CHAPTER

SQL Server Agent Security 14

INFORMATION IN THIS CHAPTER:

- Proxies
- SQL agent job steps
- Granting rights to proxies
- Job ownership

This chapter talks about the various permissions which can be used to allow non-privileged users to use some of the more powerful portions of the SQL Server Agent job system.

PROXIES

Proxies are used by the SQL Server Agent to run job steps under specific Windows accounts which are different from the account which runs the SQL Server Agent. Proxies are created based on Credentials. To create a proxy, connect to the SQL Server instance using the object explorer. Within the object explorer navigate to SQL Server Agent then Proxies. Right click on Proxies and select "New Proxies" from the context menu which opens. This opens the "New Proxy Account" screen as shown in Figure 14.1.

In the "New Proxy Account" window enter the name which you with to name the proxy account within the "Proxy name" field. In the "Credential name" field, enter the name of the Credential which you wish to bind the proxy account to. You can use the button with the three dots, shown in the upper right of Figure 14.1, to search for the Credential from the available Credentials on the instance. In the "Description" field you can enter an optional description to save information about what the proxy will be used for or other notes about the proxy.

At the bottom of Figure 14.1 you can see the list of available subsystems which can be used by the proxy. When creating a proxy it is recommended to only make the proxy account active for the specific subsystems which the account needs to be used for. As seen in Figure 14.1 multiple subsystems can be selected, however, this is not required. Additional subsystems can be added after the proxy has been created by editing the proxy and checking additional subsystems.

Proxy accounts can be created via T-SQL as well as via SQL Server Management Studio. Proxies require using two different stored procedures from the msdb database

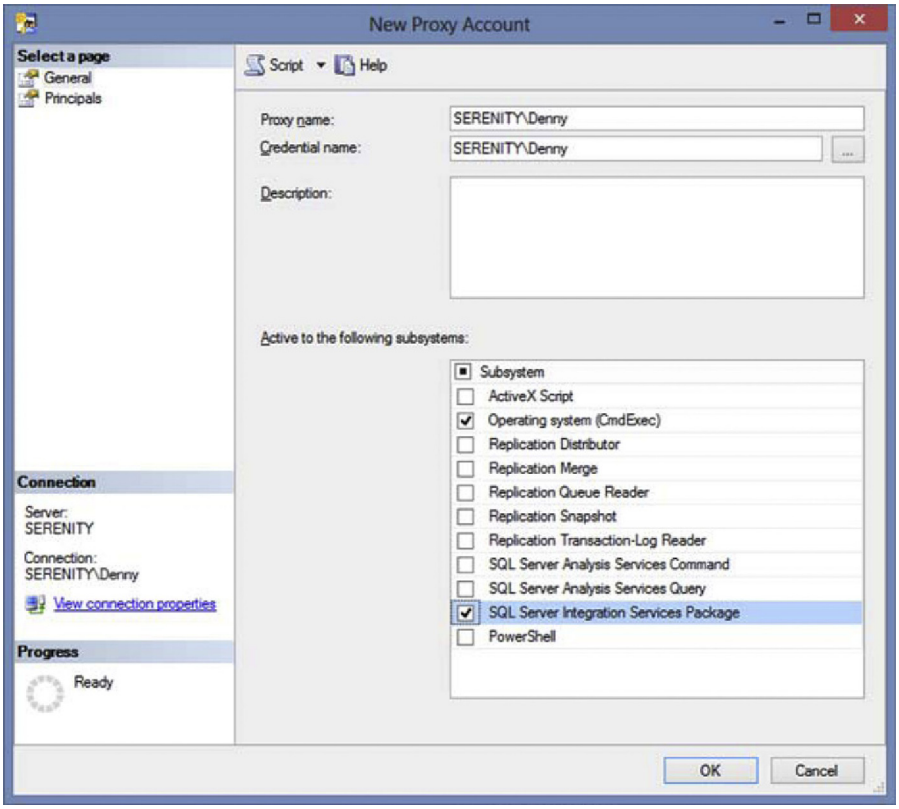

FIGURE 14.1 "New Proxy Account" window.

Table 14.1 Parameters for the sp_add_proxy Stored Procedure

Parameter Name	Description
@proxy_name	The name of the proxy account.
@credential_name	The name of the credential which the proxy account is bound to.
@enabled	If the proxy account is enabled or disabled where 1 is enabled and 0 is disabled.

to create the Proxy. The first is the sp_add_proxy stored procedure which creates the actual proxy. The second is the sp_grant_proxy_to_subsystem stored procedure which is used to grant the proxy account access to the specific subsystems.

The sp_add_proxy stored procedure accepts three input parameters as shown in Table 14.1.

The sp_grant_proxy_to_subsystem stored procedure accepts four input parameters as shown in Table 14.2.

Table 14.2 Parameters for the sp_grant_proxy_to_subsystem

Parameter Name	Description
@proxy_name	The name of the proxy account.
@proxy_id	The ID number of the proxy account.
@subsystem_id	The subsystem which the specified proxy account should be granted access to. The available list of Subsystem IDs are listed in Table 14.3
@subsystem_name	The name of the subsystem which the specified proxy account should be granted access to. The available values are ActiveScripting, CmdExec, Snapshot, LogReader, Distribution, Merge, QueueReader, ANALYSISQUERY, ANALYSISCOMMAND, Dts, and PowerShell.

Table 14.3 Available Proxy Subsystems

Subsystem ID	Subsystem Name	Subsystem Description
2	ActiveScripting	Microsoft ActiveX Script
3	CmdExec	Operating System
4	Snapshot	Replication Snapshot Agent
5	LogReader	Replication Log Reader Agent
6	Distribution	Replication Distribution Agent
7	Merge	Replication Merge Agent
8	QueueReader	Replication Queue Reader Agent
9	ANALYSISQUERY	Analysis Services Query
10	ANALYSISCOMMAND	Analysis Services Command
11	Dts	SSIS package execution
12	PowerShell	PowerShell Script

SQL AGENT JOB STEPS

Job steps within the SQL Server Agent are used to run whatever operation needs to be run by the SQL Server Agent job. Every type of SQL Server Agent job step type available has a proxy option with the exception of the T-SQL Job steps.

To change the account which will be used to run the SQL Server Agent job step, edit the job, then the job step. Change the "Run As" option to the proxy account you wish to use as shown in Figure 14.2, or select "SQL Server Agent Service Account" to use the account which runs the SQL Server Agent, which is the default.

After clicking OK on the job step and OK to save the job, the next time the job runs it will be run under the account which is specified in the drop down menu (Table 14.3).

To run a T-SQL job step a proxy account cannot be used. This is because T-SQL job steps do not need to authenticate against the operating system like all the other job step types do. In order to run a T-SQL Job Step as a different SQL Server user

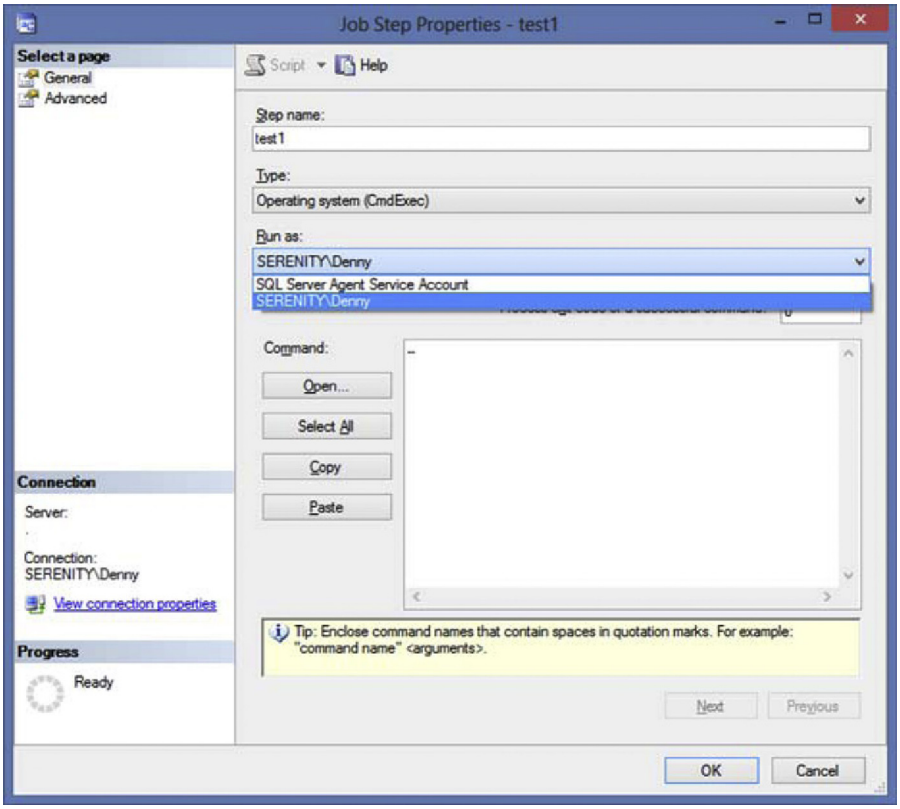

FIGURE 14.2 Job step properties.

ensure that the job step type is set to the job step type to "Transact SQL (T-SQL)" then select the Advanced tab on the left. From the advanced tab at the bottom of the window there is a "Run as user" field as shown in. Select the ellipsis button to select a user. The users which are in the master database are the users which are found within the master database. To have additional users appear in the list add them to the master database (Figure 14.3).

To add a proxy to a job step via T-SQL specify the @proxy_name parameter when using the sp_update_jobstep stored procedure and pass the name of the proxy account as the value of the parameter. This can be done for new jobs when the sp_add_jobstep stored procedure is used as well.

GRANTING RIGHTS TO PROXIES

Permissions can be granted to proxy accounts which allow specific users who are creating jobs to be able to use specific proxy accounts. This allows users who are not members of the sysadmin fixed server role to run SQL Agent job steps under a

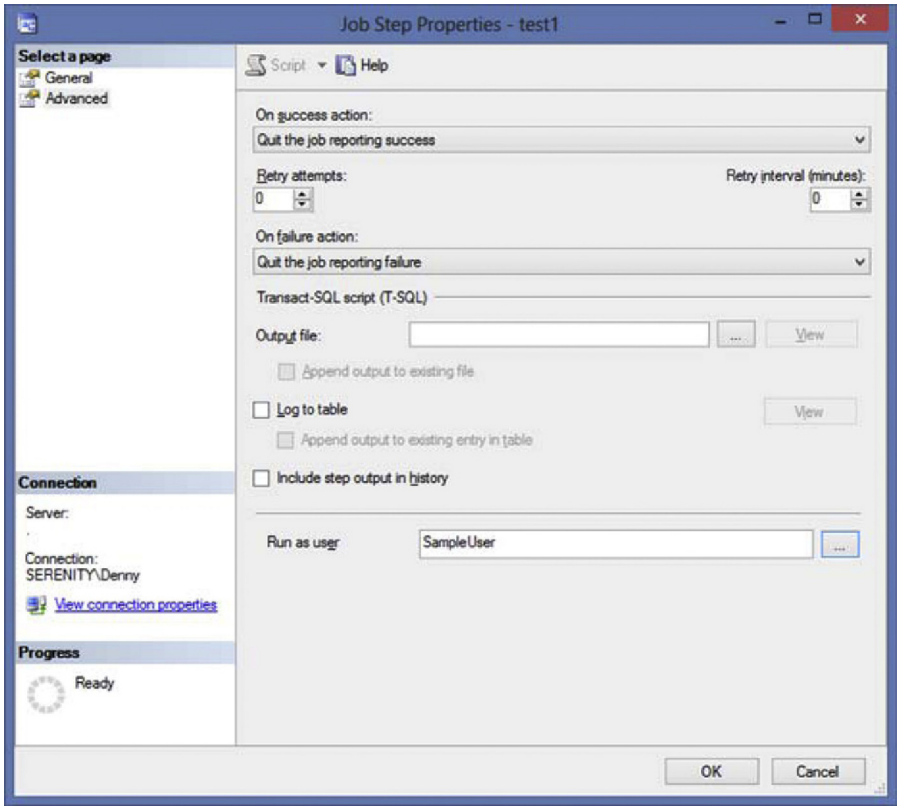

FIGURE 14.3 Advanced page of a T-SQL job step.

different Windows account. Members of the sysadmin fixed server role do not need to have permissions granted. Permissions can be mapped to a login within the SQL Server instance, a role within the MSDB database or a server role.

When viewing the properties of a proxy, either on creation of the proxy or after the fact selecting the permissions page allows you to grant users the ability to use the proxy as shown in Figure 14.4. To remove a permission from a login or role simply select the role and click the "Remove" button shown at the bottom of Figure 14.4.

After navigating to the properties page, simply click the "Add" button, then on the new window which opens select category you wish to grant permissions to, then the specific user or role you wish to grant permissions to as shown in Figure 14.5.

Granting users or roles permissions to a proxy can be done via T-SQL as well as through SQL Server Management Studio's Graphical User Interface. Two separate stored procedures in the msdb database are used for permissions management for proxies, one to grant permissions to a proxy called sp_grant_login_to_proxy and one to revoke permissions to a proxy called sp_recoke_login_from_proxy.

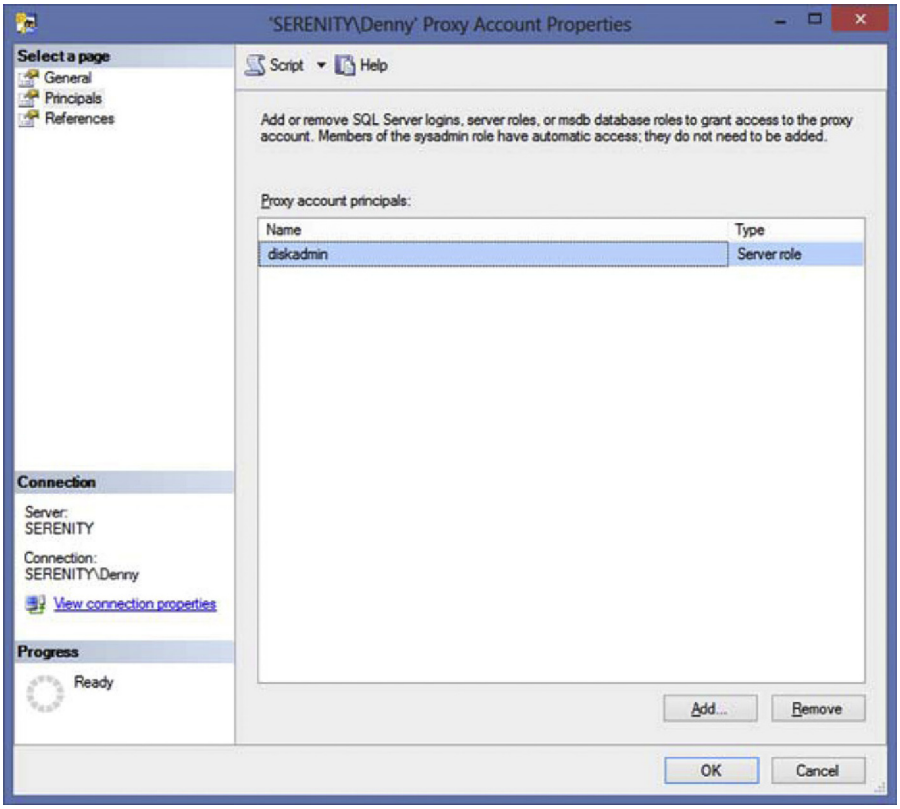

FIGURE 14.4 "Principals" page of the proxy creation and editing window.

The stored procedure sp_grant_login_to_proxy supports five parameters as shown in Table 14.4. Either the parameter @proxy_name or @proxy_id must always be specified as well as one of the other parameters.

The stored procedure sp_revoke_login_from_proxy supports three different parameters as shown in Table 14.5.

NOTE

Fixed Server Role?

You may be asking yourself why is the parameter called @fixed_server_role when it supports both fixed and user defined server roles?

The answer to that is simply because the when proxies were first introduced in SQL Server 2005 Microsoft did not have any idea that there would be user defined server roles introduced in SQL Server 2012 and it would have been much easier for the SQL Server development team to simply change the functionality of the parameter called @fixed_server_role then it would have been to change the parameter to @server_role as that would break the existing scripts that people may have which use this stored procedure.

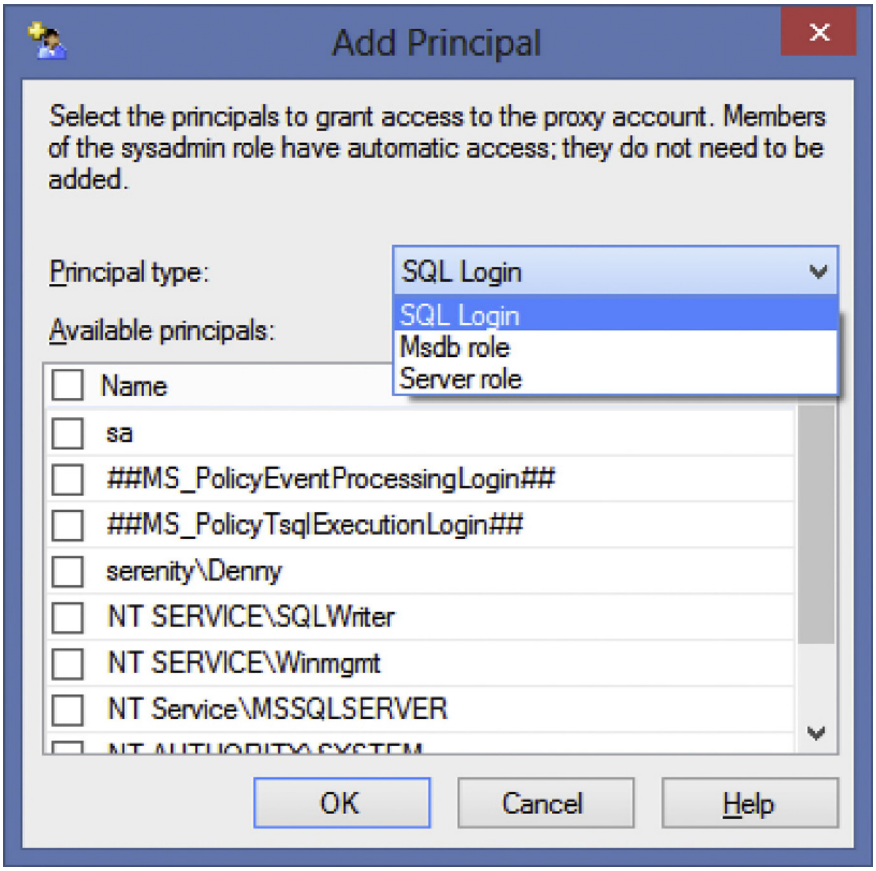

FIGURE 14.5 User or role selection page.

Table 14.4 Parameters for sp_grant_login_to_proxy

Parameter	Description
@proxy_name	The name of the proxy which permissions are being granted to.
@proxy_id	The ID number of the proxy which permissions are being granted to.
@login_name	A specific login which is being granted rights to the specified proxy.
@fixed_server_role	The name of a fixed or user defined server role which is being granted rights to the specified proxy.
@msdb_role	The name of a role from the msdb database which is being granted rights to the specified proxy.

Table 14.5 Parameters for sp_revoke_login_from_proxy

Parameter	Description
@name	The name of the login, fixed server role, user defined server role or role from the msdb database which should be removed from this proxy.
@proxy_id	The ID number of the proxy which permissions are being granted to.
@proxy_name	The name of the proxy which permissions are being granted to.

NOTE

What About Duplicate Names?

So what happens if you have a fixed server role and a role in the msdb database with both having the same name and you want to just remove one of them from having permissions to a proxy? The stored procedure sp_revoke_login_from_proxy will actually remove both of them from the database. Unfortunately there is no way around this by using this stored procedure. The stored procedure is written with the assumption that there would not ever be a user, server role or msdb database role with the same name. Because of that if you are in this situation you will need to delete the needed rows from the msdb.dbo.sysproxylogin database table manually instead of using the stored procedure. If you do use the stored procedure you will need to simply grant the needed permission back after deleting it.

JOB OWNERSHIP

Permission on jobs is one place where SQL Server fails from a security perspective. There is no way to grant specific users access to start, stop, or edit a specific SQL Server Agent job. Users have to be given rights to the SQL Server Agent roles (discussed in Chapter 13) which gives them specific global rights to the SQL Server Agent and either their jobs, or all jobs, but not to just some jobs which are owned by other users.

Members of the sysadmin fixed server role are the only users who can change the ownership of SQL Server Agent jobs.

SUMMARY

In this chapter you have learned how to run SQL Server Agent job steps as different users than the SQL Server Agent service account by using proxies. Proxies are a very powerful tool which can allow you to run jobs which have high level access to systems without the SQL Server Agent having those rights all the time when the job is not running.