Installazione di Websphere MQ 8, IBM Integration Bus 10

IBM Integration Bus Healthcare Pack 4.0

Policlinico Abano Terme

Versione 2.0

Autore: Cristina Parolari

|  |  |  |  |
| --- | --- | --- | --- |
| Vers. | Autore | Modifica | Data |
| 1.0 | Cristina Parolari | Initial version | 21/04/2017 |
| 2.0 | Andrea Panza | + capitolo configurazione PROD | 22/05/2017 |

Contents

[Azioni preliminari 2](#_Toc473013946)

[Installazione del prodotto IBM Websphere MQ 8.0.0.5 2](#_Toc473013947)

[Installazione di IBM Integration Bus 10.0.0.7 6](#_Toc473013948)

[Configurazione di WebSphere MQ 8 7](#_Toc473013949)

[Configurazione di IBM Integration Bus 10 10](#_Toc473013950)

[Creazione degli integration Node 10](#_Toc473013951)

[Impostazione dell’accesso alla console web e autorizzazioni 11](#_Toc473013952)

[Installazione di IBM Integration Bus Healthcare Pack versione 4 12](#_Toc473013953)

[Installazione di MQExporer su windows 13](#_Toc473013954)

[Riepilogo ambienti 16](#_Toc473013955)

## Azioni preliminari

Sono stati scaricati i seguenti pacchetti di installazione e messi in una cartella di appoggio:

* WS\_MQ\_V8.0.0.4\_LINUX\_ON\_X86\_64\_IM.tar.gz
* 8.0.0-WS-MQ-LinuxX64-FP0005.tar.gz
* IIB\_10.0.0.7\_LINUX\_X86-64.tar.gz
* HealthcarePack\_linux\_amd64\_4.0.0.0.tar.gz

Il pacchetto di installazione di IBM Integration Bus 10 contiene sia il runtime di prodotto che di IBM Integration Toolkit.

Sulle macchine sono stati creati i seguenti utenti e gruppi:

* utente mqm con gruppo primario mqm
* utente iibadmin con gruppo primario mqbrkrs e secondari mqm e iibadmin

## Installazione del prodotto IBM Websphere MQ 8.0.0.5

In tutti gli ambienti le installazioni sono state svolte seguendo questi passi:

* Scompattazione del file WS\_MQ\_V8.0.0.4\_LINUX\_ON\_X86\_64\_IM.tar.gz in una cartella di appoggio (sotto /opt/IBM/software) . In tal modo sono stati otteni i seguenti file :

copyright

crtmqpkg

lap

licenses

mqlicense.sh

MQSeriesAMS-8.0.0-4.x86\_64.rpm

MQSeriesClient-8.0.0-4.x86\_64.rpm

MQSeriesExplorer-8.0.0-4.x86\_64.rpm

MQSeriesFTAgent-8.0.0-4.x86\_64.rpm

MQSeriesFTBase-8.0.0-4.x86\_64.rpm

MQSeriesFTLogger-8.0.0-4.x86\_64.rpm

MQSeriesFTService-8.0.0-4.x86\_64.rpm

MQSeriesFTTools-8.0.0-4.x86\_64.rpm

MQSeriesGSKit-8.0.0-4.x86\_64.rpm

MQSeriesJava-8.0.0-4.x86\_64.rpm

MQSeriesJRE-8.0.0-4.x86\_64.rpm

MQSeriesMan-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_cs-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_de-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_es-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_fr-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_hu-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_it-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_ja-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_ko-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_pl-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_pt-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_ru-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_Zh\_CN-8.0.0-4.x86\_64.rpm

MQSeriesMsg\_Zh\_TW-8.0.0-4.x86\_64.rpm

MQSeriesRuntime-8.0.0-4.x86\_64.rpm

MQSeriesSamples-8.0.0-4.x86\_64.rpm

MQSeriesSDK-8.0.0-4.x86\_64.rpm

MQSeriesServer-8.0.0-4.x86\_64.rpm

MQSeriesXRService-8.0.0-4.x86\_64.rpm

PreReqs

READMES

repackage

1. Con utente root sono stati accettati i termini di licenza tramite il comando **./mqlicence –accept**
2. Lancio del comando rpm per installare i pacchetti strettamente necessari:

rpm -ivh MQSeriesClient-8.0.0-4.x86\_64.rpm

MQSeriesGSKit-8.0.0-4.x86\_64.rpm

MQSeriesJava-8.0.0-4.x86\_64.rpm

MQSeriesJRE-8.0.0-4.x86\_64.rpm

MQSeriesMan-8.0.0-4.x86\_64.rpm

MQSeriesRuntime-8.0.0-4.x86\_64.rpm

MQSeriesSamples-8.0.0-4.x86\_64.rpm

MQSeriesSDK-8.0.0-4.x86\_64.rpm

MQSeriesServer-8.0.0-4.x86\_64.rpm

MQSeriesExplorer-8.0.0-4.x86\_64.rpm

1. Scompattazione del file 8.0.0-WS-MQ-LinuxX64-FP0005.tar.gz in una cartella di appoggio.
2. Installazione del fixpack 5 di WebSphere MQ utilizzando il comando:

rpm -ivh MQSeriesClient-U8005-8.0.0-5.x86\_64.rpm

MQSeriesGSKit-U8005-8.0.0-5.x86\_64.rpm

MQSeriesJava-U8005-8.0.0-5.x86\_64.rpm

MQSeriesJRE-U8005-8.0.0-5.x86\_64.rpm

MQSeriesMan-U8005-8.0.0-5.x86\_64.rpm

MQSeriesRuntime-U8005-8.0.0-5.x86\_64.rpm

MQSeriesSamples-U8005-8.0.0-5.x86\_64.rpm

MQSeriesSDK-U8005-8.0.0-5.x86\_64.rpm

MQSeriesServer-U8005-8.0.0-5.x86\_64.rpm

MQSeriesExplorer-U8005-8.0.0-5.x86\_64.rpm

1. Verifica dei parametri del kernel per l’utente mqm tramite il comando ./mqconfig (lanciato da terminale di mqm)

mqconfig: V3.7 analyzing Ubuntu 14.04.5 LTS settings for WebSphere MQ V8.0

System V Semaphores

semmsl (sem:1) 250 semaphores IBM>=32 PASS

semmns (sem:2) 0 of 32000 semaphores (0%) IBM>=4096 PASS

semopm (sem:3) 32 operations IBM>=32 PASS

semmni (sem:4) 0 of 128 sets (0%) IBM>=128 PASS

System V Shared Memory

shmmax 18446744073692774399 bytes IBM>=268435456 PASS

shmmni 0 of 4096 sets (0%) IBM>=4096 PASS

shmall 0 of 18446744073692774399 pages (0%) IBM>=2097152 PASS

System Settings

file-max 992 of 810367 files (0%) IBM>=524288 PASS

Current User Limits (mqm)

nofile (-Hn) 4096 files IBM>=10240 FAIL

nofile (-Sn) 1024 files IBM>=10240 FAIL

1. Adeguamaneto dei parametri di kernel per gli utenti mqm e iibadmin inserendo le seguenti voci nel file /etc/security/limits.conf

mqm hard nofile 10240

mqm soft nofile 10240

mqm soft nproc 4096

iibadmin hard nofile 10240

iibadmin soft nofile 10240

iibadmin soft nproc 4096

Per applicare le modifiche di /etc/security/limits.conf anche alle sessioni create con "su - <non root user>" bisogna scommentare nel file /etc/pam.d/su la riga

#session required pam\_limits.so

Verifica a posteriori di parametri del kernel per l’utente mqm tramite il comando ./mqconfigmqconfig: V3.7 analyzing Ubuntu 14.04.5 LTS settings for WebSphere MQ V8.0

System V Semaphores

semmsl (sem:1) 250 semaphores IBM>=32 PASS

semmns (sem:2) 28 of 32000 semaphores (0%) IBM>=4096 PASS

semopm (sem:3) 32 operations IBM>=32 PASS

semmni (sem:4) 7 of 128 sets (5%) IBM>=128 PASS

System V Shared Memory

shmmax 18446744073692774399 bytes IBM>=268435456 PASS

shmmni 74 of 4096 sets (1%) IBM>=4096 PASS

shmall 83778 of 18446744073692774399 pages (0%) IBM>=2097152 PASS

System Settings

file-max 2368 of 810367 files (0%) IBM>=524288 PASS

Current User Limits (mqm)

nofile (-Hn) 10240 files IBM>=10240 PASS

nofile (-Sn) 10240 files IBM>=10240 PASS

1. Impostazione a primaria della installazione di MQ appena eseguita. Con terminale di root, lanciare il comando **./setmqinst -i -p /opt/mqm**
2. Verifica che al termine dell’installazione che il contenuto e le proprietà dei file sotto la cartella /opt/mqm sia analogo al seguente:

dr-xr-xr-x  3 mqm  mqm   4096 Jan 23 11:32 bin  
dr-xr-xr-x 16 mqm  mqm   4096 Jan 23 10:34 doc  
dr-xr-xr-x  7 mqm  mqm   4096 Jan 23 10:56 gskit8  
dr-xr-xr-x  4 mqm  mqm   4096 Jan 23 10:56 inc  
-r--r--r--  1 mqm  mqm  13840 May 16  2016 instinfo.tsk  
dr-xr-xr-x  8 mqm  mqm   4096 Oct 17  2015 java  
dr-xr-xr-x  4 mqm  mqm   4096 Jan 23 10:56 lib  
dr-xr-xr-x  3 mqm  mqm   4096 Jan 23 10:56 lib64  
dr-xr-xr-x  2 mqm  mqm   4096 Jan 23 10:55 licenses  
drwxr-xr-x  3 root root  4096 Jan 23 10:55 maintenance  
dr-xr-xr-x  4 mqm  mqm   4096 Jan 23 10:34 man  
dr-xr-xr-x  3 mqm  mqm   4096 Jan 23 11:32 mqexplorer  
-r--r--r--  1 mqm  mqm      0 May 16  2016 mqpatch.dat  
dr-xr-xr-x  3 mqm  mqm   4096 Jan 23 10:33 msg  
dr-xr-xr-x  3 mqm  mqm   4096 Oct 17  2015 properties  
dr-xr-xr-x  2 mqm  mqm   4096 Jan 23 10:55 READMES  
dr-xr-xr-x 12 mqm  mqm   4096 Jan 23 10:56 samp

## Installazione di IBM Integration Bus 10.0.0.7

Osservazione: Il tar di installazione di IBM Integration Bus 10 contiene sia il runtime che IBM Integration Toolkit.

Procedura di installazione del prodotto:

1. A partire dal file IIB\_10.0.0.7\_LINUX\_X86-64.tar.gz estrarre il file **iib-10.0.0.7.tar**.
2. Copiare **iib-10.0.0.7.tar** sotto la cartella **/opt/IBM/mqsi/** precedentemente creata.
3. Installare IIB 10.0.0.7 ed IBM Integration Toolkit, tramite il comando:

**tar -xvf iib-10.0.0.1.tar**

1. Da un terminale di root, accettare la licenza tramite il commando **/opt/IBM/mqsi/iib-10.0.0.2/iib make registry global accept license**.Questo comando determina la creazione della cartella /var/mqsi/ che conterrà i dati del prodotto.
2. Verifica dell’installazione tramite il comando **/opt/IBM/mqsi/iib-10.0.0.2/iib verify all**

Al termine dell’installazione sotto la cartella /opt/IBM/mqsi/iib-10.0.0.7/ abbiamo la seguente struttura:

drwxrwxr-x  9 root root  4096 Jan 23 11:57 common  
drwxrwxr-x  2 root root  4096 Nov 23 22:54 extensions  
drwxr-xr-x  5 root root  4096 Nov 23 23:00 ie02  
-rwxr-xr-x  1 root root 25006 Nov 23 23:00 iib  
drwxrwxr-x  2 root root  4096 Jan 23 11:57 license  
drwxrwxr-x  3 root root  4096 Nov 23 22:54 properties  
drwxrwxr-x 15 root root  4096 Nov 23 23:00 readmes  
drwxrwxr-x 35 root root  4096 Nov 23 22:57 server  
drwxrwxr-x 13 root root  4096 Jan 23 15:15 tools

## Configurazione di WebSphere MQ 8

### DEV e TEST

Premesse:

* Sulla stessa machina sono stati creati due queue manager, configurati in modo similare.
* Essendo queue manager dedicati allo sviluppo ed al test sono stati creati con logging lineare

Passi di configurazione del sistema:

1. creazione dei queue manager

**crtmqm -u DLQ -lp 10 -ls 5 -lf 16384 DEVQM**

**crtmqm -u DLQ -lp 10 -ls 5 -lf 16384 TESTQM**

1. Ciò ha fatto in modo che venisse prodotto un file qm.ini (sotto /var/mqm/qmgrs/NOME\_QUEUE\_MANAGER/) fatto some il seguente:

ExitPath:  
   ExitsDefaultPath=/var/mqm/exits  
   ExitsDefaultPath64=/var/mqm/exits64  
#\*                                                                 \*#  
#\*                                                                 \*#  
Log:  
   LogPrimaryFiles=10  
   LogSecondaryFiles=5  
   LogFilePages=16384  
   LogType=CIRCULAR  
   LogBufferPages=0  
   LogPath=/var/mqm/log/DEVQM/  
   LogWriteIntegrity=TripleWrite  
Service:  
   Name=AuthorizationService  
   EntryPoints=14  
ServiceComponent:  
   Service=AuthorizationService  
   Name=MQSeries.UNIX.auth.service  
   Module=amqzfu  
   ComponentDataSize=0  
TCP:  
   SndBuffSize=0  
   RcvBuffSize=0  
   RcvSndBuffSize=0  
   RcvRcvBuffSize=0  
   ClntSndBuffSize=0  
   ClntRcvBuffSize=0  
   SvrSndBuffSize=0  
   SvrRcvBuffSize=0 SvrSndBuffSize=0

SvrRcvBuffSize=0

Channels:

MaxChannels=1000 ; Maximum number of Channels allowed

MaxActiveChannels=1000 ; Max number of Chan active at any time

SSL:

OCSPAuthentication=OPTIONAL

AllowSSLV3=Y

AllowWeakCipherSpec=NULL\_SHA

1. Creazione delle risorse interne al queue manager DEVQM:

./runmqsc DEVQM

DEFINE QLOCAL(DLQ)

ALTER QMGR CHLAUTH(DISABLED)

ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) CHCKCLNT(REQDADM)

DEFINE LISTENER(MQLISTENER) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)

DEFINE CHANNEL(MQEXPLORER.CHANNEL) CHLTYPE(SVRCONN)

END

1. Creazione delle risorse interne al queue manager TESTQM:

./runmqsc TESTQM

DEFINE QLOCAL(DLQ)

ALTER QMGR CHLAUTH(DISABLED)

ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) CHCKCLNT(REQDADM)

DEFINE LISTENER(MQLISTENER) TRPTYPE(TCP) PORT(1415) CONTROL(QMGR)

DEFINE CHANNEL(MQEXPLORER.CHANNEL) CHLTYPE(SVRCONN)

END

Alla luce di questi comandi a chiunque si agganci al queue manager con utenza amministrativa verrà richiesta una utenza ed una password.

1. Al termine delle configurazioni sotto la cartella /var/mqm avrò la seguente struttura:

drwxrwsr-x 2 mqm  mqm   4096 Jan 23 10:33 config  
drwxrwsr-x 3 mqm  mqm   4096 Jan 23 10:33 conv  
drwxrwsrwx 2 mqm  mqm   4096 Jan 23 11:13 errors  
drwxrwsr-x 3 mqm  mqm   4096 Jan 23 10:33 exits  
drwxrwsr-x 3 mqm  mqm   4096 Jan 23 10:33 exits64  
-rwxrwxrwx 1 root mqm 544040 Jan 23 16:06 JDBC\_XA\_SWITCH\_SPMF3  
drwxrwsr-x 5 mqm  mqm   4096 Jan 23 11:15 log  
drwxr-xr-x 2 mqm  mqm   4096 Jan 20 14:40 logs  
-rw-rw-r-- 1 mqm  mqm    637 Jan 23 10:33 mqclient.ini  
drwxrwsr-x 5 mqm  mqm   4096 Jan 23 10:33 mqft  
-rw-rw-r-- 1 mqm  mqm   2676 Jan 23 11:15 mqs.ini  
drwxrwsr-x 5 mqm  mqm   4096 Jan 23 11:15 qmgrs  
-rw-rw-r-- 1 mqm  mqm   1941 Jan 23 10:33 service.env  
drwxrwsr-x 4 mqm  mqm   4096 Jan 23 10:33 shared  
drwxrwsr-x 5 mqm  mqm   4096 Jan 23 11:15 sockets  
drwxrwsrwx 2 mqm  mqm   4096 Jan 23 10:33 trace

### PRODUZIONE

Obbiettivo:

* Creare un qmeme manager accedibile dai sistemi esterni via canale con sicurezza SSL.
* Nel documento è stato creato un caetificato self signed, in caso di necessità sostituirlo con un certificato dell’Autorità di certificazione;
* logging circolare

Configurazione:

1. creazione dei queue manager

**crtmqm -u ABANOQM.DLQ -lc -lp 10 -ls 20 -lf 16384 ABANOQM**

**strmqm ABANOQM**

1. Verifica

Viene prodotto un path /var/mqm/qmgrs/ABANOQM/ contenente le configurazioni presenti nel file **qm.ini**:

ExitPath:  
   ExitsDefaultPath=/var/mqm/exits  
   ExitsDefaultPath64=/var/mqm/exits64  
#\*                                                                 \*#  
#\*                                                                 \*#  
Log:  
   LogPrimaryFiles=10  
   LogSecondaryFiles=5  
   LogFilePages=16384  
   LogType=CIRCULAR  
   LogBufferPages=0  
   LogPath=/var/mqm/log/ABAONQM/  
   LogWriteIntegrity=TripleWrite  
Service:  
   Name=AuthorizationService  
   EntryPoints=14  
ServiceComponent:  
   Service=AuthorizationService  
   Name=MQSeries.UNIX.auth.service  
   Module=amqzfu  
   ComponentDataSize=0  
TCP:  
   SndBuffSize=0  
   RcvBuffSize=0  
   RcvSndBuffSize=0  
   RcvRcvBuffSize=0  
   ClntSndBuffSize=0  
   ClntRcvBuffSize=0  
   SvrSndBuffSize=0  
   SvrRcvBuffSize=0 SvrSndBuffSize=0

SvrRcvBuffSize=0

Channels:

MaxChannels=1000 ; Maximum number of Channels allowed

MaxActiveChannels=1000 ; Max number of Chan active at any time

SSL:

OCSPAuthentication=OPTIONAL

AllowSSLV3=Y

AllowWeakCipherSpec=NULL\_SHA

1. Creazione delle risorse necessarie al queue manager ABANOQM:
   1. Innanzitutto avviare la console dei comandi WMQ:

runmqsc ABANOQM

* 1. Definire la coda dei messaggi non consegnabili (Dead Letter Queue)

DEFINE QLOCAL(ABANOQM.DLQ)

* 1. Per una prima verifica le autorizzazioni vengono disabilitate, vedremo successivamente come abilitare la sicurezza

ALTER QMGR CHLAUTH(DISABLED)

* 1. Modifica record di autorizzazione

ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) CHCKCLNT(REQDADM)

* 1. Creazione Listener, ossia la definizione della porta di ascolto del queueManager

DEFINE LISTENER(ABANOQM.LISTENER) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)

START LISTENER(ABANOQM.LISTENER)

* 1. Definizione di un canale riservato alle comunicazioni esterne al queue Manager

DEFINE CHANNEL(ABANOQM.SVRCONN) CHLTYPE(SVRCONN)

* 1. Uscita dalla console dei comandi

END

1. Verifica. Creare una connessione ad Queue Manager remoto

Utilizzando MQExplorer connettersi al QueueManager ABANOQM utilizzando il canale di connessione ABANOQM.SVRCONN.

1. Definizione sicurezza SSL

Per gestire i certificati viene utilizzato *runmqckm*, che è incluso nella distribuzione Websphere MQ

* 1. Verifica path del contenitore di chiavi (store) del queue Manager

runmqsc ABANOQM

DISPLAY QMGR SSLKEYR

Risultato

/var/mqm/qmgrs/ABANOQM/ssl/key

END

* 1. Crea il db delle chiavi nel path precedentemente ottenuto

runmqckm -keydb -create -db "/var/mqm/qmgrs/ABANOQM/ssl/key.kdb" -pw password -type cms -expire 3650 –stash



* 1. Crea un certificato selfsigned nello store precedentemente generato

runmqckm -cert -create -db "/var/mqm/qmgrs/ABANOQM/ssl/key.kdb" -pw password -label ibmwebspheremqabanoqm -dn "CN=ABANOQM,OU=test,O=IBM,ST=IT,C=IT" -expire 3650

* 1. Oppure inserisci nello store un certificato generato da un certificatore

runmqckm -cert -add -db "/var/mqm/qmgrs/ABANOQM/ssl/key.kdb" -pw password -label ibmwebspheremqabanoqm -file CA.crt -format ascii

***NOTA: la label del certificato privato di WMQ DEVE ESSERE:***

***ibmwebspheremqabanoqm + [nomeQueueManager]***

* 1. Verifica il contenuto dello store

runmqckm -cert -list -db "/var/mqm/qmgrs/ABANOQM/ssl/key.kdb" -pw password

risultato

ibmwebspheremqabanoqm

* 1. Inserire le chiavi pubbliche dei client per avere una sicurezza da entrambi i lati della connessione.

Nello stesso path contenente lo store copiare il certificato del client

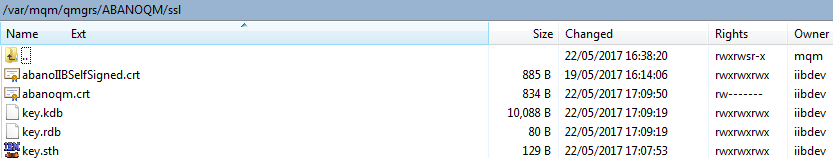
cd /var/mqm/qmgrs/ABANOQM/ssl/

runmqckm -cert -add -db "/var/mqm/qmgrs/ABANOQM/ssl/key.kdb" -pw password -label ibmwebspheremqiibselfsigned -file abanoIIBSelfSigned.crt -format ascii

* 1. Estrai la chiave pubblica del certificato Firmato dal Queue Manager (server)

Questa chiave dovrà essere fornita ai sistemi client che si connetteranno ad MQ

runmqckm -cert -extract -db "/var/mqm/qmgrs/ABANOQM/ssl/key.kdb" -pw password -label ibmwebspheremqabanoqm -target abanoqm.crt -format ascii



***NOTA: il contenitore di chiavi private deve avere generato da “mqm” oppure avere tutte le autorizzazioni di lettura e scrittura.***

1. Abilitazione della sicurezza WMQ

runmqsc ABANOQM

* 1. Eventualmente modifica il path dello store

ALTER QMGR SSLKEYR('/var/mqm/qmgrs/ABANOQM/ssl/key')

* 1. Modifica SSLFIPS

ALTER QMGR SSLFIPS(NO)

* 1. Modifica canale di comunicazione con sistemi client (Server connection)

DEFINE CHANNEL('ABANOQM.SVRCONN') CHLTYPE(SVRCONN) TRPTYPE(TCP) +

SSLCIPH(TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA) SSLCAUTH(REQUIRED) REPLACE

* 1. Configura la sicurezza del canale Server Connection

REFRESH SECURITY TYPE(SSL)

SET CHLAUTH(ABANOQM.SVRCONN) TYPE(ADDRESSMAP) ADDRESS(\*) USERSRC(CHANNEL)

SET CHLAUTH(ABANOQM.SVRCONN) TYPE(BLOCKUSER) USERLIST('nobody')

ALTER QMGR CHLAUTH(ENABLED)

REFRESH SECURITY TYPE(SSL)

REFRESH SECURITY

END

1. Verifica connessione remota (client)

In questo document vengono utilizzate le stesse chiavi e stores generati per la sicurezza IIB (vedi paragrafo successivo).

Gli stores creati sono:

abanoIIB\_keystore.jks per le chiavi private (server)

abanoIIB\_TrustStore.jks per le chiavi pubbliche (client)

* 1. Configurazione stores

runmqckm -cert -add -db "C:\Program Files\IBM\IIB\10.0.0.7\abanoIIB\_TrustStore.jks" -pw password -label ibmwebspheremqabanoqm -file abanoqm.crt -format asci

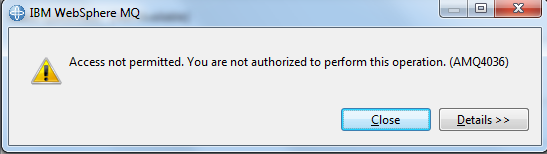
***NOTA: la label del certificato privato di WMQ DEVE ESSERE:***

***ibmwebspheremqabanoqm + [nomeQueueManager]***

runmqckm -cert -list -db "C:\Program Files\IBM\IIB\10.0.0.7\abanoIIB\_TrustStore.jks" -pw password

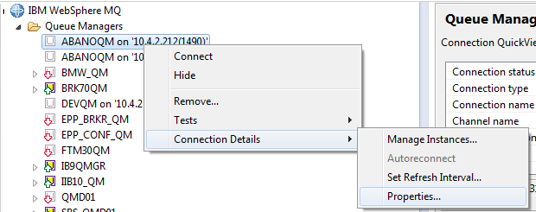
* 1. Configurare MQExplorer

La connessione precedentemente create non sarà più utilizzabile:

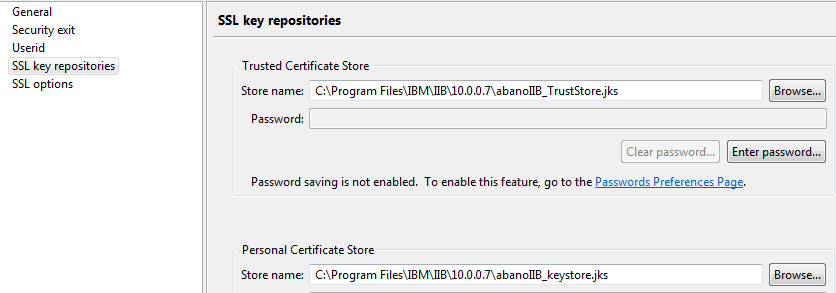


Modificare la connessione.

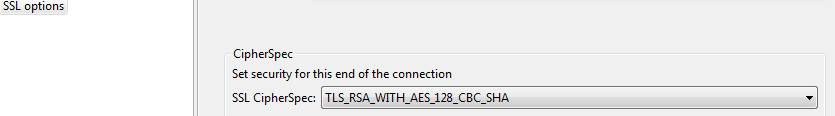
Tasto destro sul QueueManager remoto ABANOQM>>Connection Details>>Properties



Nel tab SSL Key repository inserire il file contenente le chiavi pubbliche del QueueManager (trustStore), e il file contenente le chiavi private del client(keystore)



Nel tab SSL Option inserire il metodo di criptaggio scelto nella definizione della sicurezza del canale di connessione Server Connection



Verificare la connessione

## Configurazione di IBM Integration Bus 10

### Creazione degli integration Node

1. Modifica delle proprietà dei file al di sotto della cartella /var/mqsi con il commando

**chown -R iibadmin:mqbrkrs /var/mqsi**

sotto /var/mqsi i file sono i seguenti:

drwxrwxr-x 7 iibadmin mqbrkrs 4096 Jan 23 16:05 common  
drwxrwsr-x 4 iibadmin mqbrkrs 4096 Jan 23 16:06 components  
drwxrwxr-x 6 iibadmin mqbrkrs 4096 Jan 23 15:26 config  
drwxrwsr-x 3 iibadmin mqbrkrs 4096 Jan 23 12:04 connectors  
drwxrwxr-x 2 iibadmin mqbrkrs 4096 Jan 23 11:57 node\_modules  
drwxrwxr-x 2 iibadmin mqbrkrs 4096 Jan 23 11:57 odbc  
drwxrws--- 4 iibadmin mqbrkrs 4096 Jan 23 16:06 registry  
drwxrwxr-x 2 iibadmin mqbrkrs 4096 Jan 23 11:57 shared-classes  
drwxrwxr-x 3 iibadmin mqbrkrs 4096 Jan 23 11:57 XML  
drwxrwxr-x 3 iibadmin mqbrkrs 4096 Jan 23 11:57 XSL

1. Inserimento nel file ".profile" (vi /home/iibadmin/.profile) dell'utente iibadmin la seguente riga: **. /opt/IBM/mqsi/iib-10.0.0.7/server/bin/mqsiprofile**.

Uscire e rientrare nel terminale con iibadmin per fare prendere l’impostazione. Tale impostazione fa in modo che l’utente iibadmin possa lanciare liberamente i comandi di IBM Integration Toolkit da qualsiasi cartella.

1. Da un terminale di iibadmin lanciare I seguenti comandi per creare l’integration node di sviluppo:

* mqsicreatebroker DEVNODE -q DEVQM //creazione dell’integration node collegato al queue manager
* mqsichangebroker DEVNODE -s active //per abilitazione della sicurezza
* mqsichangebroker DEVNODE -f 10.0.0.7 //impostazione delll’integration node all’uso del del fix
* mqsistart DEVNODE //avvio
* mqsicreateexecutiongroup DEVNODE -e intServer1 //creazione di un integration server
* mqsiwebuseradmin DEVNODE -c -u iibadmin -a iibadmin //creazione di un utente amministrativo
* L’integration node creato risponderà da browser all’indirizzo http://10.4.2.212:4414/

1. Da un terminale di iibadmin lanciare I seguenti comandi per creare l’integration node di sviluppo:

* mqsicreatebroker TESTNODE -q TESTQM //creazione dell’integration node
* mqsichangebroker TESTNODE -s active //per abilitazione della sicurezza
* mqsichangebroker TESTNODE -f 10.0.0.7 //impostazione delll’integration node all’uso del del fix
* mqsistart TESTNODE //avvio
* mqsicreateexecutiongroup TESTNODE -e intServer1 //creazione di un integration server
* mqsiwebuseradmin TESTNODE -c -u iibadmin -a iibadmin //creazione di un utente amministrativo
* L’integration node creato risponderà da browser all’indirizzo http://10.4.2.212:4415/

### Impostazione dell’accesso alla console web e autorizzazioni

Osservazioni:

* La console Web di amministrazione del BROKER ha URL http://nome\_host:4414/
* IBM integration bus ha visibilità delle utenze di sistema operativo tramite mq, ma poterle abilitare all’accesso alla console web occorre ricensirle tramite un comando analogo al seguente mqsiwebuseradmin DEVNODE -c -u nome\_utente -a password\_utente.
* La password specificata non è necessariamente quella di sistema operativo, ma è una password interna alle configurazioni dell’integration node creato
* Per la verifica delle autorizzazioni nell’accesso alla console web, siccome gli integration node creati sono agganciati ad un queue manager, occorre dare delle opportune autorizzazioni sulle seguenti code MQ:
  + SYSTEM.BROKER.AUTH -> per le autorizzazioni al nodo
  + SYSTEM.BROKER.DC.AUTH -> per le autorizzazioni al data capture
  + SYSTEM.BROKER.AUTH.IntegrationServer ->per le autorizzazioni allo specifico integration server
* Per dare autorizzazioni piene su un integration node occorre dare alcune autorizzazioni sulle code autorizzative del queue manager associato:

setmqaut -m NOME\_QM -t q -n 'SYSTEM.BROKER.DC.AUTH' -g NOME\_GRUPPO -all +inq +dsp +set

setmqaut -m NOME\_QM -t q -n 'SYSTEM.BROKER.AUTH' -g NOME\_GRUPPO -all +inq +dsp +set + put

setmqaut -m NOME\_QM -t q -n 'SYSTEM.BROKER.AUTH.IntServer' -g NOME\_GRUPPO –all +inq +dsp +set +put

* Per dare autorizzazioni di visualizzazione si possono lanciare i seguenti comandi:

setmqaut -m NOME\_QM -t q -n 'SYSTEM.BROKER.DC.AUTH' -g NOME\_GRUPPO -all +inq +dsp +set

setmqaut -m NOME\_QM -t q -n 'SYSTEM.BROKER.AUTH' -g NOME\_GRUPPO -all +inq +dsp +set

setmqaut -m NOME\_QM -t q -n 'SYSTEM.BROKER.AUTH.IntServer' -g NOME\_GRUPPO –all +inq +dsp +set

Nello specifico dell’ambiente del Policlinico di Abano, per l’utente iibadmin non è stato necesario lanciare ulteriori comandi autorizzativi poichè iibadmin fa parte del gruppo mqm ed è già amministratore pieno delle risorse di MQ ed è pertanto già pienamente autorizzato alle code SYSTEM.BROBER.\*.

### Configurazione sicurezza SSL nel layer di integrazione

#### Scenario

Definire in IBM Integration Bus del Policlinico di Abano (Policlinico) una comunicazione sicura con un certificato SSL.

Lo strato di Integrazione del Policlinico:

* Espone servizi REST (REST API) che vengono utilizzati da:
  + IBM Mobile First Platform(MFP)
  + IBM Business Process Manager
* Espone servizi HTTP (Healthcare)
  + MEDArchiver tramite protocollo HL7
* Richiama servizi REST esposti da:
  + IBM Business Process Manager;
  + Policlinico
* Legge su WMQ Locale messaggi
  + IBM Business Process Manager remoto

#### Soluzione

Verranno definiti due contenitori di chiavi comuni per tutta l’interfaccia IIB.

* Keystore: certificati proprietari di IIB (server)
* Truststore chiavi pubbliche di certificati di altri provider (client)

Configurare Il canale di comunicazione Server WMQ utilizzando un certificato contenuti nel Keystore.

#### Configurazione

La configurazione è una configurazione di test con certificati self signed, non approvati dalle autorità di certificazione.

La gestione dei certificati è fatta utilizzando gli opportuni contenitori:

keystore Contiene la chiave che certifica i servizi HTTP esposti.

Truststore Contiene i certificati pubblici utilizzati per chiamare altri servizi.

##### Generazione Keystore e Truststore.

Viene utilizzato keytool da linea comando per generare i contenitori (store), generare ed importare certificati.

###### Generare KeyStore

keytool -genkey -keyalg RSA -alias abanoIIBSelfSigned -keystore abanoIIB\_keystore.jks -storepass password -validity 3650 -keysize 2048

###### Esporta chiave pubblica certificate da fornire ai client:

keytool -export -alias abanoIIBSelfSigned -file abanoIIBSelfSigned.crt -keystore abanoIIB\_keystore.jks

###### Generare truststore e importare un certificate:

keytool -importcert -keystore abanoIIB\_TrustStore.jks -alias abanoIIBSelfSigned -storepass password -file abanoIIBSelfSigned.crt

###### Aggiungere chiavi pubblice di BPM

keytool -import -trustcacerts -file abanoIIBSelfSigned.crt-alias abanoIIBSelfSigned -keystore abanoIIB\_TrustStore.jks

###### UTILITY

Lista di certificate in a key/trust Store

keytool -list -v -keystore abanoDEV.jks

delete a certificate

keytool -delete -alias abanoIIBSelfSigned -keystore abanoIIBTrustStore.jks

##### Configurazione IIB

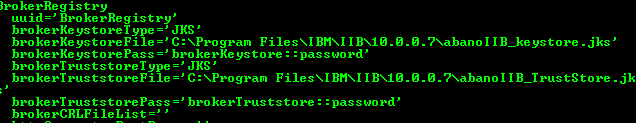
Si assume:

Integration Node IIB10ABANO\_PROD

Integration Server HL7Server

###### Configurare Integration NODE (broker)

Verifica dei contenitori di chiavi.

mqsireportproperties IIB10ABANO\_PROD -o BrokerRegistry -a 

Modifica puntamento a KeyStore

mqsichangeproperties IIB10ABANO\_PROD -o BrokerRegistry -n brokerKeystoreFile -v /var/mqsi/abanoIIB\_keystore.jks

Modifica puntamento a TrustStore

mqsichangeproperties IIB10ABANO\_PROD -o BrokerRegistry -n brokerTruststoreFile -v /var/mqsi/abanoIIB\_truststore.jks

Definizione password dei contenitori

mqsisetdbparms IIB10ABANO\_PROD -n brokerKeystore::password -u temp -p password

mqsisetdbparms IIB10ABANO\_PROD -n brokerTruststore::password -u temp -p password

Utility.

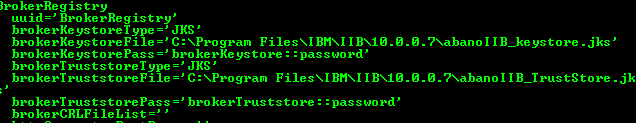
Aggiornamento del puntamento alla proprietà contenente la password

mqsichangeproperties IIB10ABANO\_PROD -o BrokerRegistry -n brokerKeystorePass -v brokerKeystore::password

mqsichangeproperties IIB10ABANO\_PROD -o BrokerRegistry -n brokerTruststorePass -v brokerKeystore::password

Verifica finale

mqsireportproperties IIB10ABANO\_PROD -o BrokerRegistry -a



Configurazione per uso SSL in HTTP Listener

mqsichangeproperties IIB10ABANO\_PROD -b httplistener -o HTTPListener -n enableSSLConnector -v true

Modifica Porta per HTTP ed HTTPS. In caso sia necessario distinguere un Integration Node da un altro presente sullo stesso server.

Default 7083.

mqsichangeproperties IIB10ABANO\_PROD -b httplistener -o HTTPConnector -n port -v 7090

mqsichangeproperties IIB10ABANO\_PROD -b httplistener -o HTTPSConnector -n port -v 7093

Attivare le chiamate HTTPS e disattivare le chiamate HTTP da browser

*mqsichangeproperties IIB10ABANO\_PROD -b httplistener -o HTTPConnector -n corsEnabled -v false*

*mqsichangeproperties IIB10ABANO\_PROD -b httplistener -o HTTPSConnector -n corsEnabled -v true*

Permettere accesso applicazioni client solo HTTPS. Default false

Verifica utilizzo di HTTPListener a livello di Integration Node. Default true.

In questo caso viene utilizzato un solo HTTP server per Integration Node.

mqsireportproperties IIB10ABANO\_PROD -b httplistener -o HTTPListener –a

per modificare

mqsichangeproperties IIB10ABANO\_PROD -b httplistener -o HTTPListener -n startListener -v true

Verifica finale

proprietà navigabili per httplistener

mqsireportproperties *IIB10ABANO\_PROD*-b httplistener -o AllReportableEntityNames –a

proprietà HTTPListener

mqsireportproperties *IIB10ABANO\_PROD*-b httplistener -o HTTPListener -a

Proprietà HTTP

mqsireportproperties *IIB10ABANO\_PROD*-b httplistener -o HTTPSConnector -a

Proprietà HTTPS

mqsireportproperties *IIB10ABANO\_PROD*-b httplistener -o HTTPSConnector -a

###### Configurare Integration server (execution group)

Generare un Integration server:

mqsicreateexecutiongroup IIB10ABANO\_PROD -e hl7Server

mqsichangeproperties IIB10ABANO\_PROD -e hl7Server –o HTTPSConnector –n ClientAuth –v true

A questo punto I flussi di integrazione possono essere rilasciati nell’integration server e verranno espoeti con sicurezza HTTPS.

## Installazione di IBM Integration Bus Healthcare Pack versione 4

Dopo avere installato IBM WebSphere MQ 8, IBM Integration Bus 10 e creati queue manager ed integration node occorre procedere alla installazione di IBM Integration Bus Healthcare Pack ed alla configurazione degli integration node e di IBM Integration Toolkit affinchè ne facciano uso.

Per l’installazione di IBM Integration Bus Health Pack 4 si è fatto riferimento alla guida posta al sito <http://www.ibm.com/support/knowledgecenter/en/SSMKHH_10.0.0/com.ibm.healthcare.doc/ha03110.htm>

E’ stato scaricato, scompattato e copiato il software HealthcarePack\_linux\_amd64\_4.0.0.0.tar.gz sotto la cartella /opt/IBM.

1. Per installare è stato lanciato il comando :

* tar -xzvf HealthcarePack\_linux\_amd64\_4.0.0.0.tar.gz

Ciò ha creato la cartella /opt/IBM/HealthcarePack

1. Per accettare ed attivare la licenza:

* cd /opt/IBM/HealthcarePack/4.0.0.0
* ./HealthcarePack\_install.sh /opt/IBM/mqsi/iib-10.0.0.7/ accept license silently

L’installazione di Health Care Pack ha di fatto modificato l’installazione di IIB aggiungendone alcune estensioni. Da questo momento in poi tutte le volte che verrà lanciato il comando mqsiprofile (ad esempio al login dell’utente iibadmin) è corretto che vengano mostrati i messaggi:

* mqsiprofile extended for Healthcare Pack Web-based Monitoring
* mqsiprofile extended for Healthcare Pack MedicalDeviceInput node

1. E’ stato poi necessario configurare gli integration node affinchè fossero abilitati all’uso di Healthcare Pack. Per fare questo sono stati lanciati i seguenti comandi:

* avvio di queue manager e integration node
* mqsimode DEVNODE -x healthcare,medicalDevices
* mqsimode TESTNODE -x healthcare,medicalDevices
* riavvio di queue manager e integration node

Ripulitura di IBM Integration Toolkit affichè avesse visibilità dei nodi specifici del pacchetto. Con utente iibadmin lanciare:

* cd /opt/IBM/mqsi/iib-10.0.0.7/
* ./iib toolkit –clean

All’avvio di IBM Integration Toolkit sono visibili nella paletta di sviluppo i nodi specifici del pacchetto.

Per la verifica della installazione è stato seguito con esito positivo quanto riportato al link <http://www.ibm.com/support/knowledgecenter/en/SSMKHH_10.0.0/com.ibm.healthcare.doc/ha03040.htm>

Installazione di MQExporer su windows

Sulla macchina del sistemista si è proceduto alla installazione di MQ Explorer in modo che fosse posibile l’amministrazione dei queue manager da remoto.

Per fare questo si è scaricato il support pack MS07 dal link <http://www-01.ibm.com/support/docview.wss?uid=swg24021041>.

L’installazione su windows è banale poichè consiste in un wizard che chiede l’accettazione dei termini di licenza e propone la scelta della directory di installazione (nel nostro caso si è scelto C:\IBM\MQExplorer).

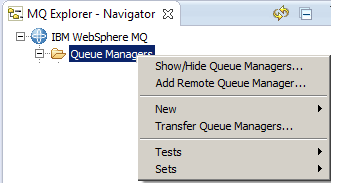
E’ stato necessario cambiare l’impostazione della lingua, in modo che MQExplorer funzionasse in inglese. Per fare questo è stato modificato il file C:\IBM\MQExplorer\eclipse\configuration\config.ini aggiungendo in fondo al file le seguenti righe:

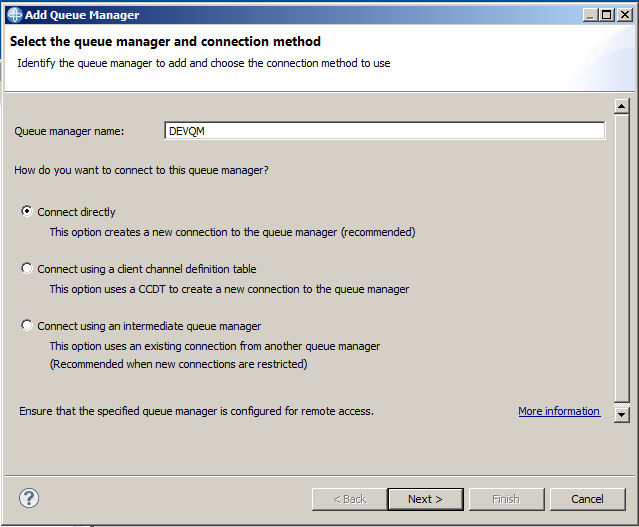
* org.osgi.framework.language=en
* osgi.nl=en\_US

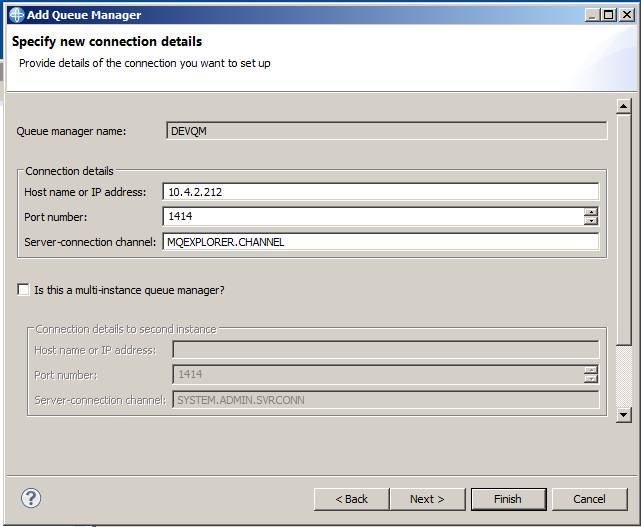
Al riavvio di MQ Explorer la lingua mostrata era quella inglese

Effettuata l’installazione è stato necessario agganciare i queue manager remoti nel seguente modo:

Selezionare Add Remote Queue Manager







* Non abilitare la security exit.
* Sul pannello successivo specificare l’utente mqm impostando la richiesta della password all’atto della connessione.
* Sui pannelli successivi lasciare i default fino alla fine del wizard.

Al termine del wizard il queue manager risultava correttamente connesso ed era pertanto amministrabile da remoto.

## Riepilogo ambienti

Utenti:

* mqm amministratore di mq
* iibadmin amministratore di mq e ibm integration bus

Queue Managers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome Queue Manager | Nome Canale SVRCONN | Secure | Ip | Porta |
| DEVQM | MQEXPLORER.CHANNEL | No | 10.4.2.212 | 1414 |
| TESTQM | MQEXPLORER.CHANNEL | No | 10.4.2.212 | 1415 |
| ABANOQM | ABANOQM.SVRCONN | SSL | 10.4.2.213 | 1414 |

Integration Nodes

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Integration Node | Porta Admin | Ip | URL Web Console |
| DEVNODE | 4414 | 10.4.2.212 | http://10.4.2.212:4414/ |
| TESTNODE | 4415 | 10.4.2.212 | http://10.4.2.212:4415/ |
| IIB10ABANO\_PROD | 4414 | 10.4.2.213 | http://10.4.2.212:4413/ |

Contenitori di chiavi IIB

|  |  |
| --- | --- |
| /var/mqsi/abanoIIB\_keystore.jks | Keystore |
| /var/mqsi/abanoIIB\_truststore.jks | TrustStore |

Comandi di MQ, sotto la cartella /opt/mqm/bin con utente mqm o iibadmin:

|  |  |
| --- | --- |
| strmqm NOMEQM | Avvio |
| endmqm -w NOMEQM | Spegnimento |
| runmqsc NOMEQM | Apertura console amministrativa command line |
| strmqcfg | Avvio di MQExporer (console amministrativa grafica) |

Comandi di IIB, con utente iibadmin (sono già nel path):

|  |  |
| --- | --- |
| mqsistart NOMENODO | Avvio |
| mqsistop NOMENODO | Spegnimento |
| mqsilist | Visualizzazione stato di tutti gli integration node sulla macchina |
| mqsilist NOMENODO -r | Visualizzazione stato di un nodo, dei suoi integration server e delle applicazioni installate |

Dalla cartella /opt/IBM/mqsi/iib10.0.0.7/

Comandoi **iib toolkit** per fare avviare lo strumento di sviluppo IBM Integration Toolkit.