**Send messages to MSMQ using ScriptCS**

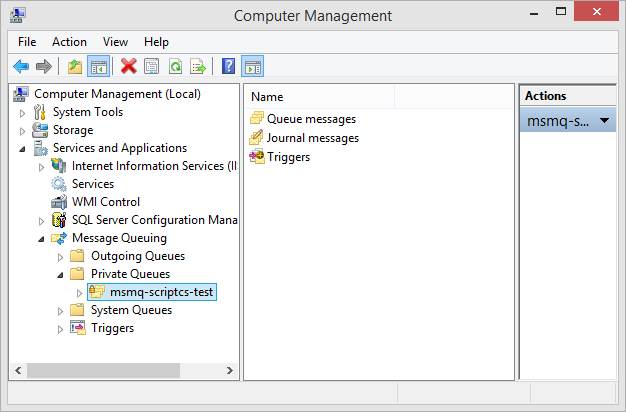
Microsoft Message Queuing [MSMQ] is one of the most commonly used framework for building distributed applications. It provides reliable, secure, optional transaction based messaging model, which guarantees message delivery. Messages can be sent and remain in queue until they processed by message handler. Related messages can be processed using a transaction to ensure they are sent in order, delivered only once and successfully retrieved in destination queue. In case of any failure, entire transaction gets rolled back. While sending the message to the queue, if the message handler goes offline, MSMQ persists the message till the message handler becomes active. MSMQ supports different messaging patterns like fire and forget, request-response, publisher – subscriber etc.

Creating a message based application using MSMQ and Visual studio is relatively easy, as both are Microsoft technologies, however sometimes you need a lightweight tool / framework which allows you to write simple C# program to integrate with .Net framework. Enter [ScriptCS](http://scriptcs.net/).

ScriptCS makes it easy to write and execute C# with a text editor of your choice. That means you can write a C# program using sublime text, notepad or any other text editor out there. Even though Microsoft provides free edition of Visual Studio [community and express editions], a simplest console based project creates too many files you don’t actually need. ScriptCS frees you from Visual Studio without sacrificing the advantages of .Net framework and strongly typed language.

In this article, we will cover how to send and receive messages using MSMQ and ScriptCS. I hope you already know how to setup MSMQ on your Windows machine. If you still need any instructions to set it up, please read [Installing MSMQ](http://msdn.microsoft.com/en-us/library/aa967729(v=vs.110).aspx) article on MSDN before reading further.

Assuming that now you have setup MSMQ on your machine, let’s create our very first message queue - msmq-scriptcs-test as shown in below screenshot. We will send messages to this queue from ScriptCS program.



In order to write the ScriptCS program, you can chose your favorite editor or use ScriptCS REPL. In this article, we will simply use Notepad. Add following code to the Notepad and save the file as msmq.csx. A ScriptCS program recognizes .csx file as a valid C# file and executes the code using Roslyn [C# compiler as a service].

#r System.Messaging;

using System.Messaging;

const string queueAddress = @".\private$\msmq-scriptcs-test";

using (var msmq = new MessageQueue(queueAddress))

{

for (int i = 0 ; i < 10; i++)

{

var message = new Message(i.ToString());

msmq.Send(message);

}

}

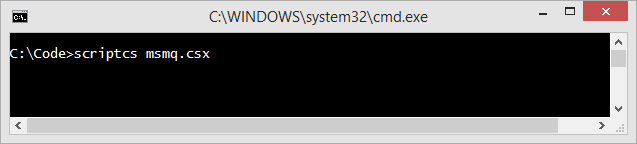
Console.WriteLine("Messages sent successfully");

Console.ReadLine();

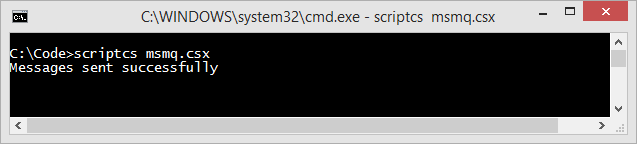
Let’s understand above code first. #r directive allow you to reference .Net assembly from ScriptCS program. In Visual Studio environment, this is same as adding a reference to .Net assembly. The queueAddress string constant defined next refers to the private queue .\private$\msmq-scriptcs-test", created on the local system earlier.

The MessageQueue class in .Net framework makes it really easy to perform common operations like send / receive/ delete messages. In above code snippet we are using Send method to send the messages to MSMQ. In this case we are sending 10 messages to the queue using a simple for loop.

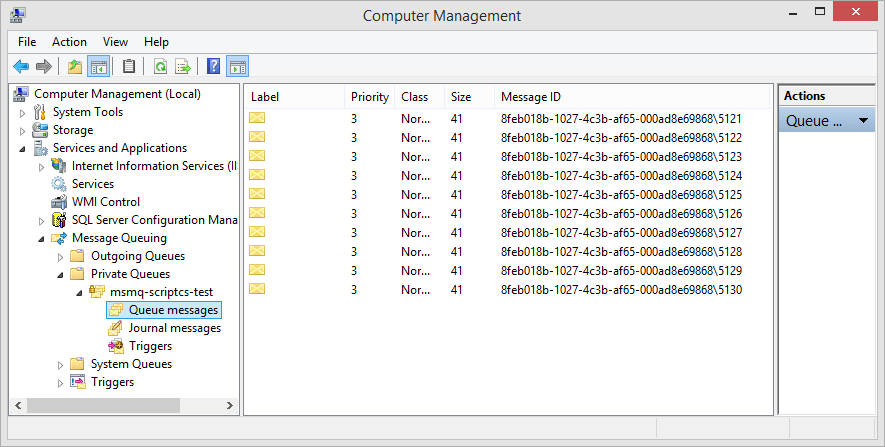
Executing msmq.csx file using scriptcs program is really simple. Just navigate to the folder where the msmq.csx file is saved and call ScriptCS program as shown below.



After successful execution of the program, you should get output message as shown below -



Navigate to the private queue in MSMQ console and it should display all the messages sent from ScriptCS program.



That’s how easy it is to use ScriptCS. In my opinion it’s an amazing tool and every C# developer should know about it. You can practically develop all types of .Net application using ScriptCS and your favorite editor and we have just seen a simple example of it.