**命名管道-官网介绍**

A *named pipe* is a named, one-way or duplex pipe for communication between the pipe server and one or more pipe clients. All instances of a named pipe share the same pipe name, but each instance has its own buffers and handles, and provides a separate conduit for client/server communication. The use of instances enables multiple pipe clients to use the same named pipe simultaneously.

命名管道是用于管道服务器和一个或多个管道客户机之间通信的命名、单向或双工管道。命名管道的所有实例共享相同的管道名称，但每个实例都有自己的缓冲区和句柄，并为客户机/服务器通信提供一个独立的管道。实例的使用允许多个管道客户端同时使用相同的命名管道。

Any process can access named pipes, subject to security checks, making named pipes an easy form of communication between related or unrelated processes.

任何进程都可以通过安全检查访问命名管道，这使得命名管道成为相关或不相关进程之间的一种简单通信形式。

Any process can act as both a server and a client, making peer-to-peer communication possible. As used here, the term pipe server refers to a process that creates a named pipe, and the term pipe client refers to a process that connects to an instance of a named pipe. The server-side function for instantiating a named pipe is [**CreateNamedPipe**](https://docs.microsoft.com/en-us/windows/desktop/api/Winbase/nf-winbase-createnamedpipea). The server-side function for accepting a connection is [**ConnectNamedPipe**](https://msdn.microsoft.com/en-us/library/Aa365146(v=VS.85).aspx). A client process connects to a named pipe by using the [**CreateFile**](https://docs.microsoft.com/windows/desktop/api/fileapi/nf-fileapi-createfilea) or [**CallNamedPipe**](https://docs.microsoft.com/en-us/windows/desktop/api/Winbase/nf-winbase-callnamedpipea) function.

任何进程都可以同时充当服务器和客户机，这使得点对点通信成为可能。这里使用的术语pipe server指的是创建命名管道的进程，而术语pipe client指的是连接到命名管道实例的进程。用于实例化命名管道的服务器端函数是CreateNamedPipe。用于接受连接的服务器端函数是ConnectNamedPipe。客户端进程通过使用CreateFile或CallNamedPipe函数连接到指定的管道。

Named pipes can be used to provide communication between processes on the same computer or between processes on different computers across a network. If the server service is running, all named pipes are accessible remotely. If you intend to use a named pipe locally only, deny access to NT AUTHORITY\NETWORK or switch to local RPC.

命名管道可用于在同一计算机上的进程之间或在网络上不同计算机上的进程之间提供通信。如果服务器服务正在运行，则可以远程访问所有命名管道。如果您打算仅在本地使用命名管道，请拒绝对NT AUTHORITY\NETWORK的访问或切换到本地RPC。

For more information, see the following topics: 有关更多信息，请参阅以下主题:

* [Pipe Names](https://docs.microsoft.com/zh-cn/windows/win32/ipc/pipe-names)

* [Named Pipe Open Modes](https://docs.microsoft.com/zh-cn/windows/win32/ipc/named-pipe-open-modes)
* [Named Pipe Type, Read, and Wait Modes](https://docs.microsoft.com/zh-cn/windows/win32/ipc/named-pipe-type-read-and-wait-modes)
* [Named Pipe Instances](https://docs.microsoft.com/zh-cn/windows/win32/ipc/named-pipe-instances)
* [Named Pipe Operations](https://docs.microsoft.com/zh-cn/windows/win32/ipc/named-pipe-operations)
* [Synchronous and Overlapped Input and Output](https://docs.microsoft.com/zh-cn/windows/win32/ipc/synchronous-and-overlapped-input-and-output)
* [Named Pipe Security and Access Rights](https://docs.microsoft.com/zh-cn/windows/win32/ipc/named-pipe-security-and-access-rights)
* [Impersonating a Named Pipe Client](https://docs.microsoft.com/zh-cn/windows/win32/ipc/impersonating-a-named-pipe-client)
* [Using Pipes](https://docs.microsoft.com/zh-cn/windows/win32/ipc/using-pipes)

# Pipe Names

Each named pipe has a unique name that distinguishes it from other named pipes in the system's list of named objects. A pipe server specifies a name for the pipe when it calls the [**CreateNamedPipe**](https://docs.microsoft.com/en-us/windows/desktop/api/Winbase/nf-winbase-createnamedpipea) function to create one or more instances of a named pipe. Pipe clients specify the pipe name when they call the [**CreateFile**](https://docs.microsoft.com/windows/desktop/api/fileapi/nf-fileapi-createfilea) or [**CallNamedPipe**](https://docs.microsoft.com/en-us/windows/desktop/api/Winbase/nf-winbase-callnamedpipea) function to connect to an instance of the named pipe.

每个命名管道都有一个惟一的名称，该名称将其与系统的命名对象列表中的其他命名管道区分开来。管道服务器在调用CreateNamedPipe函数来创建命名管道的一个或多个实例时，为管道指定一个名称。管道客户端在调用CreateFile或CallNamedPipe函数连接到指定管道的实例时指定管道名称。

Use the following form when specifying the name of a pipe in the [**CreateFile**](https://docs.microsoft.com/windows/desktop/api/fileapi/nf-fileapi-createfilea), [**WaitNamedPipe**](https://docs.microsoft.com/en-us/windows/desktop/api/Winbase/nf-winbase-waitnamedpipea), or [**CallNamedPipe**](https://docs.microsoft.com/en-us/windows/desktop/api/Winbase/nf-winbase-callnamedpipea) function:

\\ServerName\pipe\PipeName

在CreateFile、WaitNamedPipe或CallNamedPipe函数中指定管道名称时，请使用以下形式 : \\ServerName\pipe\PipeName

where ServerName is either the name of a remote computer or a period, to specify the local computer. The pipe name string specified by PipeName can include any character other than a backslash, including numbers and special characters. The entire pipe name string can be up to 256 characters long. Pipe names are not case-sensitive.

其中ServerName是远程计算机的名称或句点，用于指定本地计算机。由PipeName指定的管道名称字符串可以包含除反斜杠以外的任何字符，包括数字和特殊字符。整个管道名称字符串最长可达256个字符。管道名称不区分大小写。

The pipe server cannot create a pipe on another computer, so [**CreateNamedPipe**](https://docs.microsoft.com/en-us/windows/desktop/api/Winbase/nf-winbase-createnamedpipea) must use a period for the server name, as shown in the following example.

[\\.\pipe\PipeName](file:///\\.\pipe\PipeName)

管道服务器不能在另一台计算机上创建管道，因此CreateNamedPipe必须为服务器名使用句点，如下面的示例所示 : [\\.\pipe\PipeName](file:///\\.\pipe\PipeName)

A pipe server can provide the pipe name to its pipe clients, so they can connect to the pipe. The pipe client discovers the pipe name from some persistent source, such as a registry entry, a file, or another application. Otherwise, the clients must know the pipe name at compile time.

管道服务器可以将管道名称提供给它的管道客户机，这样它们就可以连接到管道。管道客户端从一些持久源(如注册表项、文件或其他应用程序)发现管道名称。否则，客户机必须在编译时知道管道名称。

# [Named Pipe Open Modes](https://docs.microsoft.com/zh-cn/windows/win32/ipc/named-pipe-open-modes)

The pipe server specifies the pipe access, overlap, and write-through modes in the dwOpenMode parameter of the [**CreateNamedPipe**](https://docs.microsoft.com/en-us/windows/desktop/api/Winbase/nf-winbase-createnamedpipea) function. The pipe clients can specify these open modes for their pipe handles using the [**CreateFile**](https://docs.microsoft.com/windows/desktop/api/fileapi/nf-fileapi-createfilea) function.

管道服务器在CreateNamedPipe函数的dwOpenMode参数中指定管道访问、重叠和写入通过模式。管道客户机可以使用CreateFile函数为它们的管道句柄指定这些打开模式

## 2.1 Access Mode

Setting the pipe access mode is equivalent to specifying read or write access associated with the pipe server's handles. The following table shows the equivalent generic access right for each access mode you can specify with [**CreateNamedPipe**](https://docs.microsoft.com/en-us/windows/desktop/api/Winbase/nf-winbase-createnamedpipea).

设置管道访问模式相当于指定与管道服务器句柄关联的读或写访问。下表显示了您可以使用CreateNamedPipe指定的每个访问模式的等效通用访问权限。

| **Access mode** | **Equivalent generic access right** |
| --- | --- |
| PIPE\_ACCESS\_INBOUND | GENERIC\_READ |
| PIPE\_ACCESS\_OUTBOUND | GENERIC\_WRITE |
| PIPE\_ACCESS\_DUPLEX | GENERIC\_READ | GENERIC\_WRITE |

If the pipe server creates a pipe with PIPE\_ACCESS\_INBOUND, the pipe is read-only for the pipe server and write-only for the pipe client. If the pipe server creates a pipe with PIPE\_ACCESS\_OUTBOUND, the pipe is write-only for the pipe server and read-only for the pipe client. A pipe created with PIPE\_ACCESS\_DUPLEX is read/write for both the pipe server and the pipe client.

如果管道服务器使用PIPE\_ACCESS\_INBOUND创建管道，则管道服务器的管道是只读的，管道客户机的管道是只读的。如果管道服务器使用PIPE\_ACCESS\_OUTBOUND创建管道，则管道服务器只写管道，管道客户端是只读的。使用PIPE\_ACCESS\_DUPLEX创建的管道用于管道服务器和管道客户机的读/写。

Pipe clients using [**CreateFile**](https://docs.microsoft.com/windows/desktop/api/fileapi/nf-fileapi-createfilea) to connect to a named pipe must specify an access right in the dwDesiredAccess parameter that is compatible with the access mode specified by the pipe server. For example, a client must specify GENERIC\_READ access to open a handle for a pipe that the pipe server created with PIPE\_ACCESS\_OUTBOUND. The access modes must be the same for all instances of a pipe.

To read pipe attributes such as the read mode or blocking mode, the pipe handle must have the FILE\_READ\_ATTRIBUTES access right; to write pipe attributes, the pipe handle must have the FILE\_WRITE\_ATTRIBUTES access right. These access rights can be combined with the generic access right that is appropriate for the pipe: GENERIC\_READ with FILE\_WRITE\_ATTRIBUTES for a read-only pipe, or GENERIC\_WRITE with FILE\_READ\_ATTRIBUTES for a write-only pipe. Restricting access rights in this way provides better security for the pipe.

## 2.2 Overlapped Mode

In overlapped mode, functions performing lengthy read, write, and connect operations can return immediately. This enables the thread to perform other operations while a time-consuming operation is executing in the background. To specify overlapped mode, use the FILE\_FLAG\_OVERLAPPED flag. For more information, see [Synchronous and Overlapped Input and Output](https://docs.microsoft.com/zh-cn/windows/win32/ipc/synchronous-and-overlapped-input-and-output).

The [**CreateFile**](https://docs.microsoft.com/windows/desktop/api/fileapi/nf-fileapi-createfilea) function allows the pipe client to set overlapped mode (FILE\_FLAG\_OVERLAPPED) for its pipe handles using the dwFlagsAndAttributes parameter.

## 2.3 Write-Through Mode

Specify write-through mode with FILE\_FLAG\_WRITE\_THROUGH. This mode affects only write operations to byte-type pipes between pipe clients and pipe servers on different computers. In write-through mode, the functions that write to a named pipe do not return until the data is transmitted across the network and into the pipe's buffer on the remote computer. Write-through mode is useful for applications that require synchronization for every write operation.

If write-through mode is not enabled, the system enhances the efficiency of network operations by buffering data until a minimum number of bytes have accumulated or until a maximum time period has elapsed. Buffering enables the system to combine multiple write operations into a single network transmission. This means that a write operation can be successfully completed after the system puts the data in the outbound buffer, but before the system transmits it across the network.

The [**CreateFile**](https://docs.microsoft.com/windows/desktop/api/fileapi/nf-fileapi-createfilea) function allows the pipe client to set write-through mode (FILE\_FLAG\_WRITE\_THROUGH) for its pipe handles using the dwFlagsAndAttributes parameter. The write-through mode of a pipe handle cannot be changed after the pipe handle has been created. The write-through mode can be different for server and client handles to the same pipe instance.

A pipe client can use the [**SetNamedPipeHandleState**](https://msdn.microsoft.com/en-us/library/Aa365787(v=VS.85).aspx) function to control the number of bytes and the time-out period before transmission for a pipe on which write-through mode is disabled. For a read-only pipe, the pipe handle must be opened with the GENERIC\_READ and FILE\_WRITE\_ATTRIBUTES access rights.

# [Named Pipe Type, Read, and Wait Modes](https://docs.microsoft.com/zh-cn/windows/win32/ipc/named-pipe-type-read-and-wait-modes)

# [Named Pipe Instances](https://docs.microsoft.com/zh-cn/windows/win32/ipc/named-pipe-instances)

# [Named Pipe Operations](https://docs.microsoft.com/zh-cn/windows/win32/ipc/named-pipe-operations)

# [Synchronous and Overlapped Input and Output](https://docs.microsoft.com/zh-cn/windows/win32/ipc/synchronous-and-overlapped-input-and-output)

# [Named Pipe Security and Access Rights](https://docs.microsoft.com/zh-cn/windows/win32/ipc/named-pipe-security-and-access-rights)

# [Impersonating a Named Pipe Client](https://docs.microsoft.com/zh-cn/windows/win32/ipc/impersonating-a-named-pipe-client)

# [Using Pipes](https://docs.microsoft.com/zh-cn/windows/win32/ipc/using-pipes)