# sp_who (Transact-SQL)

03/14/2017 • 3 minutes to read • Contributors

### In this article

Syntax
Arguments
Return Code Values
Result Sets
Remarks
Permissions
Examples
See Also

**THIS TOPIC APPLIES TO:** SQL Server (starting with 2008) Azure SQL Database Azure SQL Data Warehouse Parallel Data Warehouse

Provides information about current users, sessions, and processes in an instance of the Microsoft SQL Server Database Engine. The information can be filtered to return only those processes that are not idle, that belong to a specific user, or that belong to a specific session.

Transact-SQL Syntax Conventions

## Syntax

Copy

```
sp_who [ [ @loginame = ] 'login' | session ID | 'ACTIVE' ]
```

## Arguments

FEEDBACK

[ **@loginame** = ] '*login*' | *session ID* | **'ACTIVE'**
Is used to filter the result set.

*login* is **sysname** that identifies processes belonging to a particular login.

*session ID* is a session identification number belonging to the SQL Server instance. *session ID* is **smallint**.

**ACTIVE** excludes sessions that are waiting for the next command from the user.

If no value is provided, the procedure reports all sessions belonging to the instance.

# Return Code Values

0 (success) or 1 (failure)

# Result Sets

**sp_who** returns a result set with the following information.

| Column | Data type | Description |
|---|---|---|
| **spid** | **smallint** | Session ID. |
| **ecid** | **smallint** | Execution context ID of a given thread associated with a specific session ID.<br><br>ECID = {0, 1, 2, 3, ...*n*}, where 0 always represents the main or parent thread, and {1, 2, 3, ...*n*} represent the subthreads. |
| **status** | **nchar(30)** | Process status. The possible values are:<br><br>**dormant**. SQL Server is resetting the session.<br><br>**running**. The session is running one or more batches. When Multiple Active Result Sets (MARS) is enabled, a session can run multiple batches. For more information, see Using Multiple Active Result Sets (MARS).<br><br>**background**. The session is running |

| | | |
|---|---|---|
| | | a background task, such as deadlock detection.<br><br>**rollback**. The session has a transaction rollback in process.<br><br>**pending**. The session is waiting for a worker thread to become available.<br><br>**runnable**. The session's task is in the runnable queue of a scheduler while waiting to get a time quantum.<br><br>**spinloop**. The session's task is waiting for a spinlock to become free.<br><br>**suspended**. The session is waiting for an event, such as I/O, to complete. |
| **loginame** | **nchar(128)** | Login name associated with the particular process. |
| **hostname** | **nchar(128)** | Host or computer name for each process. |
| **blk** | **char(5)** | Session ID for the blocking process, if one exists. Otherwise, this column is zero.<br><br>When a transaction associated with a specified session ID is blocked by an orphaned distributed transaction, this column will return a '-2' for the blocking orphaned transaction. |
| **dbname** | **nchar(128)** | Database used by the process. |
| **cmd** | **nchar(16)** | Database Engine command ( Transact-SQL statement, internal Database Engine process, and so on) executing for the process. |
| **request_id** | **int** | ID for requests running in a specific session. |

In case of parallel processing, subthreads are created for the specific session ID. The main thread is indicated as `spid = <xxx>` and `ecid =0`. The other subthreads have the same `spid = <xxx>`, but with **ecid** > 0.

# Remarks

A blocking process, which may have an exclusive lock, is one that is holding resources that another process needs.

All orphaned distributed transactions are assigned the session ID value of '-2'. Orphaned distributed transactions are distributed transactions that are not associated with any session ID. For more information, see [Use Marked Transactions to Recover Related Databases Consistently (Full Recovery Model)](#).

Query the **is_user_process** column of sys.dm_exec_sessions to separate system processes from user processes.

# Permissions

Requires VIEW SERVER STATE permission on the server to see all executing sessions on the instance of SQL Server. Otherwise, the user sees only the current session.

# Examples

## A. Listing all current processes

The following example uses `sp_who` without parameters to report all current users.

Copy

```
USE master;
GO
EXEC sp_who;
GO
```

## B. Listing a specific user's process

The following example shows how to view information about a single current user by login name.

Copy

```
USE master;
GO
```

```
EXEC sp_who 'janetl';
GO
```

### C. Displaying all active processes

Copy

```
USE master;
GO
EXEC sp_who 'active';
GO
```

### D. Displaying a specific process identified by a session ID

Copy

```
USE master;
GO
EXEC sp_who '10' --specifies the process_id;
GO
```

## See Also

sp_lock (Transact-SQL)
sys.sysprocesses (Transact-SQL)
System Stored Procedures (Transact-SQL)

**Note**

The feedback system for this content will be changing soon. Old comments will not be carried over. If content within a comment thread is important to you, please save a copy. For more information on the upcoming change, we invite you to read our blog post.

**1 Comment**

**Sign in**

**8 people listening**

+ Follow

Post comment as...

**Bobthiele** Nov 13, 2017

status is missing sleeping and rollback

Like Reply