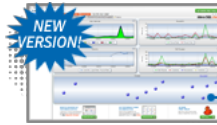




Solving SQL Server problems for
millions of DBAs & Devs since 2006.
Accurate • Complete • Trusted



SQL CHECK IS **ALL NEW, ALL FREE,**
AND ALL OVER YOUR SQL SERVER
Monitor SQL Server performance including SQL throughput

[DOWNLOAD](#)

sql DEVELOPER
BUNDLE

12 tools for efficient database development.

[Learn more](#)
redgate

SQL SERVER RESOURCES

[Database Administration](#)
[Database Development](#)
[Business Intelligence](#)
[Career Corner](#)
[All Tips Categories](#)
[Tutorials](#)
[Webcasts](#)
[Whitepapers](#)
[Tools](#)
[Questions & Answers](#)
[Authors](#)
[About Us](#)
[User Groups](#)
[Events](#)
[I am MSSQLTips](#)
[Advertise](#)


SQL Server Service Broker Poison Message Handling



By: [Arshad Ali](#) | [Read Comments](#) | Related Tips: [More](#) > [Service Broker](#)

[Free Learning - The Performance Faceoff: Oracle vs. SQL Server vs. Sybase](#)

Problem

Have you ever figured out why your Service Broker queue gets disabled automatically? What causes your Service Broker queue to get disabled in the first place? What is a poison message with respect to Service Broker? Is there any in SQL Server 2008 R2 for managing poison messages in Service Broker?

Solution

A poison message in [Services Broker](#) is message that cannot be processed by your code/activation stored procedure causing your code/activation stored procedure to rollback. In these circumstances the message is returned back to queue. Unfortunately, these messages will continue to be picked up by your code/activation stored procedure for a rolled back transaction and the message is returned back to queue. In other words, a poison message is an invalid message.

Here are some common scenarios which create poison messages:

- A message is violating foreign key/unique constraints or check constraints
- A message trying to insert a NULL values into a column (NOT NULL) that does not accept NULLs
- A message that attempts to insert an incompatible value into column
- Any data which causes your activation stored procedure to rollback

In these scenarios, the message processing code/activation stored procedure would not be able to process the message and the message would be returned back to the queue after rolling back the transaction. Please note, a poison message is not a corrupt message as Service Broker itself does integrity check for detecting message corruption.

Unfortunately there is no built-in mechanism to handle (delete) poison messages. You need to write custom code to manage these records. A simple way to approach this is in your activation stored procedure instead of rolling back a transaction in CATCH block of your [TRY...CATCH error handling](#) check the record to see if the rollback was due to a poison message. If yes, then log this message in a dedicated error queue. Then as a portion of your business process review these records to identify all the offending/poison messages which caused the control to go to CATCH block.

Another approach is to subscribe to the Broker:Queue Disabled trace event or BROKER_QUEUE_DISABLED event gets raised when a queue gets disabled after five consecutive rollbacks. On occurrence of such event, you need to receive each message from the queue. If the message is correct, then rollback the transaction so that it returns to queue for actual processing or if it is a poison message, log it to an exceptions log for auditing purposes and the transaction to remove the poison messages from your queue.

With the release of SQL Server 2008, [Service Broker](#) handles poison messages automatically by disabling (status OFF and a trace event of type Broker:Queue Disabled is generated) the queue after five consecutive rollbacks (code/activation stored procedure while retrieving and processing poison message from the queue. This is the default behavior and there is no way to disable it. You can create a [SQL Agent Job](#) to run on a predefined schedule to check the status of the queue and notify you if the queue is disabled. Or you build an application to subscribe to the BROKER_QUEUE_DISABLED event.

Once you have identified the queue is disabled, then you need to identify the messages which cannot be processed (poison messages). You need to end the conversations that have the poison messages with an error. This is the sender that the messages of the conversation cannot be processed. When a conversation is ended, Service Broker discards all the messages belonging to that conversation, so these need to be accounted for as well.

Top SQL Server Tools

[SQL safe backup](#)
[SQL compliance manager](#)
[SQL diagnostic manager](#)
[SQL Sentry Performance Advisor](#)
[SQL Developer Bundle](#)

```
-- Script #1 - Ending conversation handle which contains poison messages
END CONVERSATION {Conversation handle}
WITH ERROR = 127 DESCRIPTION = N'This conversation contains a poison
messages and it was ended without processing the messages.'
GO
```

With SQL Server 2008 R2, we have additional control in Service Broker to determine if we really want to automatically disable the queue when a poison message is detected i.e. automatic poison message detection. We now have the option to disable this automatic poison message detection and write our own custom code to handle the poison messages. In the next script block, you can see that I am creating a queue which has automatic detection of poison messages which is the default behavior.

```
-- Script #2 - Creating queue with automatic poison message detection
-- Creating a queue with enabled status and with automatic poison message detection
CREATE QUEUE {Queue Name}
WITH STATUS = ON,
POISON_MESSAGE_HANDLING (STATUS = ON)
GO
```

In the next script block, you can see that I am creating a queue which has automatic detection of poison messages with status OFF, which means Service Broker will not automatically detect poison messages and disable the queue after five consecutive rollbacks.

NEW VERSION!

**ALL NEW!
ALL FREE!**

**AND ALL OVER
YOUR SQL SERVER**

*A kick-ass
SQL Server
performance
monitoring
solution.*

SQLcheck™

DOWNLOAD NOW

idera

[Become a paid author](#)

2/3

[Solving SQL Server problems for millions of DBAs and Devs since 2006. Join now.](#)

[Demystify TempDB Performance and Manageability](#)

Copyright (c) 2006-2013 [Edgewood Solutions, LLC](#) All rights reserved

[privacy](#) | [disclaimer](#) | [copyright](#) | [advertise](#) | [about](#)
[authors](#) | [contribute](#) | [feedback](#) | [giveaways](#) | [user groups](#)

Some names and products listed are the registered trademarks of their respective owners.

[Edgewood Solutions LLC](#) | [MSSharePointTips.com](#) | [MSSQLTips.com](#)
