SQL Server Distributed Replay

SQL Server 2012 Books Online

Quick Reference

Microsoft

SQL Server Distributed Replay

SQL Server 2012 Books Online

Summary: The Microsoft SQL Server Distributed Replay feature helps you assess the impact of future SQL Server upgrades. You can also use it to help assess the impact of hardware and operating system upgrades, and SQL Server tuning.

Category: Quick Reference
Applies to: SQL Server 2012
Source: SQL Server Books Online (link to source content)
E-book publication date: June 2012

Copyright © 2012 by Microsoft Corporation
All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Microsoft and the trademarks listed at http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Contents

SQL Server Distributed Replay

The Microsoft SQL Server Distributed Replay feature helps you assess the impact of future SQL Server upgrades. You can also use it to help assess the impact of hardware and operating system upgrades, and SQL Server tuning.

Benefits of Distributed Replay

Similar to SQL Server Profiler, you can use Distributed Replay to replay a captured trace against an upgraded test environment. Unlike SQL Server Profiler, Distributed Replay is not limited to replaying the workload from a single computer.

Distributed Replay offers a more scalable solution than SQL Server Profiler. With Distributed Replay, you can replay a workload from multiple computers and better simulate a mission-critical workload.

The Microsoft SQL Server Distributed Replay feature can use multiple computers to replay trace data and simulate a mission-critical workload. Use Distributed Replay for application compatibility testing, performance testing, or capacity planning.

When to Use Distributed Relay

SQL Server Profiler and Distributed Replay provide some overlap in functionality.

You may use SQL Server Profiler to replay a captured trace against an upgraded test environment. You can also analyze the replay results to look for potential functional and performance incompatibilities. However, SQL Server Profiler can only replay a workload from a single computer. When replaying an intensive OLTP application that has many active concurrent connections or high throughput, SQL Server Profiler can become a resource bottleneck.

Distributed Replay offers a more scalable solution than SQL Server Profiler. Use Distributed Replay to replay a workload from multiple computers and better simulate a mission-critical workload.

The following table describes when to use each tool.

Tool	**Use When…**
SQL Server Profiler	• You want to use the conventional replay mechanism on a single computer. In particular, you need line-by-line debugging capabilities, such as the **Step**, **Run to Cursor**, and **Toggle**

Tool	Use When…
	Breakpoint commands. - You want to replay an Analysis Services trace.
Distributed Replay	- You want to evaluate application compatibility. For example, you want to test SQL Server and operating system upgrade scenarios, hardware upgrades, or index tuning. - The concurrency in the captured trace is so high that a single replay client cannot sufficiently simulate it.

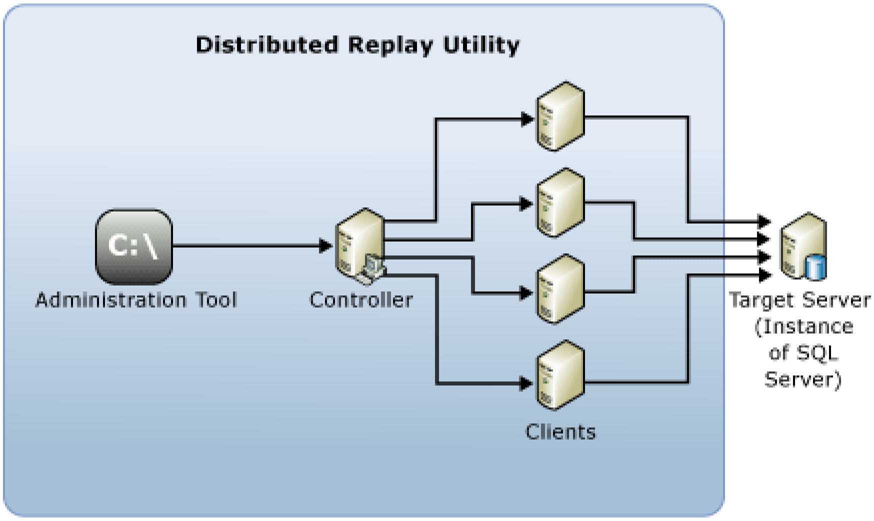
Distributed Replay Concepts

The following components make up the Distributed Replay environment:

- **Distributed Replay administration tool**: A console application, **DReplay.exe**, used to communicate with the distributed replay controller. Use the administration tool to control the distributed replay.
- **Distributed Replay controller**: A computer running the Windows service named SQL Server Distributed Replay controller. The Distributed Replay controller orchestrates the actions of the distributed replay clients. There can only be one controller instance in each Distributed Replay environment.
- **Distributed Replay clients**: One or more computers (physical or virtual) running the Windows service named SQL Server Distributed Replay client. The Distributed Replay clients work together to simulate workloads against an instance of SQL Server. There can be one or more clients in each Distributed Replay environment.
- **Target server**: An instance of SQL Server that the Distributed Replay clients can use to replay trace data. We recommend that the target server be located in a test environment.

The Distributed Replay administration tool, controller, and client can be installed on different computers or the same computer. There can be only one instance of the Distributed Replay controller or client service that is running on the same computer.

The following figure shows the SQL Server Distributed Replay physical architecture:


Distributed Replay Utility
C:\
Administration Tool
Controller
Clients
Target Server (Instance of SQL Server)

Distributed Replay Tasks

Task Description	Topic
Describes how to configure Distributed Replay.	Configure Distributed Replay
Describes how to prepare the input trace data.	Prepare the Input Trace Data
Describes how to replay trace data.	Replay Trace Data
Describes how to review the Distributed Replay trace data results.	Review the Replay Results
Describes how to use the administration tool to initiate, monitor, and cancel operations on the controller.	Administration Tool Command-line Options (Distributed Replay Utility)

Configure Distributed Replay

The Microsoft SQL Server Distributed Replay configuration details are specified in XML files on the Distributed Replay controller, clients, and where the administration tool is installed. These files include the following:

- Controller configuration file

- Client configuration file
- Preprocess configuration file
- Replay configuration file

Controller Configuration File: DReplayController.config

When the SQL Server Distributed Replay controller service starts, it loads the logging level from the controller configuration file, `DReplayController.config`. This file is located in the folder where you installed the Distributed Replay controller service:

<controller installation path>\DReplayController.config

The logging level specified by the controller configuration file includes the following:

Setting	XML Element	Description	Allowed Values	Required
Logging level	**`<LoggingLevel>`**	Specifies the logging level for the controller service.	`INFORMATION` | `WARNING` | `CRITICAL`	No. By default, the value is `CRITICAL`.

Example

This example shows a controller configuration file that has been modified to suppress `INFORMATION` and `WARNING` log entries.

```
<?xml version='1.0'?>

<Options>

<LoggingLevel>CRITICAL</LoggingLevel>

</Options>
```

Client Configuration File: DReplayClient.config

When the SQL Server Distributed Replay client service starts, it loads configuration settings from the client configuration file, `DReplayClient.config`. This file is located on each client, in the folder where you installed the Distributed Replay client service:

<client installation path>\DReplayClient.config

The settings specified by the client configuration file include the following:

Setting	XML Element	Description	Allowed Values	Required
Controller	**`<Controller>`**	Specifies the computer	You can use "`localhost`" or	No. By default, the client tries

Setting	XML Element	Description	Allowed Values	Required
		name of the controller. The client will attempt to register with the Distributed Replay environment by contacting the controller.	"`.`" to refer to the local computer.	to register with the controller instance that is running locally ("`.`"), if it exists.
Client working directory	**`<WorkingDirectory>`**	Is the local path on the client where the dispatch files are saved. The files in this directory are overwritten on the next replay.	A full directory name, starting with the drive letter.	No. If no value is specified, the dispatch files will be saved in the same location as the default client configuration file. If a value is specified and that folder does not exist on the client, the client service will not start.
Client result directory	**`<ResultDirectory>`**	Is the local path on the client where the result trace file from the replay activity (for the client) is saved. The files in this directory are overwritten on	A full directory name, starting with the drive letter.	No. If no value is specified, the result trace file will be saved in the same location as the default client configuration file. If a value is specified and that folder does not exist on the client,

Setting	XML Element	Description	Allowed Values	Required
		the next replay.		the client service will not start.
Logging level	**`<LoggingLevel>`**	Is the logging level for the client service.	`INFORMATION` | `WARNING` | `CRITICAL`	No. By default, the value is `CRITICAL`.

Example

This example shows a client configuration file that has been modified to specify that the controller service is running on a different computer, a computer named `Controller1`. The `WorkingDirectory` and `ResultDirectory` elements have been configured to use folders `c:\ClientWorkingDir` and `c:\ResultTraceDir`, respectively. The logging level has been changed from the default value to suppress `INFORMATION` and `WARNING` log entries.

```
<?xml version='1.0'?>
<Options>
    <Controller>Controller1</Controller>
    <WorkingDirectory>c:\ClientWorkingDir</WorkingDirectory>
    <ResultDirectory>c:\ResultTraceDir</ResultDirectory>
    <LoggingLevel>CRITICAL</LoggingLevel>
</Options>
```

Preprocess Configuration File: DReplay.exe.preprocess.config

When you use the administration tool to initiate the preprocess stage, the administration tool loads the preprocess settings from the preprocess configuration file, `DReplay.exe.preprocess.config`.

Use the default configuration file or use the administration tool **-c** parameter to specify the location of a modified preprocess configuration file. For more information about using the preprocess option of the administration tool, see preprocess Option (Distributed Replay Administration Tool).

The default preprocess configuration file is located in the folder where you installed the administration tool:

<administration tool installation path>\DReplayAdmin\DReplay.exe.preprocess.config

The preprocess configuration settings are specified in XML elements that are children of the **`<PreprocessModifiers>`** element in the preprocess configuration file. These settings include the following:

Setting	XML Element	Description	Allowed Values	Required
Include system session activities	**`<IncSystemSession>`**	Indicates whether system session activities during the capture will be included during replay.	`Yes` | `No`	No. By default, the value is `No`.
Maximum idle time	**`<MaxIdleTime>`**	Caps the idle time to an absolute number (in seconds).	An integer that is >= -1. • `-1` indicates no change from the original value in the original trace file. • `0` indicates that there is some activity going on at any given point in time.	No. By default, the value is `-1`.

Example

The default preprocess configuration file:

```
<?xml version='1.0'?>
```

```
<Options>
    <PreprocessModifiers>
        <IncSystemSession>No</IncSystemSession>
        <MaxIdleTime>-1</MaxIdleTime>
    </PreprocessModifiers>
</Options>
```

Replay Configuration File: DReplay.exe.replay.config

When you use the administration tool to initiate the event replay stage, the administration tool loads the replay settings from the replay configuration file, `DReplay.exe.replay.config`.

Use the default configuration file or use the administration tool **-c** parameter to specify the location of a modified replay configuration file. For more information about using the replay option of the administration tool, see replay Option (Distributed Replay Administration Tool).

The default replay configuration file is located in the folder where you installed the administration tool:

<administration tool installation path>\DReplayAdmin\DReplay.exe.replay.config

The replay configuration settings are specified in XML elements that are children of the **`<ReplayOptions>`** and **`<OutputOptions>`** elements of the replay configuration file.

<ReplayOptions> Element

The settings specified by the replay configuration file in the **`<ReplayOptions>`** element include the following:

Setting	XML Element	Description	Allowed Values	Required
Target instance of SQL Server (the test server)	**`<Server>`**	Specifies the name of the server and instance of SQL Server to connect to.	*server_name*[\instance_name] You cannot use "`localhost`" or "`.`" to represent the local host.	No, if the server name is already specified by using the **-s** target server parameter with the **replay** option of

Setting	XML Element	Description	Allowed Values	Required
				the administration tool.
Sequencing mode	**<SequencingMode>**	Specifies the mode that is used for event scheduling. For more information, see SQL Server Distributed Replay.	synchronization | stress	No. By default, the value is stress.
Stress scale granularity	**<StressScaleGranularity>**	Specifies whether all connections on the Service Profile Identifier (SPID) should be scaled together (SPID) or independently (Connection) under stress mode.	SPID | Connection	Yes. By default, the value is SPID.
Connect time scale	**<ConnectTimeScale>**	Is used to scale the connect time in stress mode. For more information, see SQL Server	An integer between 1 and 100.	No. By default, the value is 100.

Setting	XML Element	Description	Allowed Values	Required
		Distributed Replay.		
Think time scale	**<ThinkTimeScale>**	Is used to scale think time in stress mode. For more information, see SQL Server Distributed Replay.	An integer between `0` and `100`.	No. By default, the value is `100`.
Use connection pooling	**<UseConnectionPooling>**	Specifies whether connection pooling will be enabled on each Distributed Replay client.	Yes | No	Yes. By default, the value is `Yes`.
Health monitor interval	**<HealthmonInterval>**	Indicates how often to run the health monitor (in seconds). This value is only used in synchronization mode.	Integer >= 1 (`-1` to disable)	No. By default, the value is `60`.
Query time-out	**<QueryTimeout>**	Specifies the query time-out value, in seconds. This value is only effective until the first	Integer >= 1 (`-1` to disable)	No. By default, the value is `3600`.

Setting	XML Element	Description	Allowed Values	Required
		row has been returned.		
Threads per client	**<ThreadsPerClient>**	Specifies the number of replay threads to use for each replay client.	An integer between 1 and 512.	No. If not specified, Distributed Replay will use a value of 255.

<OutputOptions> Element

The settings specified by the replay configuration file in the **<OutputOptions>** element include the following:

Setting	XML Element	Description	Allowed Values	Required
Record row count	**<RecordRowCount>**	Indicates whether the row count should be recorded for each result set.	Yes | No	No. By default, the value is Yes.
Record result set	**<RecordResultSet>**	Indicates whether the content of all result sets should be recorded.	Yes | No	No. By default, the value is No.

Example

The default replay configuration file:

```
<?xml version='1.0'?>

<Options>

    <ReplayOptions>

        <Server></Server>
        <SequencingMode>stress</SequencingMode>
        <ConnectTimeScale></ConnectTimeScale>
        <ThinkTimeScale></ThinkTimeScale>
        <HealthmonInterval>60</HealthmonInterval>
        <QueryTimeout>3600</QueryTimeout>
        <ThreadsPerClient></ThreadsPerClient>
    </ReplayOptions>
    <OutputOptions>
        <ResultTrace>
            <RecordRowCount>Yes</RecordRowCount>
            <RecordResultSet>No</RecordResultSet>
        </ResultTrace>
    </OutputOptions>
</Options>
```

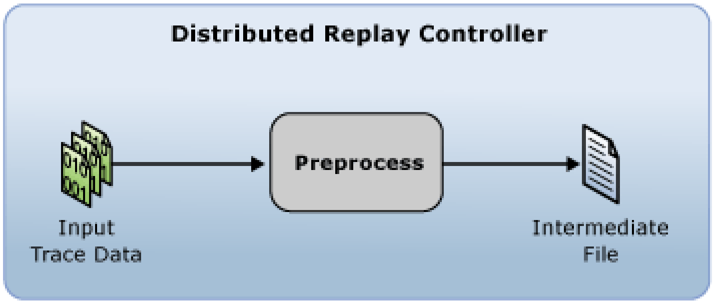
See Also

Administration Tool Command-line Options (Distributed Replay)

SQL Server Distributed Replay

Prepare the Input Trace Data

Before you can start a distributed replay with the Microsoft SQL Server Distributed Replay feature, you must prepare the input trace data by initiating the preprocess stage from the distributed replay administration tool. In the preprocess stage, the distributed replay controller processes the trace data and generates an intermediate file:



For more information about the preprocess stage, see SQL Server Distributed Replay.

Note

The input trace data must be captured in a version of SQL Server that is compatible with Distributed Replay. The input trace data must also be compatible with the target server that you want to replay the trace data against. For more information about version requirements, see Distributed Replay Requirements.

Procedures

▶ To prepare the input trace data

1. **(Optional) Modify preprocess configuration settings**: If you want to modify the preprocess configuration settings, such as whether to filter system sessions or to configure the maximum idle time, you must modify the **`<PreprocessModifiers>`** element of the XML-based preprocess configuration file, `DReplay.exe.preprocess.config`. If you modify the preprocess configuration file, we recommend that you modify a copy rather than the original. To modify settings, follow these steps:

   a. Make a copy of the default preprocess configuration file, `DReplay.exe.preprocess.config`, and rename the new file. The default preprocess configuration file is located in the administration tool installation folder.

   b. Modify the preprocess configuration settings in the new configuration file.

   c. When initiating the preprocess stage (the next step), use the config_file parameter of the **preprocess** option to specify the location of the modified configuration file.

   For more information about the preprocess configuration file, see Distributed Replay Configuration Files.

2. **Initiate the preprocess stage**: To prepare the input trace data, you must run the administration tool with the **preprocess** option. For more information, see dreplay preprocess Command.

   a. Open the Windows Command Prompt utility (**CMD.exe**), and navigate to the installation location of the Distributed Replay administration tool (**DReplay.exe**).

   b. (Optional) Use the controller parameter, **-m**, to specify the controller, if the controller service is running on a computer different from the administration tool.

   c. Use the input_trace_file parameter, **-i**, to specify the location and name of the input trace files.

   d. Use the controller_working_directory parameter, **-d**, to specify where the intermediate file should be saved on the controller.

   e. (Optional) Use the config_file parameter, **-c**, to specify location of the preprocess configuration file. Use this parameter to point to the new configuration file if you have modified a copy of the default preprocess configuration file.

f. (Optional) Use the status_interval parameter, **-f**, to specify if you want the administration tool to display status messages at a frequency different than 30 seconds.

For example, initiating the preprocess stage on the same computer as the controller service, for a trace file located at `c:\trace1.trc`, a controller working directory located at `c:\WorkingDir` , and a status message displayed at the default value of 30 seconds, requires the syntax: `dreplay preprocess -i c:\trace1.trc -d c:\WorkingDir`

3. After the preprocess stage is complete, the intermediate file is stored in the controller working directory. To initiate the event replay stage, you must run the administration tool with the **replay** option. For more information, see How to: Replay the Trace.

See Also

SQL Server Distributed Replay

Distributed Replay Requirements

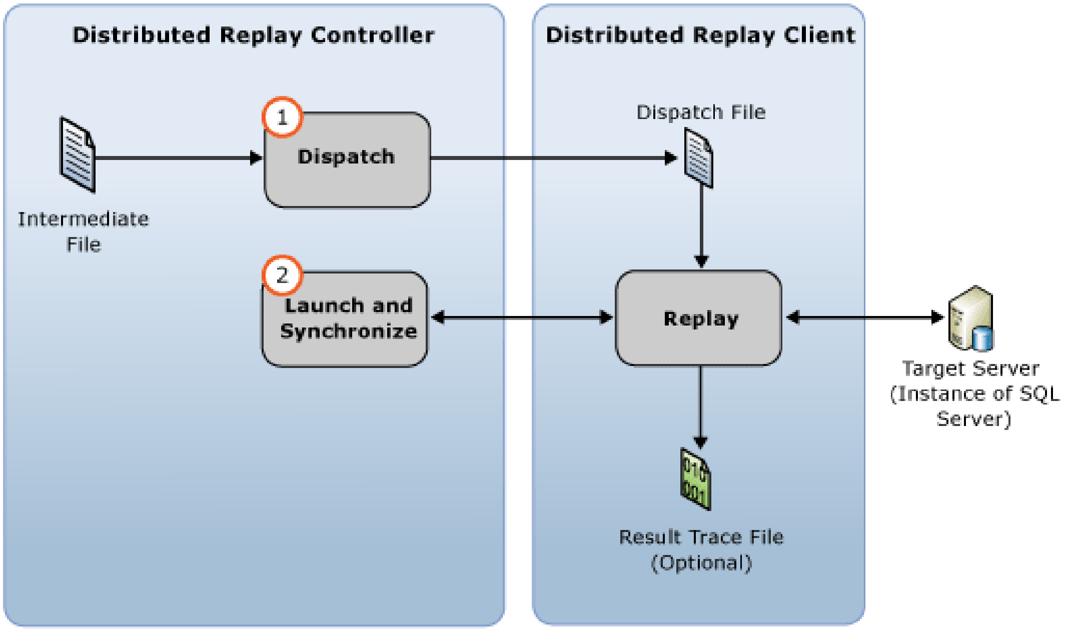
Administration Tool Command-line Options (Distributed Replay)

Distributed Replay Configuration Files

Replay Trace Data

You can start a distributed replay with the Microsoft SQL Server Distributed Replay feature after you have prepared the input trace data. For more information, see How to: Prepare the Input Trace Data.

Use the administration tool **replay** option to initiate the event replay stage of the distributed replay. This stage consists of two parts: the trace data dispatch and the starting and synchronizing of the distributed replay.


Distributed Replay Controller
Distributed Replay Client
1
Dispatch
Intermediate File
2
Launch and Synchronize
Dispatch File
Replay
Target Server (Instance of SQL Server)
Result Trace File (Optional)

You can replay trace data in one of two sequencing modes: stress mode or synchronization mode. The default behavior is to replay trace data in stress mode. For more information about the event replay stage and sequencing modes, see SQL Server Distributed Replay

Note
The input trace data must be captured in a version of SQL Server that is compatible with Distributed Replay. The input trace data must also be compatible with the target server that you want to replay the trace data against. For more information about version requirements, see Distributed Replay Requirements.

Procedures

To replay the trace

1. **(Optional) Modify replay configuration settings**: If you want to modify the replay configuration settings, such as the sequencing mode and various scaling values, you must modify the **`<ReplayOptions>`** element of the XML-based replay configuration file `DReplay.exe.replay.config`. You can also modify the **`<OutputOptions>`** element to specify output settings, such as whether to record the row count. If you modify the replay configuration file, we recommend that you modify a copy rather than the original. To modify settings, follow these steps:

   a. Make a copy of the default replay configuration file, `DReplay.exe.replay.config`, and rename the new file. The default replay configuration file is located in the administration tool installation folder.

b. Modify the replay configuration settings in the new configuration file.

c. When initiating the event replay stage (the next step), use the config_file parameter of the **replay** option to specify the location of the modified configuration file.

For more information about the replay configuration file, see [Distributed Replay Configuration Files].

2. **Initiate the event replay stage**: To start the distributed replay, you must run the administration tool with the **replay** option. For more information, see [dreplay replay Command].

a. Open the Windows Command Prompt utility (**CMD.exe**), and navigate to the installation location of the Distributed Replay administration tool (**DReplay.exe**).

b. (Optional) Use the controller parameter, **-m**, to specify the controller, if the controller service is running on a computer different from the administration tool.

c. Use the controller_working_directory parameter, **-d**, to specify where the intermediate file was saved on the controller during the preprocess stage.

d. (Optional) Use the **-o** parameter to capture the replay activity in a result trace file on each client.

e. (Optional) Use the target_server parameter, **-s**, to specify the instance of SQL Server where the distributed replay clients should replay the trace workload. This parameter is not required if you used the **`<Server>`** element to specify the target server in the **`<ReplayOptions>`** element of the replay configuration file.

f. Use the clients parameter, **-w**, to specify the distributed replay clients that should participate in the replay. List the client computer names, separated by commas. Note: IP addresses are not allowed.

g. (Optional) Use the config_file parameter, **-c**, to specify location of the replay configuration file. Use this parameter to point to the new configuration file if you have modified a copy of the default replay configuration file.

h. (Optional) Use the status_interval parameter, **-f**, to specify if you want the administration tool to display status messages at a frequency other than 30 seconds.

For example, the following syntax initiates the replay stage on the same computer as the controller service, uses a controller working directory located at `c:\WorkingDir`, captures the replay activity on each participating client, uses clients `client1` and `client2` to perform the replay, and obtains the remaining replay configuration settings from a modified replay configuration file located at `c:\modifiedreplay.config`:

```
dreplay replay -d c:\WorkingDir -o -w client1,client2 -c
```

```
c:\modifiedreplay.config
```

3. When the distributed replay has finished, the administration tool returns summary information. If you specified the **-o** option, the replay activity has been saved in result trace files on each client. For more information about the result trace files, see How to: Review the Result Trace.

See Also

Distributed Replay Requirements

Administration Tool Command-line Options (Distributed Replay)

Distributed Replay Configuration Files

Review the Replay Results

After the Microsoft SQL Server Distributed Replay feature completes a distributed replay, the replay activity for each client can be captured and saved in result trace files on each client. In order to capture this activity, you must use the **-o** parameter when you run the administration tool with the **replay** option. For more information about the replay option, see replay Option (Distributed Replay Administration Tool).

The location of where the result trace files are stored is specified by the **`<ResultDirectory>`** XML element in the client configuration file, `DReplayClient.xml`, located on each client. The trace files in the client result directory are overwritten on each replay.

To specify what kind of output should be captured in the result trace files, modify the replay configuration file, `DReplay.exe.replay.config`. You can use the **`<OutputOptions>`** XML element to specify whether the row count or result set contents should be recorded.

For more information about these configuration settings, see Distributed Replay Configuration Files.

Event Classes Captured in Result Trace Files

The following table lists all of the event classes that are captured in the result trace data.

Category	EventClass Name	Capture Frequency	Point of Capture
Replayable Events	Audit Login	One time for each Audit Login event in the original trace data	Upon successful completion or failure of the event

Category	EventClass Name	Capture Frequency	Point of Capture
	Audit Logout	One time for each Audit Logout event in the original trace data	Upon successful completion or failure of the event
	SQL:BatchCompleted	One time for each SQL:BatchStarting event in the original trace data	Upon successful completion or failure of the event
	RPC:Completed	One time for each RPC:Starting event in the original trace data	Upon successful completion or failure of the event
Statistics and Results	Replay Settings Event	One time	First event of the result trace
	Replay Statistics Event	One time	Last event of the result trace
	Replay Result Set Event	One time for each SQL:BatchStarting and RPC:Starting event. Only captured if the value of the **`<RecordResultSet>`** option in the replay configuration file was set to `Yes`.	
	Replay Result Row Event	One time for each row in the result set for SQL:BatchStarting and RPC:Starting events. Only captured if the value of the **`<RecordResultSet>`** option in the replay configuration file was set to `Yes`.	
Errors and Warnings	Replay Internal Error	One time for each internal error	Upon internal error condition

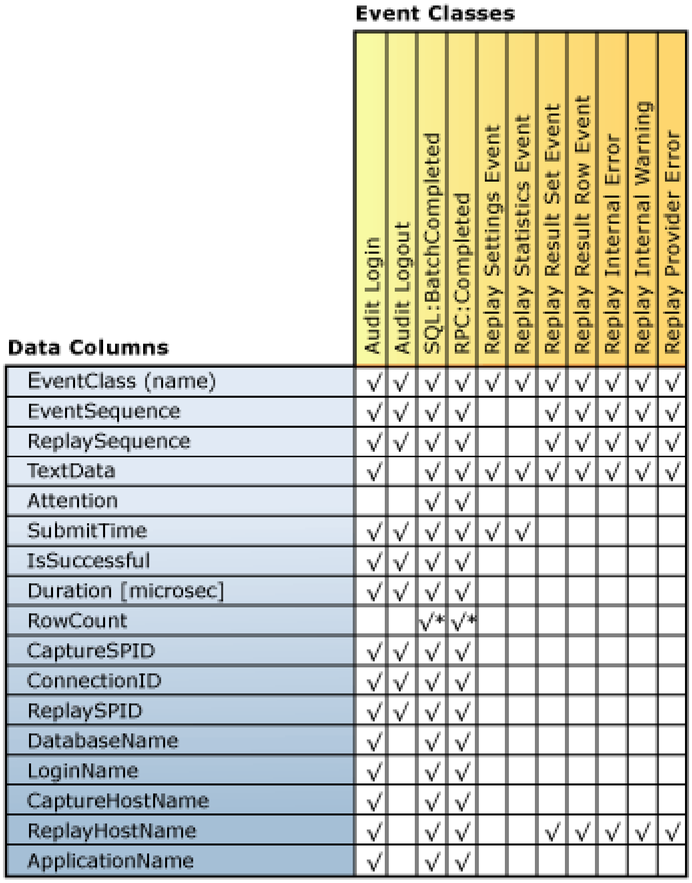
Category	EventClass Name	Capture Frequency	Point of Capture
	Replay Provider Error	One time for each provider error	Upon provider error condition

Note the following:

- For each event that is successfully replayed on the target server, there is one corresponding output event class.
- For each event failure or cancellation, there may be multiple errors that are generated.

Event Class Column Mapping

The following figure lists which columns of the result trace are available for each type of event class that is captured during the replay.



Event Classes

Data Columns	Audit Login	Audit Logout	SQL:BatchCompleted	RPC:Completed	Replay Settings Event	Replay Statistics Event	Replay Result Set Event	Replay Result Row Event	Replay Internal Error	Replay Internal Warning	Replay Provider Error
EventClass (name)	√	√	√	√	√	√	√	√	√	√	√
EventSequence	√	√	√	√			√	√	√	√	√
ReplaySequence	√	√	√	√			√	√	√	√	√
TextData	√		√	√	√	√	√	√	√	√	√
Attention			√	√							
SubmitTime	√	√	√	√	√	√					
IsSuccessful	√	√	√	√							
Duration [microsec]	√	√	√	√							
RowCount			√*	√*							
CaptureSPID	√	√	√	√							
ConnectionID	√	√	√	√							
ReplaySPID	√	√	√	√							
DatabaseName	√		√	√							
LoginName	√		√	√							
CaptureHostName	√		√	√							
ReplayHostName	√		√	√			√	√	√	√	√
ApplicationName	√		√	√							

*Optional data column

Column Descriptions for Result Trace

The following table describes the columns of the result trace data.

Data Column Name	Data Type	Description	Column ID
EventClass	**nvarchar**	The name of the event class.	1
EventSequence	**bigint**	For provider errors, and internal errors and warnings, this is the capture event sequence that corresponds to the error or warning. For all other event classes, this is the sequence of the event in the original trace data.	2
ReplaySequence	**bigint**	For provider errors, and internal errors and warnings, this is the replay event sequence that corresponds to the error or warning. For all other event classes, this is the sequence of the event that is assigned during replay.	3
TextData	**ntext**	The content of TextData depends on the EventClass. For Audit Login and ExistingConnection, this is the set options for the connection. For SQL:BatchStarting, this is the body of the batch request. For RPC:Starting, this is the stored procedure that was called. For Replay Settings Event, this column contains the settings that are defined in the replay configuration file. For Replay Statistics Event, this contains the following information: • The replay target SQL server	4

Data Column Name	Data Type	Description	Column ID
		• Total number of replayable events • The number of provider errors • The number of internal errors • Internal warnings • Total number of errors • Overall pass rate • The replay time (HH:MM:SS:MMM) For Replay Result Set Event, this shows the list of return result column headers. For Replay Result Row Event, this shows the return value of all columns for that row. For Replay Internal Warning and Replay Provider Error, this column contains the provider warnings or errors.	
Attention	**bigint**	The attention duration (in microseconds) for the event. This is calculated from the Attention event from the capture trace. If there was no query time-out specified for the event, this column is not populated (null).	5
SubmitTime	**datetime**	The time when the event was submitted to SQL Server.	6
IsSuccessful	**int**	A Boolean flag that indicates whether a particular event was successfully executed, and that result sets were returned to the client-side. An event that generates a warning (such as when an event is cancelled because of Attention or a user-specified time-out) is	7

Data Column Name	Data Type	Description	Column ID
		considered successful. IsSuccessful can be one of the following: • 1 = successful • 0 = failed	
Duration [microsec]	**bigint**	Response time duration (in microseconds) for the event. The measurement starts when the logon/log off/RPC/Language event was submitted to SQL Server. If the event succeeds, the measurement ends when the complete result set has been consumed. If the event does not succeed, the measurement ends at the time of event failure or cancellation.	8
RowCount	**bigint**	Populated depending on the value of **`<RecordRowCount>`** in the replay configuration file: • If **`<RecordRowCount>`** equals Yes, this cell contains the number of rows in the result set that are returned by SQL Server. • If **`<RecordRowCount>`** equals No, this cell is not populated (null).	9
CaptureSPID	**int**	The ID of the capture session for the event.	10
ConnectionID	**int**	The ID of the capture connection for the event.	11
ReplaySPID	**int**	The ID of the replay session for the event.	12
DatabaseName	**nvarchar**	The name of the database in which the user statement is	13

Data Column Name	Data Type	Description	Column ID
		running.	
LoginName	**nvarchar**	The user login name. This can be either a SQL Server security login or the Microsoft Windows login credentials, in the format *domain_name**user_name*.	14
CaptureHostName	**nvarchar**	The name of the computer on which the client service is running during capture.	15
ReplayHostName	**nvarchar**	The name of the computer on which the client is running during replay.	16
ApplicationName	**nvarchar**	The name of the client application that created the SQL Server connection during capture.	17

See Also

SQL Server Distributed Replay

Distributed Replay Requirements

Administration Tool Command-line Options (Distributed Replay)

Distributed Replay Configuration Files

Distributed Replay Security

Before you install and use the Microsoft SQL Server Distributed Replay feature, you should review the important security information in this topic. This topic describes the post-installation security configuration steps that are required before you can use Distributed Replay. This topic also describes important considerations with regard to data protection and important removal steps.

User and Service Accounts

The following table describes the accounts that are used for Distributed Replay. After the Distributed Replay installation, you must assign the security principals that the controller and client service accounts will run as. Therefore, we recommend that you configure the corresponding domain user accounts before you install the Distributed Replay features.

User Account	Requirements
SQL Server Distributed Replay controller service account	Can be a domain user account or local user account. If you use a local user account, the administration tool, controller, and client must all be running on the same computer. **noteDXDOC112778PADS Security Note** We recommend that the account is not a member of the local Administrators group in Windows.
SQL Server Distributed Replay client service account	Can be a domain user account or local user account. If you use a local user account, the controller, client, and target SQL Server must all be running on the same computer. **noteDXDOC112778PADS Security Note** We recommend that the account is not a member of the local Administrators group in Windows.
Interactive user account that is used to run the Distributed Replay administration tool	Can be either a local user or a domain user account. To use a local user account, the administration tool and controller must be running on the same computer.

Important: When you configure Distributed Replay controller, you can specify one or more user accounts that will be used to run the Distributed Replay client services. The following is the list of supported accounts:

- Domain user account
- User created local user account
- Administrator
- Virtual account and MSA (Managed Service Account)
- Network Services, Local Services, and System

Group accounts (local or domain) and other built-in accounts (like Everyone) are not accepted.

To set the service accounts or their passwords after you install Distributed Replay, you can use the Windows Services tool. To change the service accounts associated with the Distributed Replay controller or client services, follow these steps:

1. Do either of the following, depending on the operating system:
   - Click **Start**, type **services.msc** in the **Search** box, and then press ENTER.
   - Click **Start**, click **Run**, type **services.msc**, and then press ENTER.
2. In the **Services** dialog box, right-click the service that you want to configure, and then click **Properties**.
3. On the **Log On** tab, click **This account**.
4. Configure the user account that you want to use.

File and Folder Permissions

After the service accounts have been specified, you must grant the necessary file and folder permissions to those service accounts. Configure file and folder permissions according to the following table:

Account	**Folder Permissions**
SQL Server Distributed Replay controller service account	• `<Controller_Installation_Path>\DReplayController` (Read, Write, Delete) • `DReplayServer.xml` file (Read, Write)
SQL Server Distributed Replay client service account	• `<Client_Installation_Path>\DReplayClient` (Read, Write, Delete) • `DReplayClient.xml` file (Read, Write) • The working and result directories, as specified in the client configuration file by the `WorkingDirectory` and `ResultDirectory` elements, respectively. (Read, Write)

DCOM Permissions

DCOM is used for remote procedure call (RPC) communication between the controller and the administration tool, and between the controller and all clients. You must configure computer-wide and application-specific DCOM permissions on the controller after the Distributed Replay features have been installed.

To configure the controller DCOM permissions, follow these steps:

1. **Open dcomcnfg.exe, the Component Services snap-in**: This is the tool that is used to configure DCOM permissions.
   a. On the controller computer, click **Start**.

   b. Type **dcomcnfg.exe** in the **Search** box.

   c. Press ENTER.

2. **Configure computer-wide DCOM permissions**: Grant the corresponding computer-wide DCOM permissions for each account listed in the following table. For more information about how to set computer-wide permissions, see Checklist: Manage DCOM Applications.

3. **Configure application-specific DCOM permissions**: Grant the corresponding application-specific DCOM permissions for each account listed in the following table. The DCOM application name for the controller service is **ProfilerServer**. For more information about how to set application-specific permissions, see Checklist: Manage DCOM Applications.

The following table describes which DCOM permissions are required for the administration tool interactive user account and the client service accounts:

Feature	Account	Required DCOM Permissions on Controller
Distributed Replay administration tool	The interactive user account	Local Access Remote Access Local Launch Remote Launch Local Activation Remote Activation
Distributed Replay client	SQL Server Distributed Replay client service account	Local Access Remote Access Local Launch Remote Launch Local Activation Remote Activation

Important

To help protect against malicious queries or denial of service attacks, make sure that you only use a trusted user account for the client service account. This account will be able to connect and replay workloads against the target instance of SQL Server.

SQL Server Permissions

The SQL Server Distributed Replay client service accounts are used to connect to the workload's target instance of SQL Server. Only Windows Authentication mode is supported for these connections.

After you install the SQL Server Distributed Replay client service on a set of computers, the security principal used for those service accounts must be granted the sysadmin server role on the instance of SQL Server that you intend to replay the trace workload against. This step is not performed automatically during Distributed Replay Setup.

Data Protection

In the Distributed Replay environment, the following user accounts are granted full access to the target server instance of SQL Server, the input trace data and result trace files:

- The interactive user account that is used to run the administration tool.
- The controller service account.
- The client service account.
- Members of the local Administrators group on the controller.
- Members of the local Administrators group on the clients.

> **Important**
> These accounts have full access to any personally identifiable information (PII) or sensitive information that is contained in the trace, intermediate, dispatch, or SQL Server data files that were used by Distributed Replay.

We recommend that you take the following security precautions:

- Store the input trace data, output trace results, and database files in a location that uses the NTFS file system (NTFS), and apply the appropriate access control lists (ACLs). If it is needed, encrypt the data that is stored on the SQL Server computer. Be aware that SQL Server Profiler does not apply ACLs to the trace files, or perform any kind of data masking or obfuscation. You should delete these files quickly after use.
- Apply the appropriate ACLs and retention policy to all intermediate and dispatch files that are generated by Distributed Replay.
- Use Secure Sockets Layer (SSL) to help secure the network transport.

Important Removal Steps

We recommend that you only use Distributed Replay in a test environment. After you have completed testing, and before you provision those computers for a different task, make sure that you do the following:

- Uninstall the Distributed Replay features and remove the related configuration files from the controller and all clients.

- Delete any trace, intermediate, dispatch, and SQL Server database files that were used for testing. The intermediate and dispatch files are stored in the working directory on the controller and client, respectively.

See Also

SQL Server Distributed Replay

Distributed Replay Installation How-to Topics

Administration Tool Command-line Options (Distributed Replay Utility)

The Microsoft SQL Server Distributed Replay administration tool, **DReplay.exe**, is a command-line tool that you can use to communicate with the distributed replay controller. Use the administration tool to initiate, monitor, and cancel operations on the controller.

For more information about the syntax conventions that are used with the administration tool syntax, see Transact-SQL Syntax Conventions (Transact-SQL).

Syntax

```
dreplay {preprocess|replay|status|cancel} [options] [-?]}

Usage:

  dreplay preprocess [-m controller] -i input_trace_file
    -d controller_working_dir [-c config_file] [-f status_interval]

  dreplay replay [-m controller] -d controller_working_dir [-o]
    [-s target_server] -w clients [-c config_file]
    [-f status_interval]

  dreplay status [-m controller] [-f status_interval]

  dreplay cancel [-m controller] [-q]
```

Remarks

You can issue the following command-line options with **DReplay.exe**:

preprocess

Initiates the preprocess stage. The controller prepares the input trace data, which you captured from the production environment, for replay against the target server.

replay

Initiates the event replay stage. The controller dispatches replay data to the specified clients, launches the distributed replay, and synchronizes the clients. Optionally, each client that was selected records the replay activity and saves result trace files locally.

status

Queries the controller and displays the current status.

cancel

Cancels the current operation that is running on the controller.

For detailed syntax information that includes the command arguments and examples, see the following topics:

- dreplay preprocess Command
- dreplay replay Command
- dreplay status Command
- dreplay cancel Command

RPCs are replayed as RPCs and not as language events.

Permissions

You must run the administration tool as an interactive user, as either a local user or a domain user account. To use a local user account, the administration tool and controller must be running on the same computer.

For more information, see Distributed Replay Security.

See Also

SQL Server Distributed Replay

Preprocess Option (Distributed Replay Administration Tool)

The Microsoft SQL Server Distributed Replay administration tool, **DReplay.exe**, is a command-line tool that you can use to communicate with the distributed replay controller. This topic describes the **preprocess** command-line option and corresponding syntax.

The **preprocess** option initiates the preprocess stage. During this stage, the controller prepares the input trace data for replay against the target server.

For more information about the syntax conventions that are used with the administration tool syntax, see Transact-SQL Syntax Conventions (Transact-SQL).

Syntax

```
dreplay preprocess [-m controller] -i input_trace_file
   -d controller_working_dir [-c config_file] [-f status_interval]
```

Parameters

Parameter	Description
-mcontroller	Specifies the computer name of the controller. You can use "`localhost`" or "`.`" to refer to the local computer. If the **-m** parameter is not specified, the local computer is used.
-iinput_trace_file	Specifies the full path of the input trace file on the controller, such as `D:\Mytrace.trc`. The **-i** parameter is required. If there are rollover files in the same directory, they will be loaded and used automatically. The files must follow the file rollover naming convention, for example: `Mytrace.trc`, `Mytrace_1.trc`, `Mytrace_2.trc`, `Mytrace_3.trc`, ... `Mytrace_n.trc`. **Note** If you are using the administration tool on a different computer than the controller, you will need to copy the input trace files to the controller so that a local path can be used for this parameter.
-dcontroller_working_dir	Specifies the directory on the controller where the intermediate file will be stored. The **-d** parameter is required. The following requirements apply: • The directory must reside on the controller. • You must specify the full path, starting

	with a drive letter (for example, `c:\WorkingDir`). - The path must not end with a backslash "\". - UNC paths are not supported.
-cconfig_file	Is the full path of the preprocess configuration file; used to specify the location of the preprocess configuration file when stored in a different location. This parameter can be a UNC path, or can reside locally on the computer where you run the administration tool. The **-c** parameter is not required if no filtering is needed, or if you do not want to modify the maximum idle time. Without the **-c** parameter, the default preprocess configuration file, `DReplay.exe.preprocess.config`, is used.
-fstatus_interval	Specifies the frequency (in seconds) at which to display status messages. If **-f** is not specified, the default interval is 30 seconds.

Examples

In this example, the preprocess stage is initiated with all of the default settings. The value `localhost` indicates that the controller service is running on the same computer as the administration tool. The input_trace_file parameter specifies the location of the input trace data, `c:\mytrace.trc`. Because there is no trace file filtering involved, the **-c** parameter does have to be specified.

```
dreplay preprocess –m localhost -i c:\mytrace.trc -d c:\WorkingDir
```

In this example, the preprocess stage is initiated and a modified preprocess configuration file is specified. Unlike the previous example, the **-c** parameter is used to point to the modified configuration file, if you have stored it in a different location. For example:

```
dreplay preprocess –m localhost -i c:\mytrace.trc -d c:\WorkingDir -c
c:\DReplay.exe.preprocess.config
```

In the modified preprocess configuration file, a filter condition is added that filters out system sessions during distributed replay. The filter is added by modifying the

`<PreprocessModifiers>` element in the preprocess configuration file, `DReplay.exe.preprocess.config`.

The following shows an example of the modified configuration file:

```
<?xml version='1.0'?>

<Options>

    <PreprocessModifiers>

        <IncSystemSession>No</IncSystemSession>

        <MaxIdleTime>-1</MaxIdleTime>

    </PreprocessModifiers>

</Options>
```

Permissions

You must run the administration tool as an interactive user, as either a local user or a domain user account. To use a local user account, the administration tool and controller must be running on the same computer.

For more information, see Distributed Replay Security.

See Also

Distributed Replay

How to: Preprocess the Trace

SQL Server Distributed Replay

Distributed Replay Configuration Files

Replay Option (Distributed Replay Administration Tool)

The Microsoft SQL Server Distributed Replay administration tool, **DReplay.exe**, is a command-line tool that you can use to communicate with the distributed replay controller. This topic describes the **replay** command-line option and corresponding syntax.

The **replay** option initiates the event replay stage, in which the controller dispatches replay data to the specified clients, launches the distributed replay and synchronizes the clients. Optionally, each client participating in the replay can record the replay activity and save a result trace file locally.

For more information about the syntax conventions that are used with the administration tool syntax, see Transact-SQL Syntax Conventions (Transact-SQL).

Syntax

```
dreplay replay [-m controller] -d controller_working_dir [-o]
  [-s target_server] -w clients [-c config_file]
  [-f status_interval]
```

Parameters

Parameter	Description
-mcontroller	Specifies the computer name of the controller. You can use "`localhost`" or "`.`" to refer to the local computer. If the **-m** parameter is not specified, the local computer is used.
-dcontroller_working_dir	Specifies the directory on the controller where the intermediate file will be stored. The **-d** parameter is required. The following requirements apply: • The directory must reside on the controller. • You must specify the full path, starting with a drive letter (for example, `c:\WorkingDir`). • The path must not end with a backslash "`\`". • UNC paths are not supported.
-o	Captures the clients' replay activity and saves it to a result trace file in the path specified by the **`<ResultDirectory>`** element in the client configuration file, `DReplayClient.xml`. When the **–o** parameter is not specified, the result trace file is not generated. The console output returns summary information at the end of replay, but no other replay statistics are available.
-starget_server	Specifies the target instance of SQL Server that the distributed workload should be replayed against. You must specify this parameter in the format **server_name[\instance name]**.

	You cannot use "`localhost`" or "`.`" as the target server. The **-s** parameter is not required if the **`<Server>`** element is specified in the **`<ReplayOptions>`** section of the replay configuration file, `DReplay.exe.replay.config`. If the **-s** parameter is used, the **`<Server>`** element in the **`<ReplayOptions>`** section of the replay configuration file will be ignored.
-wclients	This required parameter is a comma-separated list (without spaces) that specifies the computer names of clients that should participate in the distributed replay. IP addresses are not allowed. Be aware that the clients must already be registered with the controller. **Note** Each client registers with the controller that is specified in the client configuration file when the client service starts.
-cconfig_file	Is the full path of the replay configuration file; used to specify the location when it is stored in a different location. The **-c** parameter is not required if you want to use the default values of the replay configuration file, `DReplay.exe.replay.config`.
-fstatus_interval	Specifies the frequency (in seconds) at which to display the status. If **-f** is not specified, the default interval is 30 seconds.

Examples

In this example, the distributed replay derives much of its behavior from a modified replay configuration file, `DReplay.exe.replay.config`.

- The **-m** parameter specifies that a computer named `controller1` acts as the controller. The computer name must be specified when the controller service is running on a different computer.

- The **-d** parameter specifies the location of the intermediate file on the controller, `c:\WorkingDir`.
- The **-o** parameter specifies that each specified client capture the replay activity and save it to a result trace file. Note: The **`<ResultTrace>`** element in the configuration file can be used to specify if row count and result set be recorded.
- The **-w** parameter specifies that computers `client1` through `client4` participate as clients in the distributed replay.
- The **-c** parameter is used to point to the modified configuration file, `DReplay.exe.replay.config`.
- The **-s** parameter is not required because the **`<Server>`** element is specified in the **`<ReplayOptions>`** element of the replay configuration file, `DReplay.exe.replay.config`.

The event replay stage is initiated with the following syntax, when the administration tool is run from a different computer than the controller:

```
dreplay replay -m controller1 -d c:\WorkingDir -o -w
client1,client2,client3,client4 -c c:\DReplay.exe.replay.config
```

To specify a synchronous sequencing mode, the **`<SequencingMode>`** element of the `DReplay.exe.replay.config` file is set equal to the value `synchronization`. The **`<ResultTrace>`** section of the replay configuration file is modified to specify that row count be recorded. These changes are shown in the following XML example:

```
<?xml version='1.0'?>
<Options>
    <ReplayOptions>
        <Server>server_name\replay_target_instance</Server>
        <SequencingMode>synchronization</SequencingMode>
        <ConnectTimeScale></ConnectTimeScale>
        <ThinkTimeScale></ThinkTimeScale>
        <HealthmonInterval>60</HealthmonInterval>
        <QueryTimeout>3600</QueryTimeout>
        <ThreadsPerClient></ThreadsPerClient>
    </ReplayOptions>
    <OutputOptions>
        <ResultTrace>
            <RecordRowCount>Yes</RecordRowCount>
            <RecordResultSet>No</RecordResultSet>
        </ResultTrace>
    </OutputOptions>
</Options>
```

To specify a stress sequencing mode, the **`<SequencingMode>`** element of the `DReplay.exe.replay.config` file is set equal to the value `stress`. The **`<ConnectTimeScale>`** and **`<ThinkTimeScale>`** elements are set to the value `50` (to specify 50 percent). For more information about connect time and think time, see SQL Server Distributed Replay. These changes are shown in the following XML example:

```
<?xml version='1.0'?>
<Options>
    <ReplayOptions>
        <Server>server_name\replay_target_instance_name</Server>
        <SequencingMode>stress</SequencingMode>
        <ConnectTimeScale>50</ConnectTimeScale>
        <ThinkTimeScale>50</ThinkTimeScale>
        <HealthmonInterval>60</HealthmonInterval>
        <QueryTimeout>3600</QueryTimeout>
        <ThreadsPerClient></ThreadsPerClient>
    </ReplayOptions>
    <OutputOptions>
        <ResultTrace>
            <RecordRowCount>Yes</RecordRowCount>
            <RecordResultSet>No</RecordResultSet>
        </ResultTrace>
    </OutputOptions>
</Options>
```

Permissions

You must run the administration tool as an interactive user, as either a local user or a domain user account. To use a local user account, the administration tool and controller must be running on the same computer.

For more information, see Distributed Replay Security.

See Also

How to: Replay the Trace

How to: Review the Result Trace

SQL Server Distributed Replay

Distributed Replay Configuration Files

Status Option (Distributed Replay Administration Tool)

The Microsoft SQL Server Distributed Replay administration tool, **DReplay.exe**, is a command-line tool that you can use to communicate with the distributed replay controller. This topic describes the **status** command-line option and corresponding syntax.

The **status** option queries the controller and displays the current status.

For more information about the syntax conventions that are used with the administration tool syntax, see Transact-SQL Syntax Conventions (Transact-SQL).

Syntax

dreplay status [-m **controller**] [-f **status_interval**]

Parameters

Parameter	Description
-mcontroller	Specifies the computer name of the controller. You can use "`localhost`" or "`.`" to refer to the local computer. If the **-m** parameter is not specified, the local computer is used.
-fstatus_interval	Specifies the frequency (in seconds) at which to display the status. If the **-f** parameter is not specified, the default interval is 30 seconds.

Examples

In the following example, the current status is displayed every 60 seconds. The value `localhost` indicates that the controller service is running on the same computer as the administration tool.

```
dreplay status –m localhost -f 60
```

Permissions

You must run the administration tool as an interactive user, as either a local user or a domain user account. To use a local user account, the administration tool and controller must be running on the same computer.

For more information, see Distributed Replay Security.

See Also

Distributed Replay

Transact-SQL Debugger

Cancel Option (Distributed Replay Administration Tool)

The Microsoft SQL Server Distributed Replay administration tool, **DReplay.exe**, is a command-line tool that you can use to communicate with the distributed replay controller. This topic describes the **cancel** command-line option and corresponding syntax.

The **cancel** option cancels the current operation that is running on the controller.

For more information about the syntax conventions that are used with the administration tool syntax, see Transact-SQL Syntax Conventions (Transact-SQL).

Syntax

```
dreplay cancel [-m controller] [-q]
```

Parameters

Parameter	Description
-mcontroller	The computer name of the controller. You can use "`localhost`" or "`.`" to refer to the local computer. If the **-m** parameter is not specified, the local computer is used.
-q	Quiet mode. Does not prompt for confirmation. The **-q** parameter is optional.

Examples

In the following example, a cancel request is submitted in quiet mode. The value `localhost` indicates that the controller service is running on the same computer as the administration tool.

```
dreplay cancel -m localhost -q
```

Permissions

You must run the administration tool as an interactive user, as either a local user or a domain user account. To use a local user account, the administration tool and controller must be running on the same computer.

For more information, see Distributed Replay Security.

See Also

SQL Server Distributed Replay