

# Streaming Message Transfer

Windows Communication Foundation (WCF) transports support two modes for transferring messages:

- Buffered transfers hold the entire message in a memory buffer until the transfer is complete. A buffered message must be completely delivered before a receiver can read it.
- Streamed transfers expose the message as a stream. The receiver starts processing the message before it is completely delivered.
- Streamed transfers can improve the scalability of a service by eliminating the requirement for large memory buffers. Whether changing the transfer mode improves scalability depends on the size of the messages being transferred. Large message sizes favor using streamed transfers.

By default, the HTTP, TCP/IP, and named pipe transports use buffered transfers. This document describes how to switch these transports from a buffered to streamed transfer mode and the consequences of doing so.

## Enabling Streamed Transfers

Selecting between buffered and streamed transfer modes is done on the binding element of the transport. The binding element has a [TransferMode](#) property that can be set to **Buffered**, **Streamed**, **StreamedRequest**, or **StreamedResponse**. Setting the transfer mode to **Streamed** enables streaming communication in both directions. Setting the transfer mode to **StreamedRequest** or **StreamedResponse** enables streaming communication in the indicated direction only.

The [BasicHttpBinding](#), [NetTcpBinding](#), and [NetNamedPipeBinding](#) bindings expose the [TransferMode](#) property. For other transports, you must create a custom binding to set the transfer mode.

The decision to use either buffered or streamed transfers is a local decision of the endpoint. For HTTP transports, the transfer mode does not propagate across a connection, or to servers and other intermediaries. Setting the transfer mode is not reflected in the description of the service interface. After generating a client class for a service, you must edit the configuration file for services intended to be used with streamed transfers to set the mode. For TCP and named pipe transports, the transfer mode is propagated as a policy assertion.

For code samples, see [How to: Enable Streaming](#).

## Enabling Asynchronous Streaming

To enable asynchronous streaming, add the [DispatcherSynchronizationBehavior](#) endpoint behavior to the service host and set its [AsynchronousSendEnabled](#) property to **true**.

This version of WCF also adds the capability of true asynchronous streaming on the send side. This improves scalability of the service in scenarios where it is streaming messages to multiple clients some of which are slow in reading; possibly due to network congestion or are not reading at all. In these scenarios WCF no longer blocks individual threads on the service per client. This ensures that the service is able to process many more clients thereby improving the scalability of the service.

# Restrictions on Streamed Transfers

Using the streamed transfer mode causes the run time to enforce additional restrictions.

Operations that occur across a streamed transport can have a contract with at most one input or output parameter. That parameter corresponds to the entire body of the message and must be a [Message](#), a derived type of [Stream](#), or an [IXmlSerializable](#) implementation. Having a return value for an operation is equivalent to having an output parameter.

Some WCF features, such as reliable messaging, transactions, and SOAP message-level security, rely on buffering messages for transmissions. Using these features may reduce or eliminate the performance benefits gained by using streaming. To secure a streamed transport, use transport-level security only or use transport-level security plus authentication-only message security.

SOAP headers are always buffered, even when the transfer mode is set to streamed. The headers for a message must not exceed the size of the **MaxBufferSize** transport quota. For more information about this setting, see [Transport Quotas](#).

## Differences Between Buffered and Streamed Transfers

Changing the transfer mode from buffered to streamed also changes the native channel shape of the TCP and named pipe transports. For buffered transfers, the native channel shape is [IDuplexSessionChannel](#). For streamed transfers, the native channels are [IRequestChannel](#) and [IReplyChannel](#). Changing the transfer mode in an existing application that uses these transports directly (that is, not through a service contract) requires changing the expected channel shape for channel factories and listeners.

## See Also

[How to: Enable Streaming](#)

;