

MSMQ

事前準備:

環境:

安裝MSMQ

創建佇列

透過JAVA發送與接收訊息

遠端傳送/接收

遠端使用者權限設置

建立使用者

只能接收及查看訊息及傳送(接收與傳送權限無法個別分開)

事前準備:

環境:

- 1.win10
- 2.Eclipse 版本：2020-06、（2021-12也可）
- 3.jdk版本：java-1.8.0-openjdk-1.8.0.332-1.b09

GitHub - ojdkbuild/ojdkbuild: Community builds using source code from OpenJDK project

Community builds using source code from OpenJDK project - GitHub - ojdkbuild/ojdkbuild:

Community builds using source code from OpenJDK project

 <https://github.com/ojdkbuild/ojdkbuild>

ojdkbuild/ojdkbuild

Community builds using source code from
OpenJDK project

 5 Contributors  0 Issues  3k Stars  361 Forks

4.Visual Studio 2013 的 Visual C++

Download 適用於 Visual Studio 2013 的 Visual C++ 可轉散發套件 from Official Microsoft Download Center

Visual C++ 可轉散發套件會安裝執行階段元件，這些是執行以 Visual Studio 2013 建置的 C++ 應用程式所必備的元件。有關這些可轉散發套裝軟體的更新版本，請參閱 kb 3138367。

 <https://www.microsoft.com/zh-tw/download/confirmation.aspx?id=40784>

5.MSMQMsmqJava-master.zip

GitHub - ikerlan2015/MsmqJava: Port of Msmq to VS2013 with 64 bit support

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

 <https://github.com/ikerlan2015/MsmqJava>

ikerlan-digital/
MsmqJava

Port of Msmq to VS2013 with 64 bit support

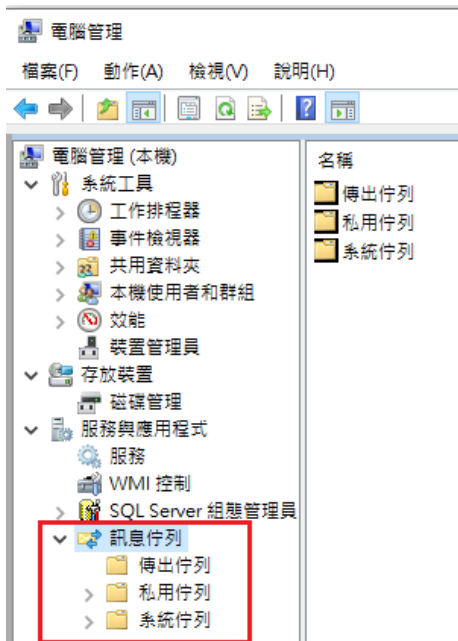
 1 Contributor  4 Issues  11 Stars  9 Forks

ikerlan

參考資料：

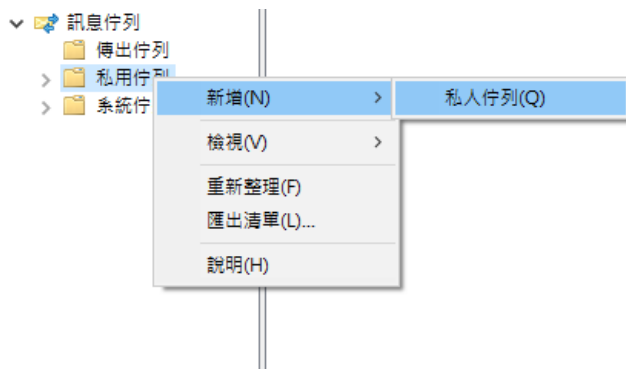
 <https://blog.miniasp.com/post/2010/04/06/MSMQ-Message-Queuing-Learning-Notes>

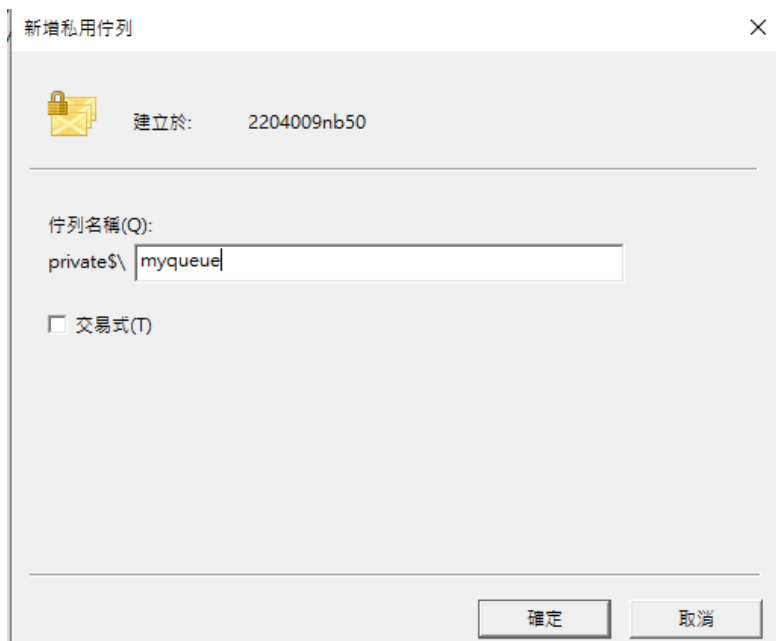




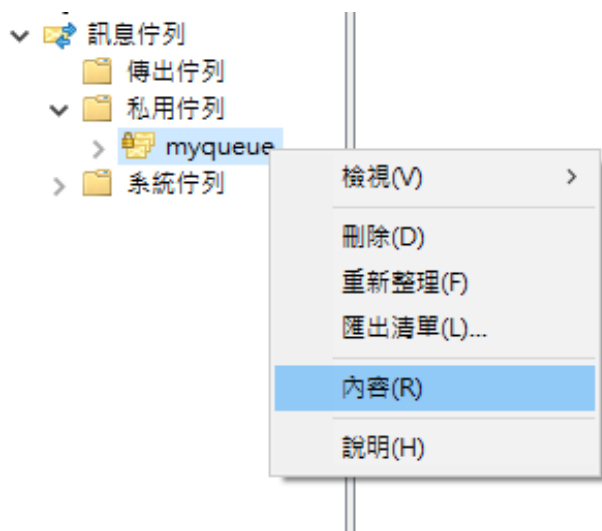
創建佇列

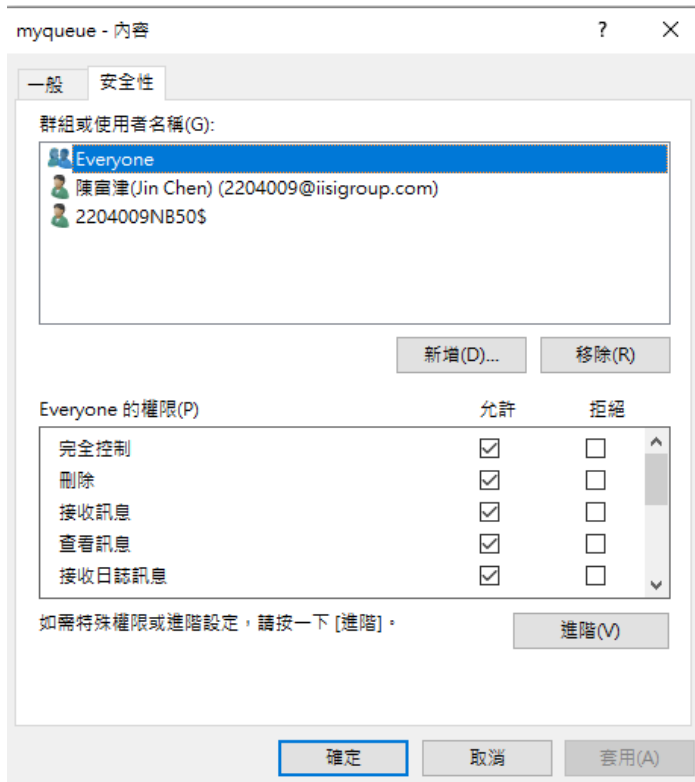
1.透過UI建立(也可透過java建立)





2.在剛建立的佇列上右鍵內容→安全性→勾選全部權限





透過JAVA發送與接收訊息

- 1.必須先進行第一步「安裝MSMQ」
- 2.透過java建立佇列(也可透過介面操作建立)

```
String fullname= ".\\private$\\myqueue";
String qLabel="Created by testtest.java";
boolean transactional= false;
Queue queue= Queue.create(fullname, qLabel, transactional);
```

4.發送訊息

```
package test;

import java.util.UUID;

import ionic.Msmq.Message;
import ionic.Msmq.MessageQueueException;
import ionic.Msmq.Queue;

public class send {

    public static void main(String[] args) throws MessageQueueException {
        try {
```

```

        String fullname = "direct=tcp:192.168.201.221\\private$\\myqueue";
        Queue queue = new Queue(fullname);
        String label;
        String body;
        String correlationID;

        for (int index = 0; index < 10; index++) {
            label = UUID.randomUUID().toString();
            body = String.format("SEND MSG : %s , Datalength : %s", label, label.length());
            correlationID = label;
            Message msg = new Message(body, label, correlationID);
            queue.send(msg);
            System.out.printf("Round :%d, 成功發送  %s\n", index, new String(msg.getBody()));
        }

    } catch (MessageQueueException ex1) {
        ex1.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

5.接收訊息

```

package test;

import ionic.Msmq.Message;
import ionic.Msmq.MessageQueueException;
import ionic.Msmq.Queue;

public class Receive {
    public static void main(String[] args) {
        int index = 0;
        try {

            String fullname = "direct=tcp:192.168.201.221\\private$\\myqueue";
            Queue queue = new Queue(fullname);

            while (true) {
                try {

                    for (int i = 0; i < 1; i++) {
                        Message message = queue.receive();
                        System.out.printf("收到的第 %d筆訊息 : %s\n", ++index, message.getBody());
                    }
                } catch (MessageQueueException e) {
                    e.printStackTrace();
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

        } catch (Exception e) {
            System.out.println("無佇列訊息");
        }
        Thread.sleep(2000);
    }
} catch (MessageQueueException ex1) {
    System.out.println("Put failure:" + ex1.toString());
    ex1.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

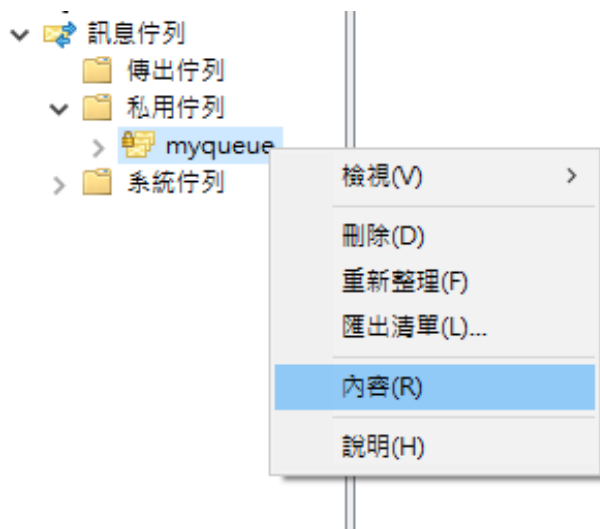
遠端傳送/接收

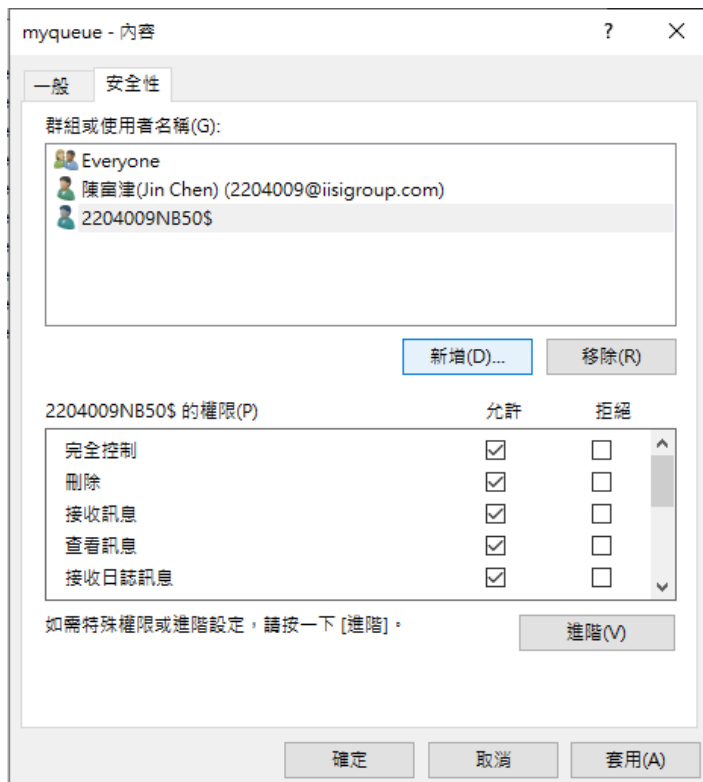
遠端MSMQ對本機MSMQ進行訊息接收與發送。

本機端上操作

由於遠端電腦與本機電腦之間互相並不認識對方，所以就算選擇 **Everyone**也是無效的，而要選取本機電腦的 **ANONYMOUS LOGON**(匿名登入) 使用者才行。

1.在建立的佇列上右鍵內容→安全性→新增使用者

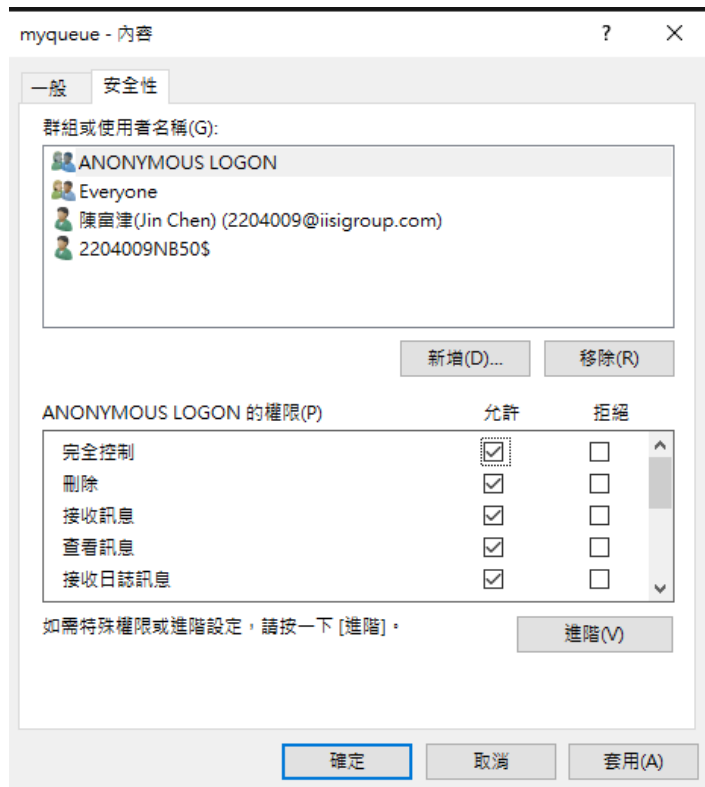




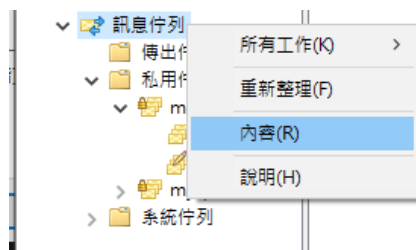
輸入**ANONYMOUS LOGON**後確定

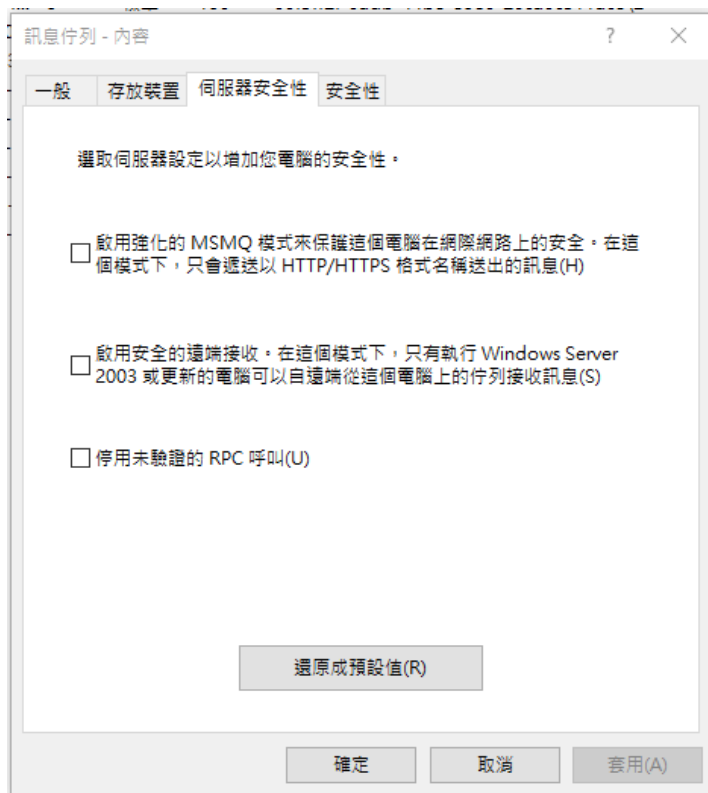


給予全部權限



2.在訊息佇列上右鍵→內容→伺服器安全性→取消勾選三個選項





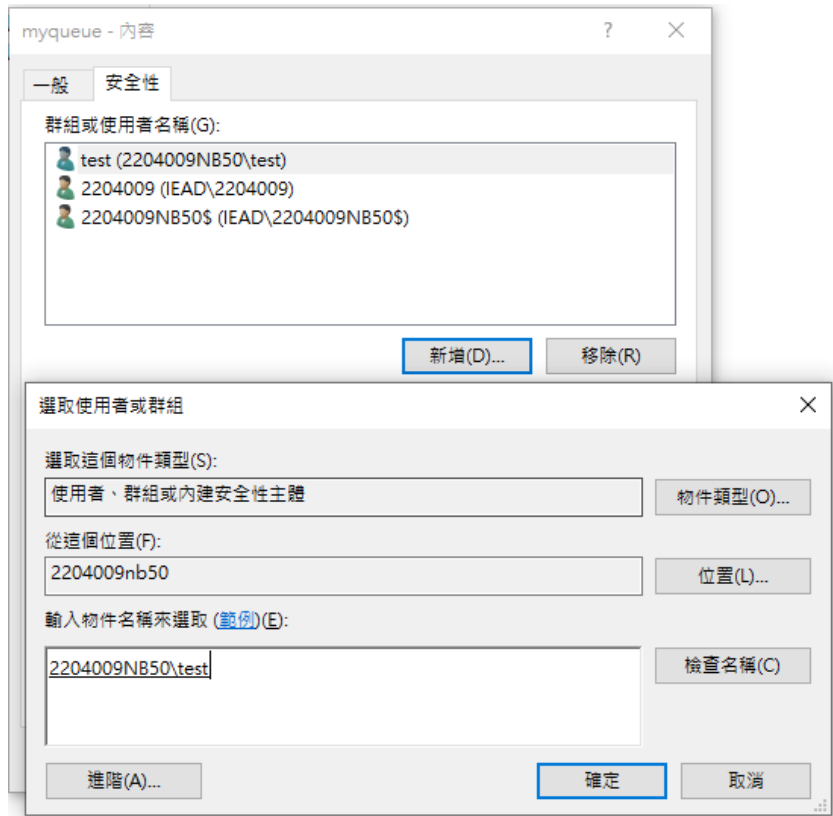
3.程式

遠端執行接收與傳送範例時，修改以下IP位置為本機端IP，即可進行對本機端Msmq發送與接收訊息

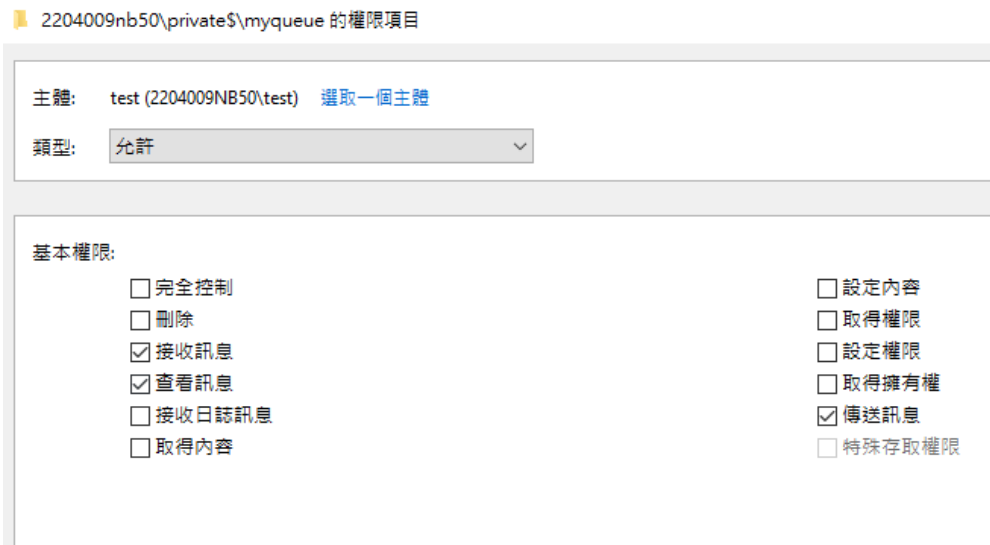
```
String fullname = "direct=tcp:本機IP\\private$\\myqueue";
```

遠端使用者權限設置

▼ 建立使用者

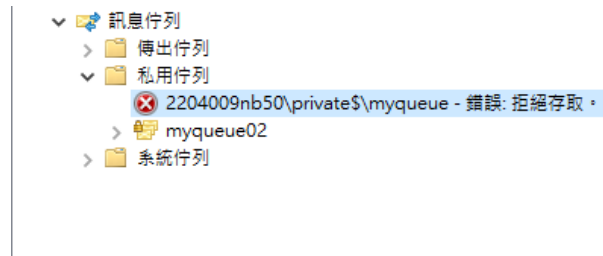


▼ 只能接收及查看訊息及傳送(接收與傳送權限無法個別分開)



▼ 取得內容權限

若未勾選「取得內容」，則該使用者無法查看特定佇列內容



▼ 開啟日誌功能

勾選「已啟用」

