# Server side

## Listening for incoming connections

Choose which ports listen for incoming connections using:

Morph.Internet.ListenerManager

Use the various Obtain methods to obtain port listeners. If there is no port listener, then a new one will be created. Use the various Find methods to find existing port listeners.

Control the state of each port listener by using the various Start and Stop methods.

Call RemoveAllInactive() to delete all inactive port listeners.

## Register services

In these examples we publish a service named "Test".

The object to be published is of type TestClass. The implementation of DefaultServletObjectFactory that creates TestClass objects is TestFactory.

1. Registering a shared service. This is overkill, unless you want to do something fancy.

Services.Start("Test", new ApartmentsShared(new TestFactory()));

1. Registering a shared service. We could use the factory that is already provided, which simply returns the given TestClass object. Still overkill, and provides no benefit.

Services.Start("Test", new ApartmentsShared(new DefaultServletObjectFactoryShared(new TestClass())));

1. Registering a shared service. The easiest is to use the shortform. (Identical to example 2.)

Services.Start("Test", new ApartmentsShared(new TestClass()));

1. Registering a sessioned sevice with a 5 minute timeout:

Services.Start("Test", new ApartmentsSession(new TestFactory(), new TimeSpan(0, 5, 0)));

# Client side

## Creating a service proxy

A service proxy (actually apartment proxy) is a local representation for a service. Create a proxy by calling any of these methods:

ApartmentProxy Morph.Endpoint.ApartmentProxy.Via...();

You will need to supply information about:

* Server address
* Service name
* Timeout (not currently used)

For example, accessing a service named "Test" on a server with IP address "test.com":

ApartmentProxy Service = ApartmentProxy.ViaString("test.com", "Test", 600000);

## Obtaining a servlet

To find the servlet you want to call/access, use either:

ServletProxy Servlet = Service.DefaultServlet;

...or...

ServletProxy Servlet = Service.FindServlet(ServletID);

## Calling a servlet method

In these examples we call a method named "Method".

Before calling a servlet method, we must prepare parameters:

object[] InParams = ...;

object[] OutParams;

InParams may be null.

Now the servlet can be called. Here are the options:

1. Call a method (with reply):

object result = Servlet.CallMethod("Method", InParams, out OutParams);

1. Call a method (with no reply):

object result = Servlet.SendMethod("Method", InParams);

## Accessing a servlet property

In these examples we access a property named "Property".

Before accessing a servlet property, we must prepare parameters:

object[] Index = ...;

object Value = ...;

Both may be null.

Now the servlet can be accessed. Here are the options:

1. Get a property value:

Value = Servlet.GetProperty("Property", Index);

1. Set a property value:

Servlet.SetProperty("Property", Value, Index);