This page is specific to

Microsoft Visual Studio 2010/.NET Framework 4

.NET Framework 4

**Caspol.exe (Code Access Security Policy Tool)**

Updated: April 2011

The Code Access Security (CAS) Policy tool (Caspol.exe) enables users and administrators to modify security policy for the machine policy level, the user policy level, and the enterprise policy level.

|  |
| --- |
| **Description: Important noteImportant** |
| In the .NET Framework version 4 and later versions, Caspol.exe does not affect CAS policy unless the [<legacyCasPolicy> element](http://msdn.microsoft.com/en-us/library/dd409253.aspx) is set to **true**. For more information, see [Security Changes in the .NET Framework 4](http://msdn.microsoft.com/en-us/library/dd233103.aspx). |
| **Description: NoteNote** |
| 64-bit computers include both 64-bit and 32-bit versions of security policy. To ensure that your policy changes apply to both 32-bit and 64-bit applications, run the 32-bit version of Caspol.exe by using the Visual Studio Command Prompt, and run the 64-bit version by using the Visual Studio x64 Win64 Command Prompt. |

This tool is automatically installed with Visual Studio and with the Windows SDK. You can find Caspol.exe in %windir%\Microsoft.NET\Framework\*version* on 32-bit systems or %windir%\Microsoft.NET\Framework64\*version* on 64-bit systems. (For example, the location is %windir%\Microsoft.NET\Framework64\v4.030319\caspol.exe for the .NET Framework 4 on a 64-bit system.) To run the tool, we recommend that you use the Visual Studio Command Prompt or the Windows SDK Command Prompt (CMD Shell). These utilities enable you to run the tool easily, without navigating to the installation folder. For more information, see [Visual Studio and Windows SDK Command Prompts](http://msdn.microsoft.com/en-us/library/ms229859.aspx).

* If you have Visual Studio installed on your computer: On the taskbar, click **Start**, click **All Programs**, click **Visual Studio**, click **Visual Studio Tools**, and then click **Visual Studio Command Prompt**.

-or-

If you have the Windows SDK installed on your computer: On the taskbar, click **Start**, click **All Programs**, click the folder for the Windows SDK, and then click **Command Prompt** (or **CMD Shell**).

* At the command prompt, type the following:

caspol [options]

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifParameters

|  |  |
| --- | --- |
| **Option** | **Description** |
| **-addfulltrust** *assembly\_file*  or  **-af** *assembly\_file* | Adds an assembly that implements a custom security object (such as a custom permission or a custom membership condition) to the full trust assembly list for a specific policy level. The *assembly\_file* argument specifies the assembly to add. This file must be signed with a [strong name](http://msdn.microsoft.com/en-us/library/wd40t7ad.aspx). You can sign an assembly with a strong name using the [Strong Name Tool (Sn.exe)](http://msdn.microsoft.com/en-us/library/k5b5tt23.aspx).  Whenever a permission set containing a custom permission is added to policy, the assembly implementing the custom permission must be added to the full trust list for that policy level. Assemblies that implement custom security objects (such as custom code groups or membership conditions) used in a security policy (such as the machine policy) should always be added to the full trust assembly list.  Description: Caution note**Caution**  If the assembly implementing the custom security object references other assemblies, you must first add the referenced assemblies to the full trust assembly list. Custom security objects created using Visual Basic, C++, and JScript reference either Microsoft.VisualBasic.dll, Microsoft.VisualC.dll, or Microsoft.JScript.dll, respectively. These assemblies are not in the full trust assembly list by default. You must add the appropriate assembly to the full trust list before you add a custom security object. Failure to do so will break the security system, causing all assemblies to fail to load. In this situation, the Caspol.exe **-all -reset** option will not repair security. To repair security, you must manually edit the security files to remove the custom security object. |
| **-addgroup** {*parent\_label | parent\_name*} *mship pset\_name* [*flags*]  or  **-ag** {*parent\_label | parent\_name*} *mship pset\_name* [*flags*] | Adds a new code group to the code group hierarchy. You can specify either the *parent\_label* or *parent\_name*. The *parent\_label* argument specifies the label (such as 1. or 1.1.) of the code group that is the parent of the code group being added. The *parent\_name* argument specifies the name of the code group that is the parent of the code group being added. Because *parent\_label* and *parent\_name* can be used interchangeably, Caspol.exe must be able to distinguish between them. Therefore, *parent\_name* cannot begin with a number. Additionally, *parent\_name* can only contain A-Z, 0-9 and the underscore character.  The *mship* argument specifies the membership condition for the new code group. For more information, see the table of *mship* arguments later in this section.  The *pset\_name* argument is the name of the permission set that will be associated with the new code group. You can also set one or more *flags* for the new group. For more information, see the table of *flags* arguments later in this section. |
| **-addpset** {*psfile* | *psfile* p*set\_name*}  or  **-ap** {*named*\_*psfile* | *psfile* *pset\_name*} | Adds a new named permission set to policy. The permission set must be authored in XML and stored in an .xml file. If the XML file contains the name of the permission set, only that file (*psfile*) is specified. If the XML file does not contain the permission set name, you must specify both the XML file name (*psfile*) and the permission set name (*pset\_name*).  Note that all permissions used in a permission set must be defined in assemblies contained in the global assembly cache. |
| **-a**[**ll**] | Indicates that all options following this one apply to the machine, user, and enterprise policies. The **-all** option always refers to the policy of the currently logged-on user. See the **-customall** option to refer to the user policy of a user other than the current user. |
| **-chggroup** {*label |name*} {*mship* | *pset\_name* |  *flags*}  or  **-cg** {*label |name*} {*mship* | *pset\_name* |  *flags*} | Changes a code group's membership condition, permission set, or the settings of the **exclusive**, **levelfinal**, **name**, or **description** flags. You can specify either the *label* or *name*. The *label* argument specifies the label (such as 1. or 1.1.) of the code group. The *name* argument specifies the name of the code group to change. Because *label* and *name* can be used interchangeably, Caspol.exe must be able to distinguish between them. Therefore, *name* cannot begin with a number. Additionally, *name* can only contain A-Z, 0-9 and the underscore character.  The *pset\_name* argument specifies the name of the permission set to associate with the code group. See the tables later in this section for information on the *mship* and *flags* arguments. |
| **-chgpset** *psfile pset\_name*  or  **-cp** *psfile pset\_name* | Changes a named permission set. The *psfile* argument supplies the new definition for the permission set; it is a serialized permission set file in XML format. The *pset\_name* argument specifies the name of the permission set you want to change. |
| **-customall** *path*  or  **-ca** *path* | Indicates that all options following this one apply to the machine, enterprise, and the specified custom user policies. You must specify the location of the custom user's security configuration file with the *path* argument. |
| **-cu**[**stomuser**] *path* | Allows the administration of a custom user policy that does not belong to the user on whose behalf Caspol.exe is currently running. You must specify the location of the custom user's security configuration file with the *path* argument. |
| **-enterprise**  or  **-en** | Indicates that all options following this one apply to the enterprise level policy. Users who are not enterprise administrators do not have sufficient rights to modify the enterprise policy, although they can view it. In nonenterprise scenarios, this policy, by default, does not interfere with machine and user policy. |
| **-e**[**xecution**] {**on** | **off**} | Turns on or off the mechanism that checks for the permission to run before code starts to execute.  Description: Note**Note**  This switch is removed in the .NET Framework 4 and later versions. For more information, see [Security Changes in the .NET Framework 4](http://msdn.microsoft.com/en-us/library/dd233103.aspx). |
| **-f**[**orce**] | Suppresses the tool's self-destruct test and changes the policy as specified by the user. Normally, Caspol.exe checks whether any policy changes would prevent Caspol.exe itself from running properly; if so, Caspol.exe does not save the policy change and prints an error message. To force Caspol.exe to change policy even if this prevents Caspol.exe itself from running, use the **–force** option. |
| **-h**[**elp**] | Displays command syntax and options for Caspol.exe. |
| **-l**[**ist**] | Lists the code group hierarchy and the permission sets for the specified machine, user, enterprise, or all policy levels. Caspol.exe displays the code group's label first, followed by the name, if it is not null. |
| **-listdescription**  or  **-ld** | Lists all code group descriptions for the specified policy level. |
| **-listfulltrust**  or  **-lf** | Lists the contents of the full trust assembly list for the specified policy level. |
| **-listgroups**  or  **-lg** | Displays the code groups of the specified policy level or all policy levels. Caspol.exe displays the code group's label first, followed by the name, if it is not null. |
| **-listpset** or **-lp** | Displays the permission sets for the specified policy level or all policy levels. |
| **-m**[**achine**] | Indicates that all options following this one apply to the machine level policy. Users who are not administrators do not have sufficient rights to modify the machine policy, although they can view it. For administrators, **-machine** is the default. |
| **-polchgprompt** {**on** | **off**}  or  **-pp** {**on** | **off**} | Enables or disables the prompt that is displayed whenever Caspol.exe is run using an option that would cause policy changes. |
| **-quiet**  or  **-q** | Temporarily disables the prompt that is normally displayed for an option that causes policy changes. The global change prompt setting does not change. Use the option only on a single command basis to avoid disabling the prompt for all Caspol.exe commands. |
| **-r**[**ecover**] | Recovers policy from a backup file. Whenever a policy change is made, Caspol.exe stores the old policy in a backup file. |
| **-remfulltrust** *assembly\_file*  or  **-rf** *assembly\_file* | Removes an assembly from the full trust list of a policy level. This operation should be performed if a permission set that contains a custom permission is no longer used by policy. However, you should remove an assembly that implements a custom permission from the full trust list only if the assembly does not implement any other custom permissions that are still being used. When you remove an assembly from the list, you should also remove any other assemblies that it depends on. |
| **-remgroup** {*label |name*}  or  **-rg** {l*abel | name*} | Removes the code group specified by either its label or name. If the specified code group has child code groups, Caspol.exe also removes all the child code groups. |
| **-rempset** *pset\_name*  or  **-rp** *pset\_name* | Removes the specified permission set from policy. The *pset\_name* argument indicates which permission set to remove. Caspol.exe removes the permission set only if it is not associated with any code group. The default (built-in) permission sets cannot be removed; for details, see [Named Permission Sets](http://msdn.microsoft.com/en-us/library/4652tyx7.aspx). |
| **-reset**  or  **-rs** | Returns policy to its default state and persists it to disk. This is useful whenever a changed policy seems to be beyond repair and you want to start over with the installation defaults. Resetting can also be convenient when you want to use the default policy as a starting point for modifications to specific security configuration files. For more information, see [Manually Editing the Security Configuration Files](http://msdn.microsoft.com/en-us/library/cb6t8dtz.aspx#cpgrfcodeaccesssecuritypolicyutilitycaspolexeanchor1). |
| **-resetlockdown**  or  **-rsld** | Returns policy to a more restrictive version of the default state and persists it to disk; creates a backup of the previous machine policy and persists it to a file called **security.config.bac**. The locked down policy is similar to the default policy, except that the policy grants no permission to code from the **Local Intranet**, **Trusted Sites**, and **Internet** zones and the corresponding code groups have no child code groups. |
| **-resolvegroup** *assembly\_file*  or  **-rsg** *assembly\_file* | Shows the code groups that a specific assembly (*assembly\_file*) belongs to. By default, this option displays the machine, user, and enterprise policy levels to which the assembly belongs. To view only one policy level, use this option with either the **-machine**, **-user**, or **-enterprise** option. |
| **-resolveperm** *assembly\_file*  or  **-rsp** *assembly\_file* | Displays all permissions that the specified (or default) level of security policy would grant the assembly if the assembly were allowed to run. The *assembly\_file* argument specifies the assembly. If you specify the **-all** option, Caspol.exe calculates the permissions for the assembly based on user, machine, and enterprise policy; otherwise, default behavior rules apply. |
| **-s**[**ecurity**] {**on** | **off**} | Turns code access security on or off. Specifying the **-s off** option does not disable role-based security.  Description: Note**Note**  This switch is removed in the .NET Framework 4 and later versions. For more information, see [Security Changes in the .NET Framework 4](http://msdn.microsoft.com/en-us/library/dd233103.aspx).  Description: Caution note**Caution**  When code access security is disabled, all code access demands succeed. Disabling code access security makes the system vulnerable to attacks by malicious code such as viruses and worms. Turning off security gains some extra performance but should only be done when other security measures have been taken to help make sure overall system security is not breached. Examples of other security precautions include disconnecting from public networks, physically securing computers, and so on. |
| **-u**[**ser**] | Indicates that all options following this one apply to the user level policy for the user on whose behalf Caspol.exe is running. For nonadministrative users, **-user** is the default. |
| **-?** | Displays command syntax and options for Caspol.exe. |

The *mship* argument, which specifies the membership condition for a code group, can be used with the **-addgroup** and **-chggroup** options. Each *mship* argument is implemented as a .NET Framework class. To specify *mship,* use one of the following.

|  |  |
| --- | --- |
| **Argument** | **Description** |
| **-allcode** | Specifies all code. For more information about this membership condition, see the [AllMembershipCondition Class](http://msdn.microsoft.com/en-us/library/system.security.policy.allmembershipcondition.aspx). |
| **-appdir** | Specifies the application directory. If you specify **–appdir** as the membership condition, the URL evidence of code is compared with the application directory evidence of that code. If both evidence values are the same, this membership condition is satisfied. For more information about this membership condition, see the [ApplicationDirectoryMembershipCondition Class](http://msdn.microsoft.com/en-us/library/system.security.policy.applicationdirectorymembershipcondition.aspx). |
| **-custom** *xmlfile* | Adds a custom membership condition. The mandatory *xmlfile* argument specifies the .xml file that contains XML serialization of the custom membership condition. |
| **-hash** *hashAlg* {**-hex** *hashValue* | **-file** *assembly\_file* } | Specifies code that has the given assembly hash. To use a hash as a code group membership condition, you must specify either the hash value or the assembly file. For more information about this membership condition, see the [HashMembershipCondition Class](http://msdn.microsoft.com/en-us/library/system.security.policy.hashmembershipcondition.aspx). |
| **-pub** { **-cert** *cert\_file\_name* |  **-file** *signed\_file\_name* | **-hex** *hex\_string* } | Specifies code that has the given software publisher, as denoted by a certificate file, a signature on a file, or the hexadecimal representation of an X509 certificate. For more information about this membership condition, see the [PublisherMembershipCondition Class](http://msdn.microsoft.com/en-us/library/system.security.policy.publishermembershipcondition.aspx). |
| **-site** *website* | Specifies code that has the given site of origin. For example:  **-site** www.proseware.com  For more information about this membership condition, see the [SiteMembershipCondition Class](http://msdn.microsoft.com/en-us/library/system.security.policy.sitemembershipcondition.aspx). |
| **-strong -file** *file\_name* {*name* | **-noname**} {*version* | **-noversion**} | Specifies code that has a specific strong name, as designated by the file name, the assembly name as a string, and the assembly version in the format *major*.*minor*.*build*.*revision*. For example:  **-strong -file** myAssembly.exe myAssembly 1.2.3.4  For more information about this membership condition, see the [StrongNameMembershipCondition Class](http://msdn.microsoft.com/en-us/library/system.security.policy.strongnamemembershipcondition.aspx). |
| **-url** *URL* | Specifies code that originates from the given URL. The URL must include a protocol, such as http:// or ftp://. Additionally, a wildcard character (\*) can be used to specify multiple assemblies from a particular URL.  Description: Note**Note**  Because a URL can be identified using multiple names, using a URL as a membership condition is not a safe way to ascertain the identity of code. Where possible, use a strong name membership condition, a publisher membership condition, or the hash membership condition.  For more information about this membership condition, see the [UrlMembershipCondition Class](http://msdn.microsoft.com/en-us/library/system.security.policy.urlmembershipcondition.aspx). |
| **-zone** *zonename* | Specifies code with the given zone of origin. The *zonename* argument can be one of the following values: **MyComputer**, **Intranet**, **Trusted**, **Internet**, or **Untrusted**. For more information about this membership condition, see the [ZoneMembershipCondition](http://msdn.microsoft.com/en-us/library/system.security.policy.zonemembershipcondition.aspx) Class. |

The *flags* argument, which can be used with the **–addgroup** and **–chggroup** options, is specified using one of the following.

|  |  |
| --- | --- |
| **Argument** | **Description** |
| **-description "***description***"** | If used with the **–addgroup** option, specifies the description for a code group to add. If used with the **–chggroup** option, specifies the description for a code group to edit. The *description* argument must be enclosed in double quotes. |
| **-exclusive** {**on**|**off**} | When set to **on**, indicates that only the permission set associated with the code group you are adding or modifying is considered when some code fits the membership condition of the code group. When this option is set to **off**, Caspol.exe considers the permission sets of all matching code groups in the policy level. |
| **-levelfinal** {**on**|**off**} | When set to **on**, indicates that no policy level below the level in which the added or modified code group occurs is considered. This option is typically used at the machine policy level. For example, if you set this flag for a code group at the machine level and some code matches this code group's membership condition, Caspol.exe does not calculate or apply the user level policy for this code. |
| **-name "***name***"** | If used with the **–addgroup** option, specifies the scripting name for a code group to add. If used with the **-chggroup** option, specifies the scripting name for a code group to edit. The *name* argument must be enclosed in double quotes. The *name* argument cannot begin with a number, and can only contain A-Z, 0-9, and the underscore character. Code groups can be referred to by this *name* instead of by their numeric label. The *name* is also highly useful for scripting purposes. |

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifRemarks

Security policy is expressed using three policy levels: machine policy, user policy, and enterprise policy. The set of permissions that an assembly receives is determined by the intersection of the permission sets allowed by these three policy levels. Each policy level is represented by a hierarchical structure of code groups. Every code group has a membership condition that determines which code is a member of that group. A named permission set is also associated with each code group. This permission set specifies the permissions the runtime allows code that satisfies the membership condition to have. A code group hierarchy, along with its associated named permission sets, defines and maintains each level of security policy. You can use the**–user**, **-customuser**, **–machine** and **-enterprise** options to set the level of security policy.

For more information about security policy and how the runtime determines which permissions to grant to code, see [Security Policy Management](http://msdn.microsoft.com/en-us/library/c1k0eed6.aspx).

**Referencing Code Groups and Permission Sets**

To facilitate references to code groups in a hierarchy, the **-list** option displays an indented list of code groups along with their numerical labels (1, 1.1, 1.1.1, and so on). The other command-line operations that target code groups also use the numerical labels to refer to specific code groups.

Named permission sets are referenced by their names. The **–list** option displays the list of code groups followed by a list of named permission sets available in that policy. For a list and descriptions of the built-in permission sets provided by the runtime, see [Named Permission Sets](http://msdn.microsoft.com/en-us/library/4652tyx7.aspx).

**Caspol.exe Behavior**

All options except **-s**[**ecurity**] {**on** | **off**} use the version of the .NET Framework that Caspol.exe was installed with. If you run the Caspol.exe that was installed with version *X* of the runtime, the changes apply only to that version. Other side-by-side installations of the runtime, if any, are not affected. If you run Caspol.exe from the command line without being in a directory for a specific runtime version, the tool is executed from the first runtime version directory in the path (usually the most recent runtime version installed).

The **-s**[**ecurity**] {**on** | **off**} option is a computer-wide operation. Turning off code access security terminates security checks for all managed code and for all users on the computer. If side-by-side versions of the .NET Framework are installed, this command turns off security for every version installed on the computer. Although the **-list** option shows that security is turned off, nothing else clearly indicates for other users that security has been turned off.

When a user without administrative rights runs Caspol.exe, all options refer to the user level policy unless the **–machine** option is specified. When an administrator runs Caspol.exe, all options refer to the machine policy unless the **–user** option is specified.

Caspol.exe must be granted the equivalent of the **Everything** permission set to function. The tool has a protective mechanism that prevents policy from being modified in ways that would prevent Caspol.exe from being granted the permissions it needs to run. If you try to make such changes, Caspol.exe notifies you that the requested policy change will break the tool, and the policy change is rejected. You can turn this protective mechanism off for a given command by using the **–force** option.

**Manually Editing the Security Configuration Files**

Three security configuration files correspond to the three policy levels supported by Caspol.exe: one for the machine policy, one for a given user's policy, and one for the enterprise policy. These files are created on disk only when machine, user, or enterprise policy is changed using Caspol.exe. You can use the **–reset** option in Caspol.exe to save the default security policy to disk, if needed.

In most cases, manually editing the security configuration files is not recommended. But there might be scenarios in which modifying these files becomes necessary, such as when an administrator wants to edit the security configuration for a particular user.

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifExamples

**-addfulltrust**

Assume that a permission set containing a custom permission has been added to machine policy. This custom permission is implemented in MyPerm.exe, and MyPerm.exe references classes in MyOther.exe. Both assemblies must be added to the full trust assembly list. The following command adds the MyPerm.exe assembly to the full trust list for the machine policy.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl00_code');" \o "Copy Code)

caspol -machine -addfulltrust MyPerm.exe

The following command adds the MyOther.exe assembly to the full trust list for the machine policy.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl01_code');" \o "Copy Code)

caspol -machine -addfulltrust MyOther.exe

**-addgroup**

The following command adds a child code group to the root of the machine policy code group hierarchy. The new code group is a member of the **Internet** zone and is associated with the **Execution** permission set.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl02_code');" \o "Copy Code)

caspol -machine -addgroup 1. -zone Internet Execution

The following command adds a child code group that gives the share \\netserver\netshare local intranet permissions.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl03_code');" \o "Copy Code)

caspol -machine -addgroup 1. -url \\netserver\netshare\\* LocalIntranet

**-addpset**

The following command adds the Mypset permission set to the user policy.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl04_code');" \o "Copy Code)

caspol -user -addpset Mypset.xml Mypset

**-chggroup**

The following command changes the permission set in the user policy of the code group labeled 1.2. to the **Execution** permission set.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl05_code');" \o "Copy Code)

caspol -user -chggroup 1.2. Execution

The following command changes the membership condition in the default policy of the code group labeled 1.2.1. and changes the setting of the **exclusive** flag. The membership condition is defined to be code that originates from the **Internet** zone and the **exclusive** flag is switched on.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl06_code');" \o "Copy Code)

caspol -chggroup 1.2.1. -zone Internet -exclusive on

**-chgpset**

The following command changes the permission set with name Mypset to the permission set contained in newpset.xml. Note that the current release does not support changing permission sets that are being used by the code group hierarchy.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl07_code');" \o "Copy Code)

caspol -chgpset Mypset newpset.xml

**-force**

The following command causes the user policy's root code group (labeled 1) to be associated with the **Nothing** named permission set. This prevents Caspol.exe from running.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl08_code');" \o "Copy Code)

caspol -force -user -chggroup 1 Nothing

**-recover**

The following command recovers the most recently saved machine policy.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl09_code');" \o "Copy Code)

caspol -machine -recover

**-remgroup**

The following command removes the code group labeled 1.1. If this code group has any child code groups, those groups are also deleted.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl10_code');" \o "Copy Code)

caspol -remgroup 1.1.

**-rempset**

The following command removes the **Execution** permission set from the user policy.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl11_code');" \o "Copy Code)

caspol -user -rempset Execution

The following command removes Mypset from the user policy level.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl12_code');" \o "Copy Code)

caspol -rempset MyPset

**-resolvegroup**

The following command shows all code groups of the machine policy that myassembly belongs to.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl13_code');" \o "Copy Code)

caspol -machine -resolvegroup myassembly

The following command shows all code groups of the machine, enterprise, and specified custom user policy that myassembly belongs to.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl14_code');" \o "Copy Code)

caspol -customall "c:\config\_test\security.config" -resolvegroup myassembly

**-resolveperm**

The following command calculates the permissions for testassembly based on the machine and user policy levels.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl63_ctl00_ctl15_code');" \o "Copy Code)

caspol -all -resolveperm testassembly