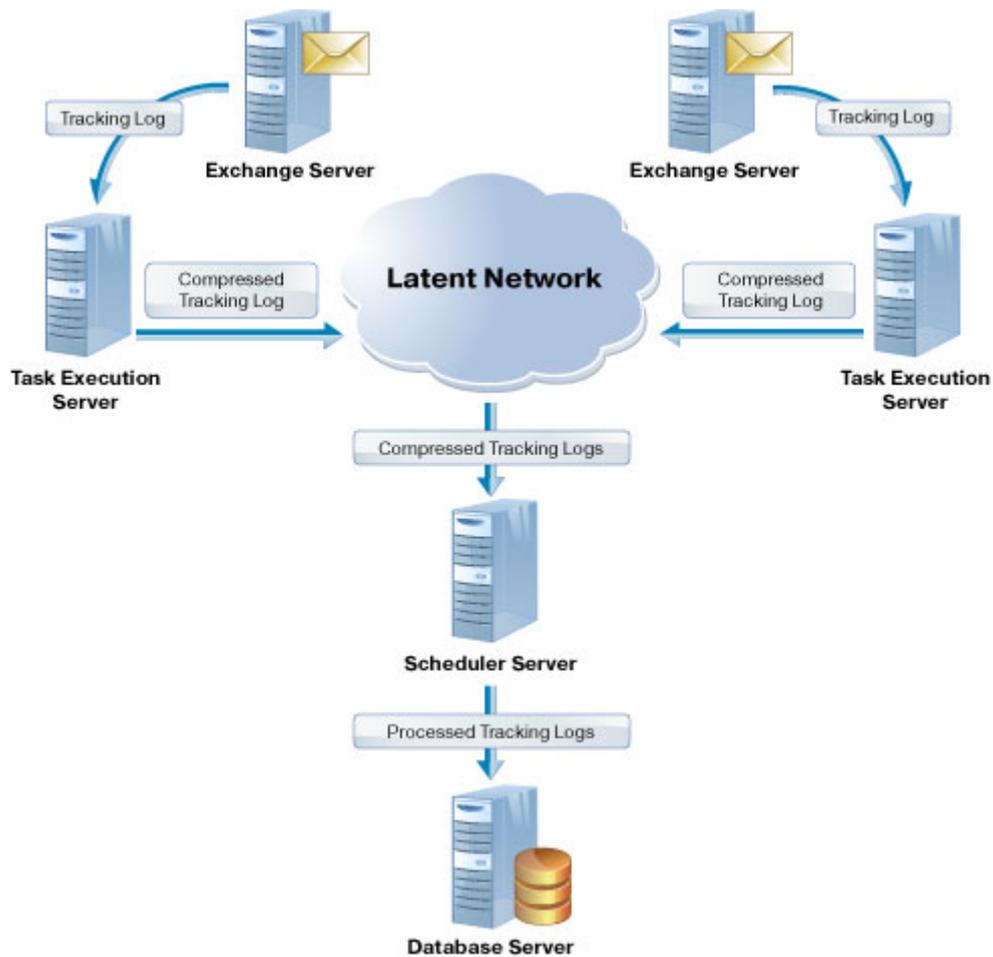
Figure 1. Optimal design for tracking log collection.

Optimal Gathering for Exchange Attributes

Exchange attributes are collected and stored with little processing. This information is best collected near the Exchange servers themselves.

The following diagram shows how, by positioning the Task Execution Servers geographically near and by connecting to the Exchange servers that contain the information, you can optimize the amount of time required to perform a data collection.
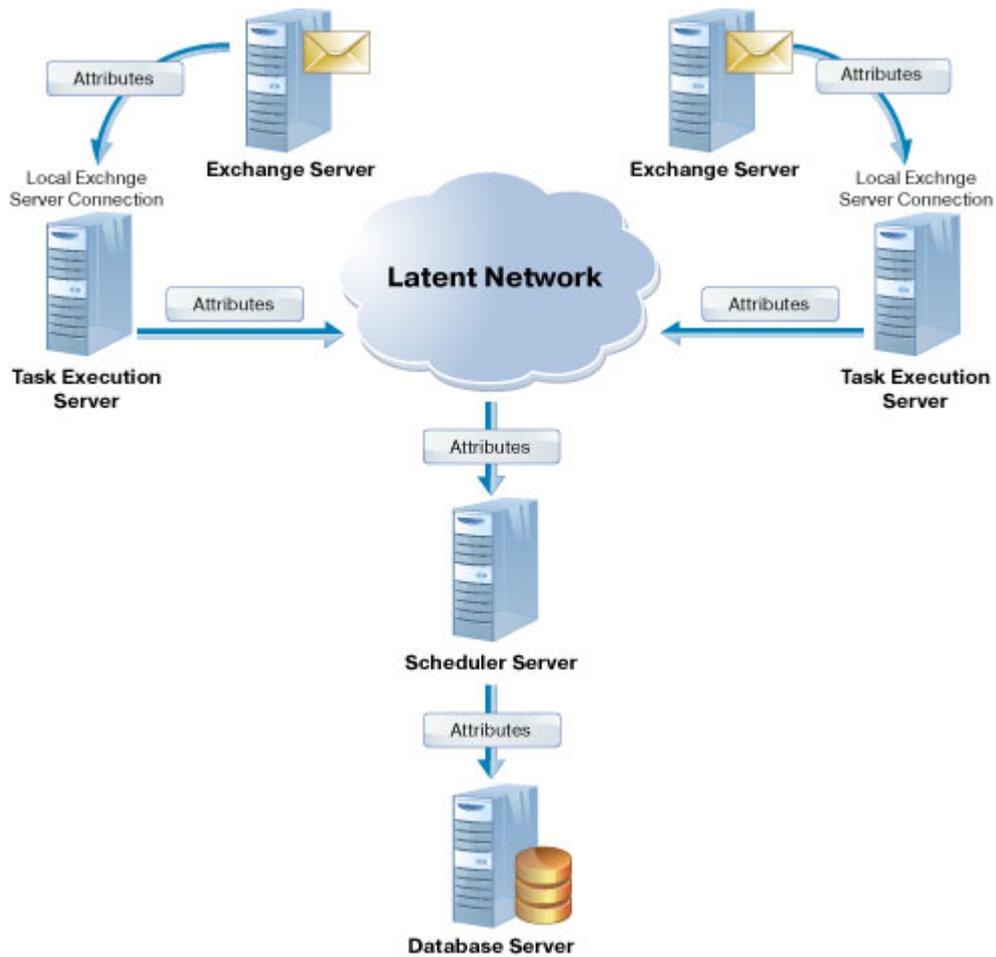
Figure 2. Optimal design for collecting Exchange attributes.

Deployment Recommendations and Solutions

- Deploying MessageStats in Small or Large Installations
- Recommendations for Large Installations
- Deploying a Distributed Installation Using Windows Authentication
- Deploying MessageStats Across Forests
- Manually Creating the MessageStats Database in a Distributed Installation

Deploying MessageStats in Small or Large Installations

Depending on the size of your Exchange deployment and whether it is situated in one geographic location or in several locations, you must determine whether to install MessageStats on a single server or on several servers.

Single Geographic Location

For a small-to-medium business that has its messaging system in a single location, the number of report consumers, either direct or indirect, is usually low. In this case, you can install all the MessageStats components on a single server. If the gatherings take longer than 24 hours, then you can install additional Task Execution Servers on other computers to reduce gathering times.

If your Exchange environment is in a single geographic location, you can install MessageStats on a single server or distribute MessageStats components on more than one server, depending on the number of mailboxes.

- If you have 10,000 mailboxes in your organization, you can use single server deployment.
- If you have more than 10,000 but less than 50,000 mailboxes, you should consider a distributed installation. You can place the MessageStats Server (which includes the scheduler service and the task processors) and the MessageStats database on different computers.
- For 50,000+ mailboxes, you should have a MessageStats Server, MessageStats Database, and additional MessageStats task processors on separate computers.

Distributed Task Processors

This scenario applies to distributed locations where the locations are connected through internal networks with no firewalls between them. The largest location is considered the central location. You can install the MessageStats Scheduler Service, the MessageStats Database, at least one Task Execution Server, and MessageStats Reports on this computer.

For smaller locations, identify the locations that share similar connectivity. If some locations share a common pipe back to the central location, install an additional Task Execution Server to improve gathering performance.

Locations must have a reasonable connection to the MessageStats database. If the database connection is poor, you can install multiple Task Execution Servers in that location.

Multiple Geographic Locations

You can distribute your task processors on different servers if your Exchange environment is distributed across more than one geographic location. A server on which task processors are installed is known as a task execution server.

It is recommended that you run the Tracking Log Gathering task on the task execution server that is closest to the MessageStats database server. Tracking logs should be compressed. Run all other gatherings on the task execution server closest to the Exchange environment.

Recommendations for Large Installations

MessageStats is designed to work efficiently immediately after it is installed. The Default Gathering is the quickest gathering, stores the least amount of information, and is designed for performance.

When MessageStats is run in a large-scale environment with more than 50 Exchange servers, large Exchange tracking logs can cause the database to grow significantly in a short amount of time, despite MessageStats optimizations to store a minimum amount of data.

Database growth can lead to gathering problems if there is insufficient memory available on the database server, or if the Microsoft SQL Server version is not set to support more than 2 GB of RAM. To process large amounts of data, SQL Server must have the appropriate resources available.

Database performance can be affected by platform software that is not enterprise quality.

Recommended Hardware and Software

For large installations, it is recommended that you run 64-bit Microsoft SQL Server 2012 or later (Enterprise Edition). The 64-bit version ensures that sufficient RAM is allocated.

MessageStats calculates statistics for each mailbox and against each Exchange server, and the calculations require locks on several tables at once. MessageStats performs the calculations at gathering time rather than at report time to ensure report rendering performance.

Even on a server with 4 GB of RAM, the full 4 GB is not available to programs. Regardless of the amount of physical memory, Windows uses a virtual address space of 4 GB, with 2 GB allocated to user-mode processes (such as applications) and 2 GB allocated to kernel-mode processes (such as the operating system and kernel-mode drivers). Since there are usually other user applications on the server, only about 1.3 GB of RAM is likely dedicated to SQL, rather than the 2 GB maximum that Standard Edition uses.

It is recommended that you use a computer with a minimum of 16 GB of RAM, and the 64-bit version of Microsoft SQL Server 2012 or later, Enterprise Edition, when running in large scale environments.

For a detailed list of the hardware and software prerequisites, see the *MessageStats Quick Start Guide*.

Deploying a Distributed Installation Using Windows Authentication

Typically, if you install the MessageStats database and the MessageStats reports server on separate computers, you would use SQL authentication.

However, if you want to use Windows authentication, you must set up Kerberos delegation from the MessageStats Reports server to the MessageStats Database server.

The following steps explain how to set up Kerberos delegation for installations in which the MessageStats database and reports are installed on separate servers. Procedures are provided for each step.

1. Register the MSSQLSvc SPN with the run-as domain account.

2. Enable delegation from the Reports Server to the Database Server.

3. Specify Windows authentication in the QMSReports.udl file.

4. Enable IIS Settings and Restart IIS.

1. Register the MSSQLSvc SPN with the run-as domain account.

To set up delegation to MessageStats database server, the MSSQLSvc Service Principal Name (SPN) for the SQL Server instance that hosts the MessageStats database must be registered with the run-as domain account. The SPN must be in the following format:

MSSQLSvc/*fully-qualified server DNS name*:*TCPport*

For example, if the server's fully qualified DNS name is QMS-DB.QMS-RP.QC.CA.QST and the port number is 1433, the SPN would be:

MSSQLSvc/QMS-DB.QMS-RP.QC.CA.QST:1433

When the SQL Server service runs as a machine account (such as LocalSystem, NetworkService, or LocalService), the SQL Server service automatically registers the appropriate MSSQLSvc SPN with the domain machine account. In this case, you do not need to register the SPN manually.

However, when the SQL Server service runs as a domain user account, you must register the MSSQLSvc SPN manually with the AD user object. To determine the required SPN, you need the following information:

- the fully-qualified DNS name of the server

- the TCP port of the SQL Server instance hosting the MessageStats database.

To find the TCP port of the SQL Server instance, you can view the IP Addresses property page for the TCP/IP protocol in the SQL Server Configuration Manager.

To determine the TCP Port Number

1. On the SQL server that hosts the MessageStats database, open the SQL Server Configuration Manager.

2. Select **Protocols for** the instance that hosts the MessageStats database and select **TCP/IP** in the list.

3. Click the **IP Address** tab and view the TCP Port number.

Once you know the TCP port and fully-qualified server name, use the SetSPN utility to register the SPN with the run-as account.

For example, suppose the SQL Server instance is on TCP port 1433 of server QMS-DB.QMS-RP.QC.CA.QST and is running as the domain account QMS-E10\db-svc. You would register the MSSQLSvc SPN with the domain account by entering the SetSPN command as follows:

```
setspn –A MSSQLSvc/QMS-DB.QMS-RP.QC.CA.QST:1433 QMS-E10\db-svc
```

Register the SPN with only one Active Directory object. Delegation will not work properly if the SPN is registered with more than one AD object. To verify this, you can use the SetSPN utility included with Windows.

The SetSPN utility has a -X switch that shows any SPNs that are registered to more than one AD object.

2. Enable delegation from the Reports Server to the Database Server.

The AD account used by the Web Reports application pool in IIS must be enabled for delegation to the MSSQLSvc SPN of the MessageStats database SQL server. MessageStats creates the Web Reports application pool to "run as" NetworkService. Consequently, you must enable the report server's AD account for delegation to the MSSQLSvc SPN.

To enable the report server AD Account for delegation to the database MSSQLSvc SPN

1 Open Active Directory Users and Computers and navigate to the properties for the MessageStats Reports server object.

2 Select the **Delegation** tab.
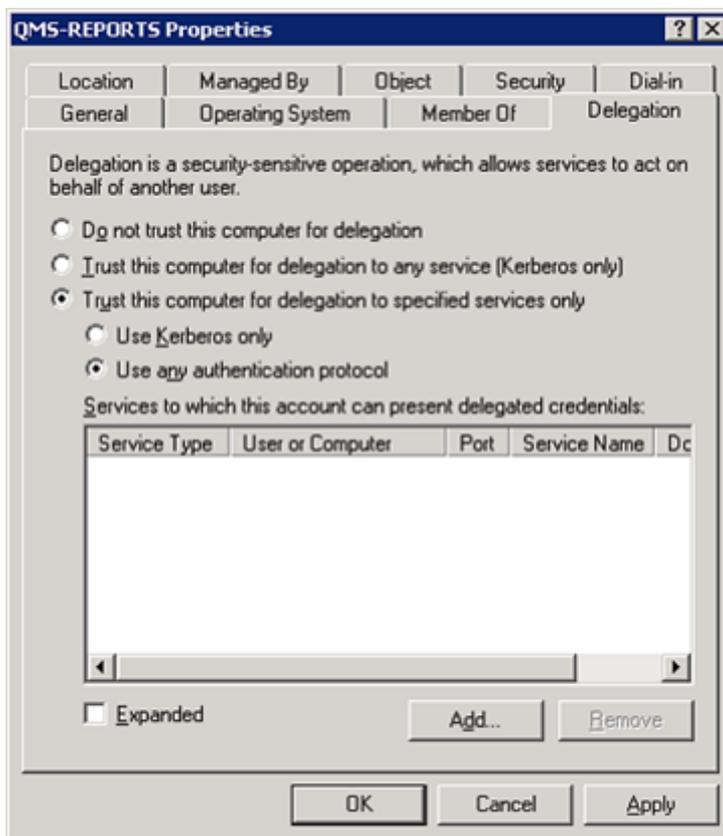
3 Select **Trust this computer for delegation to any service** to enable delegation from the server AD account to any SPN.

- OR-

Select **Trust this computer for delegation to specified services only** to add the MSSQLSvc SPN the list of specified services.

a Click **Add**.

b On the Add Services dialog, click **Users or Computers**.

  c Select the SQL service domain run-as account.

  - OR -

  Select the computer running the SQL server service (if the service is running as LocalSystem, NetworkService, or LocalService).

  The SPNs for the account are added to the list of available services.

4 Select the MSSQLSvc SPN with the matching TCP port for the SQL Server instance that hosts the MessageStats database and click **Add**.

5 Click **OK** to save.

Once you have completed steps 1 and 2, you must either wait for Active Directory replication or you can manually force replication between all the domain controllers.

3. Specify Windows authentication in the QMSReports.udl file.

Now you must edit QMSReports.udl file located in MessageStats installation folder and ensure that Windows NT integrated security is used to log onto the MesssageStats database.

To verify authentication type in the QMSReports.udl file

1 On the server on which the MessageStats reports are installed, navigate to the MessageStats installation folder. By default, the installation folder is located at:

  ▪ On 32-bit Windows computers: C:\Program Files\Quest\MessageStats

  ▪ On 64-bit Windows computers: C:\Program Files (x86)\Quest\MessageStats

2 Double-click the **QMSReports.udl** file.

3 Select the **Connection** tab.

4 Ensure that **Windows NT authentication** is selected.

4. Enable IIS Settings and Restart IIS.

For MessageStats report servers that are hosted by Windows 2012 and later, you must ensure specific IIS settings are enabled under Authentication and that the applicationHost.config file is updated. For all Windows server versions, you must reset IIS by performing an iisreset.

To enable the Authentication Settings for Windows

1 In the Internet Information Services (IIS) Manager, select **Sites | Default Web Site | MessageStats Reports**.

2 Double-click Authentication and ensure the following settings are enabled:

  ▪ ASP.NET Impersonation

  ▪ Windows Authentication

IMPORTANT: As of release 7.7, for Web Reports, Anonymous Authentication must be disabled.

To update the config file for Windows

1 Locate the following file in the %windir%\System32\inetsrv\config folder:

  applicationHost.config

2 Using a text editor such as Notepad, open the file and locate the <location> section that contains the following entry:

location path="Default Web Site/MessageStatsReports"

3   In the <authentication> section, find the entry for <windowsAuthentication> and look for the useAppPoolCredentials option:

- If the useAppPoolCredentials option is in the <windowsAuthentication> entry, change the value to true.

- If the useAppPoolCredentials option does not exist. add the option as follows: useAppPoolCredentials="true"

For example, the edited code might appear as follows:

```
<location path="Default Web Site/MessageStatsReports">
   <system.webServer>
     <security>
       <authentication>
         <anonymousAuthentication enabled="false" />
         <windowsAuthentication enabled="true" authPersistSingleRequest="false"
useKernelMode="true" useAppPoolCredentials="true">
         </windowsAuthentication>
       </authentication>
     </security>
```

Perform an iisreset on the MessageStats reports server to ensure that IIS uses the updated delegation and QMSReport.UDL settings.

Deploying MessageStats Across Forests

MessageStats can be used in cross-forest scenarios to gather information from an Exchange organization that resides in a different forest by using an alternate account from the external forest and establishing a trust relationship.

The following steps must be completed to use MessageStats to gather information from an Exchange organization in an external forest.

In the following scenario, Domain A has MessageStats installed. Domain B holds the alternate account you want to use to connect to the second organization.

- Domain A: MessageStats Console
- Domain B (represents the client forest): Client Source Exchange Organization

1. Establish the Trust Relationship

In order to use an alternate account for connecting to the Exchange organization, at a minimum a one-way Trust must be established.

1   In Domain A, create an Outgoing Trust with Domain B.

2   In Domain B, create an Incoming Trust with Domain A.

2. Create the MessageStats Service Account

To connect to the Exchange organization within Domain B, you must create an account within Domain B and grant the account the proper permissions.

To create the service account in Domain B

1 In Domain B, create a user account. For example, QMSTATS.

2 Grant the account membership in the Local Administrators group of each Exchange server in the Exchange organization.

3 Grant the account membership in the Exchange Organization Management security group in the Exchange organization

3. Configure Permissions, Set Credentials, and Create Gathering

Since the MessageStats is to be installed in Domain A (HP ECS), you must configure MessageStats to use alternate credentials for connecting to the Exchange organization in Domain B.

To grant the required memberships

1 Grant the Domain B\QMMSTATS account membership in the Local Administrators group on the MessageStats computer.

2 Grant the Domain B\QMMSTATS account membership in the MessageStats Admin Group on the SQL server that hosts the MessageStats database.

To configure the connection

1 Open the MessageStats console and right click on **Exchange Organizations**.

2 Select **Connect**.

3 Select the Exchange versions for which there are mailbox servers in the organization.

4 Specify the NetBIOS name of an Exchange server in the other organization and click **Next**.

- Exchange 2010: Specify a server with a CAS role. You must also specify a mailbox to which the account you are using to connect has full mailbox rights.

- Exchange 2013: Specify a server with a CAS role to provide proxy service to MAPI client. You must also specify a mailbox to which the account you are using to connect has full mailbox rights.

- Exchange 2016/2019: Specify a server with the Mailbox role.

5 Select **Specify Explicit Credentials** and enter the domain\username of the alternate account (such as DomainB\QMSTATS) and the password.

6 Complete the wizard. The organization should now appear in the console.

To configure the gathering

To run a gathering using the new connection to the Exchange organization in Domain B, perform the following steps:

1 Right-click on the Domain B Exchange Organization and select **Create Task**.

2 Click **Next** through the wizard to display the Task Credentials page.

3 Clear the **Use Default Configuration** check box at the bottom and specify the alternate account that you created to connect to the Exchange organization in Domain B (DomainB\QMSTATS)

4 Complete the wizard and run the gathering task.

Manually Creating the MessageStats Database in a Distributed Installation

In some distributed installation scenarios, due to security considerations, you might be unable to run the MessageStats installation wizard on the SQL server. In this case, you can use the MessageStats database creation script (MessageStatsDatabase.sql) to manually create the SQL database.

You must also manually create the necessary security groups to allow access to the MessageStats database.

To manually create the MessageStats database on both clustered and non-clustered SQL servers

1 Copy the MessageStats database creation script (MessageStatsDatabase.sql) from the SQL folder in the MessageStats Installation folder to your SQL server.

2 Open the SQL Server Management Studio.

3 Connect to the SQL Server that is going to host the MessageStats Database (for example, Server Type = Database Engine, Server Name = SQL Server Name and Authentication = either Windows Authentication or SQL Server Authentication).

4 Expand SQL Server Name and expand Databases.

5 Right-click **Databases** and select **New Database**.

6 Create an empty database called MessageStats.

7 Right-click the newly created MessageStats database and select **New Query**

8 After the Query window opens, select the **File | Open | File**.

9 Locate the MessageStatsDatabase.sql script and open it.

10 Provide either Windows Authentication or SQL Server Authentication and click **Connect**.

   To use Windows authentication for an installation in which the database and reports are on different servers, Kerberos delegation is required. For information about configuring Kerberos delegation, see Deploying a Distributed Installation Using Windows Authentication on page 18.

11 Select the **Query Menu** and select **Execute**.

This will create all the MessageStats database tables and views.

Creating the required security groups and assigning permissions

Security groups are needed to assign rights to the database. You can use either local or global groups. The groups must be named MessageStats Admin and MessageStats Web.

If the MessageStats database resides on a SQL cluster, you must create the groups in Active Directory.

To create the required security groups

1 Launch Computer Management and expand System Tools | Local Users and Groups| Groups to create local groups

   - OR -

   Open Active Directory Users and Computers to create global groups.

2 Create two groups that are named as follows:

   ▪ MessageStats Admin
     (for accounts that perform gatherings and that administer the MessageStats console).

- MessageStats Web
  (for accounts that view the reports)

3 Ensure that the MessageStats service account is added to the MessageStats Admin group.

Typically the Everyone group is added to the MessageStats Web group if you want to ensure all your users can view the reports.

To assign the MessageStats Admin group to the MessageStats database

1 Open the SQL Management Studio and expand Security | Logins.

2 Right-click **Logins** and choose **New Login**.

3 Click **Search** and locate the MessageStats Admin group. Click **OK**.

4 Select **Windows Authentication**.

5 For the Default Database, select **MessageStats** from the drop-down list.

6 In the left column, click **User Mapping**.

7 For the Users Mapped to this Login section, check the box beside the MessageStats database.

8 Under the Database Role Memberships for section, select the following:

- Public
- MessageStats_Admin
- db_securityadmin
- db_ddladmin.

9 Click **OK**.

To assign the MessageStats Web group to the MessageStats database

1 Open the SQL Management Studio and expand Security | Logins.

2 Right-click **Logins** and select **New Login**.

3 Click **Search** and find the MessageStats Web group. Click **OK**.

4 Select **Windows Authentication**.

5 For the Default Database, select **MessageStats** from the drop-down list.

6 In the left column, click **User Mapping**.

7 For the Users Mapped to this Login section, check the box beside the MessageStats database.

8 Under the Database Role Memberships for section, select the following:

- Public
- MessageStats_web.

9 Click **OK**.

About us

Quest provides software solutions for the rapidly-changing world of enterprise IT. We help simplify the challenges caused by data explosion, cloud expansion, hybrid datacenters, security threats, and regulatory requirements. We are a global provider to 130,000 companies across 100 countries, including 95% of the Fortune 500 and 90% of the Global 1000. Since 1987, we have built a portfolio of solutions that now includes database management, data protection, identity and access management, Microsoft platform management, and unified endpoint management. With Quest, organizations spend less time on IT administration and more time on business innovation. For more information, visit www.quest.com.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request.

- View Knowledge Base articles.

- Sign up for product notifications.

- Download software and technical documentation.

- View how-to-videos.

- Engage in community discussions.

- Chat with support engineers online.

- View services to assist you with your product.