Process Security and Access Rights

Article • 01/07/2022 • 8 contributors

Feedback

The Microsoft Windows security model enables you to control access to process objects. For more information about security, see Access-Control Model.

When a user logs in, the system collects a set of data that uniquely identifies the user during the authentication process, and stores it in an access token. This access token describes the security context of all processes associated with the user. The security context of a process is the set of credentials given to the process or the user account that created the process.

You can use a token to specify the current security context for a process using the **CreateProcessWithTokenW** function. You can specify a security descriptor for a process when you call the **CreateProcess**, **CreateProcessAsUser**, or **CreateProcessWithLogonW** function. If you specify **NULL**, the process gets a default security descriptor. The ACLs in the default security descriptor for a process come from the primary or impersonation token of the creator.

To retrieve a process's security descriptor, call the **GetSecurityInfo** function. To change a process's security descriptor, call the **SetSecurityInfo** function.

The valid access rights for process objects include the standard access rights and some process-specific access rights. The following table lists the standard access rights used by all objects.

Expand table

Value	Meaning
DELETE (0x00010000L)	Required to delete the object.
READ_CONTROL (0x00020000L)	Required to read information in the security descriptor for the object, not including the information in the SACL. To read or write the SACL, you must request the **ACCESS_SYSTEM_SECURITY** access right. For more information, see SACL Access Right.
SYNCHRONIZE (0x00100000L)	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state.
WRITE_DAC (0x00040000L)	Required to modify the DACL in the security descriptor for the object.
WRITE_OWNER (0x00080000L)	Required to change the owner in the security descriptor for the object.

The following table lists the process-specific access rights.

Expand table

Value	Meaning
PROCESS_ALL_ACCESS (STANDARD_RIGHTS_REQUIRED (0x000F0000L) | SYNCHRONIZE (0x00100000L) | 0xFFFF)	All possible access rights for a process **Server 2003 and Windows XP:** The siz **PROCESS_ALL_ACCESS** flag increased Server 2008 and Windows Vista. If an a compiled for Windows Server 2008 and is run on Windows Server 2003 or Win **PROCESS_ALL_ACCESS** flag is too larg function specifying this flag fails with

	ERROR_ACCESS_DENIED. To avoid this specify the minimum set of access right the operation. If **PROCESS_ALL_ACCES** used, set _WIN32_WINNT to the minim system targeted by your application (fo `#define _WIN32_WINNT _WIN32_WINNT_` more information, see Using the Windo
PROCESS_CREATE_PROCESS (0x0080)	Required to use this process as the par with **PROC_THREAD_ATTRIBUTE_PARI**
PROCESS_CREATE_THREAD (0x0002)	Required to create a thread in the proc
PROCESS_DUP_HANDLE (0x0040)	Required to duplicate a handle using **D**
PROCESS_QUERY_INFORMATION (0x0400)	Required to retrieve certain information process, such as its token, exit code, an (see **OpenProcessToken**).
PROCESS_QUERY_LIMITED_INFORMATION (0x1000)	Required to retrieve certain information process (see **GetExitCodeProcess**, **GetF IsProcessInJob**, **QueryFullProcessImag** handle that has the **PROCESS_QUERY_INFORMATION** acc automatically granted **PROCESS_QUERY_LIMITED_INFORMA** **Server 2003 and Windows XP:** This ac supported.
PROCESS_SET_INFORMATION (0x0200)	Required to set certain information abo such as its priority class (see **SetPriority**
PROCESS_SET_QUOTA (0x0100)	Required to set memory limits using **SetProcessWorkingSetSize**.
PROCESS_SUSPEND_RESUME (0x0800)	Required to suspend or resume a proc
PROCESS_TERMINATE (0x0001)	Required to terminate a process using **TerminateProcess**.
PROCESS_VM_OPERATION (0x0008)	Required to perform an operation on t space of a process (see **VirtualProtectE WriteProcessMemory**).
PROCESS_VM_READ (0x0010)	Required to read memory in a process **ReadProcessMemory**.

PROCESS_VM_WRITE (0x0020)	Required to write to memory in a proc **WriteProcessMemory**.
SYNCHRONIZE (0x00100000L)	Required to wait for the process to ter the wait functions.

To open a handle to another process and obtain full access rights, you must enable the **SeDebugPrivilege** privilege. For more information, see Changing Privileges in a Token.

The handle returned by the **CreateProcess** function has **PROCESS_ALL_ACCESS** access to the process object. When you call the **OpenProcess** function, the system checks the requested access rights against the DACL in the process's security descriptor. When you call the **GetCurrentProcess** function, the system returns a pseudohandle with the maximum access that the DACL allows to the caller.

You can request the **ACCESS_SYSTEM_SECURITY** access right to a process object if you want to read or write the object's SACL. For more information, see Access-Control Lists (ACLs) and SACL Access Right.

> ⚠ **Warning**
>
> A process that has some of the access rights noted here can use them to gain other access rights. For example, if process A has a handle to process B with **PROCESS_DUP_HANDLE** access, it can duplicate the pseudo handle for process B. This creates a handle that has maximum access to process B. For more information on pseudo handles, see GetCurrentProcess.

Protected Processes

Windows Vista introduces *protected processes* to enhance support for Digital Rights Management. The system restricts access to protected processes and the threads of protected processes.

The following standard access rights are not allowed from a process to a protected process:

- **DELETE**
- **READ_CONTROL**
- **WRITE_DAC**
- **WRITE_OWNER**

The following specific access rights are not allowed from a process to a protected process:

- **PROCESS_ALL_ACCESS**
- **PROCESS_CREATE_PROCESS**
- **PROCESS_CREATE_THREAD**
- **PROCESS_DUP_HANDLE**
- **PROCESS_QUERY_INFORMATION**
- **PROCESS_SET_INFORMATION**
- **PROCESS_SET_QUOTA**
- **PROCESS_VM_OPERATION**
- **PROCESS_VM_READ**
- **PROCESS_VM_WRITE**

The **PROCESS_QUERY_LIMITED_INFORMATION** right was introduced to provide access to a subset of the information available through **PROCESS_QUERY_INFORMATION**.

Feedback