Dsacls

Article • 08/31/2016

In this article

Applies To: Windows Server 2003, Windows Server 2008, Windows Server 2003 R2, Windows Server 2012, Windows Server 2003 with SP1, Windows 8

Displays and changes permissions (access control entries) in the access control list (ACL) of objects in Active Directory Domain Services (AD DS).

Dsacls is a command-line tool that is built into Windows Server 2008. It is available if you have the AD DS server role installed. To use **dsacls**, you must run the **dsacls** command from an elevated command prompt. To open an elevated command prompt, click **Start**, right-click **Command Prompt**, and then click **Run as administrator**.

For examples of how to use this command, see Examples.

Dsacls is the command-line equivalent of the **Security** tab in the p**roperties** dialog box for an Active Directory object in tools such as Active Directory Users and Computers. You can use either tool to view and change permissions to an Active Directory object.

> **Note**
>
> The access control entries (ACEs) that you add by using **dsacls** must be object-specific permissions that override the default permissions that are defined in the Active Directory schema for that object type. Do not add ACEs unless you are well-informed about security for Active Directory objects.

To view an ACL, the user must have Read permissions on Active Directory objects. To change an ACL, the user must have Write permissions on the Active Directory object.

Syntax

```
dsacls "[\\<Computer>\]<ObjectDN>" [/A] [/D <PermissionState
```

Parameters

If you specify an object without additional parameters, **Dsacls** displays the ACEs in the ACL.

Expand table

Parameter	Description
"[\\<Computer>\]<ObjectDN>"	Identifies the Active Directory object to investigate. Type the di of the object. To specify an object on a remote computer, type name followed by the distinguished name. This parameter mus quotation marks. For example: "CN=Jeff Akers,CN=Users,DC=domain,DC=test,DC=contoso,D or "\\Server01\CN=Jeff Akers,CN=Users,DC=domain,DC=test,DC
/A	Adds ownership and auditing information to the results.
/D	Denies the permissions that you specify to the user or group. You can deny permissions to multiple users in each **/D** comma /D Domain1\User1:CCDC Domain1\User2:DC;computer For more information, see Syntax for PermissionStatement[*Per*
/G	Grants the permissions that you specify to the user or group. You can grant permissions to multiple users in each **/G** comma /G Domain1\User1:CCDC Domain1\User2:DC;computer For more information, see Syntax for PermissionStatement[*Per*
/I:{T | S | P}	Specifies the objects to which you are applying the permissions determines whether the permissions are inheritable. **T** is the de • **T**: The object and its child objects

	• **S**: The child objects only • **P**: The object and child objects down to one level only (p permissions to one level only)
/N	Provides that the specified ACE replaces the current ACEs in th dsacls adds the ACE to the ACL.
/P:{Y | N}	Determines whether the object can inherit permissions from its you omit this parameter, the inheritance properties of the obje • **Y**: The object is protected and cannot inherit permissions • **N**: The object is not protected and can inherit permissior **Note** This parameter changes a property of the object, not of ar whether an ACE is inheritable, use the **/I** parameter.
/R {<User> | <Group>} [{<User> | <Group>}].	Deletes all ACEs for the users or groups that you specify. You c *User@Domain* or as *Domain\User*. You can specify *Group* as C as *Domain\Group*. You can delete ACEs for multiple users and groups in a single / example: /R Domain1\User1 Domain1\User2
/S	Restores the security on the object to the default for that obje the Active Directory schema.
/T	Restores the security on the tree of objects to the default for e This parameter is valid only with the **/S** parameter.
/?	Displays help at the command prompt.

Syntax for PermissionStatement

```
{<User> | <Group>}:<Permissions>[;{<ObjectType> | <Property>
```

Parameters

Expand table

Parameter	Description
{<User> | <Group>}	Specifies the user or group to whom the rights apply. You can specify *User* as *User@Domain* or *Domain\User*. You can specify *Group* as *Group@Domain* or *Domain\Group*.
<Permissions>	Specifies the type of permissions that you are applying. You can specify one or more of the following values (without spaces). **Generic permissions** • **GR**: Generic Read • **GE**: Generic Execute • **GW**: Generic Write • **GA**: Generic All **Specific permissions** • **SD**: Delete an object. • **DT**: Delete an object and all of its child objects. • **RC**: Read security information. • **WD**: Change security information. • **WO**: Change owner information. • **LC**: List the child objects of the object. • **CC**: Create a child object. If you do not specify {*ObjectType* | *Property*} to define a specific child object type, this permission applies to all types of child objects; otherwise, it applies only to the child object type that you specify.

- **DC**: Delete a child object.

  If you do not specify {*ObjectType* | *Property*} to define a specific child object type, this permission applies to all types of child objects; otherwise, it applies only to the child object type that you specify.
- **WS**: Write to a self object.

  This is meaningful only on group objects and when {*ObjectType* | *Property*} is a "member."
- **RP**: Read a property.

  If you do not specify {*ObjectType* | *Property*} to define a specific property, this permission applies to all properties of the object; otherwise, it applies only to the property of the object that you specify.
- **WP**: Write to a property.

  If you do not specify {*ObjectType* | *Property*} to define a specific property, this permission applies to all properties of the object; otherwise, it applies only to the property of the object that you specify.
- **CA**: Control access.

  If you do not specify {*ObjectType* | *Property*} to define the specific extended right for control access, this permission applies to all meaningful control accesses on the object; otherwise, it applies only to the specific extended right for that object.
- **LO**: List the object access.

  You can use this permission to grant list access to a specific object if List Children (**LC**) is not also granted to the parent object. You can also use this permission to deny access to list an object to hide an object if the user or group has **LC** permission on the parent object.

Expand table

Note

AD DS does not enforce this permission by default. You must configure AD DS to check for this permission.

{<ObjectType> | <Property>}	Limits the permission to the specified object type or property. Enter the display name of the object type or the property. If you do not specify an object type or property, the permission applies to all object types and properties. For example, the following command permits the user to create all types of child objects: /G Domain\User:CC In contrast, the following command permits the user to create only child computer objects: /G Domain\User:CC;computer
<InheritedObjectType>	Limits inheritance of the permission to the specified object type. Enter the display name of the object type. If you do not specify an object type, all object types can inherit the permission. You can use this parameter only when permissions are inheritable. For example, the following command permits all objects types to inherit the permission: /G Domain\User:CC In contrast, the following command permits only User objects to inherit the permission: /G Domain\User:CC;;user

Examples

To grant the permission to delete, read security information, change security information, and change ownership permissions on a User object, type:

```
SDRCWDWO;;user
```

To grant permission to create child objects and delete child objects of a Group object, type:

```
CCDC;group;
```

To grant permissions to read property and write property values on a Telephonenumber property, type:

```
RPWP;telephonenumber;
```

Additional references

Command-Line Syntax Key