



TEMENOS™

T24 Java deployment in WebSphere v7- v8 standalone

R14/R15

10/3/2015

Temenos



Information in this document is subject to change without notice.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TEMENOS Holdings NV.


Amendment History:

Revisio n	Date Amended	Name	Description
11	1 st April 2011	TAFJ team	Initial version
12	7 st February 2012	H. Aubert	R12GA review
13	15 January 2013	JN.Charpin	R13GA review
14	5 th Feb. 2014	R. Vincent	R14GA review
15	13 th March 13, 2014	JN.Charpin	R14GA review and additional script parameters
16	19 March 2014	JN. Charpin	Servlet review
17	7 th July 2014	R. Vincent	Add appendix, Browser with remote WAS
18	5 February 2015	JN. Charpin	Memory and JVM settings for UAT and Production env.
19	6 th March 2015	H. Aubert	R15 AMR review
20	13 th March 2015	G.Gowrimani	MSSQL datasource details and MQ details

Copyright

Copyright (c) 2014 TEMENOS HOLDINGS NV
All rights reserved.

This document contains proprietary information that is protected by copyright. No part of this document may be reproduced, transmitted, or made available directly or indirectly to a third party without the express written agreement of TEMENOS UK Limited. Receipt of this material directly from TEMENOS UK Limited constitutes its express permission to copy. Permission to use or copy this document expressly excludes modifying it for any purpose, or using it to create a derivative therefrom.

Errata and Comments

If you have any comments regarding this manual or wish to report any errors in the documentation, please document them and send them to the address below:
Technology Department

Temenos Headquarters SA
2 Rue de l'Ecole-de-Chimie,
CH - 1205 Geneva,
Switzerland

Tel SB: +41 (0) 22 708 1150
Fax: +41 (0) 22 708 1160

Please include your name, company, address, and telephone and fax numbers, and email address if applicable. TAFJdev@temenos.com

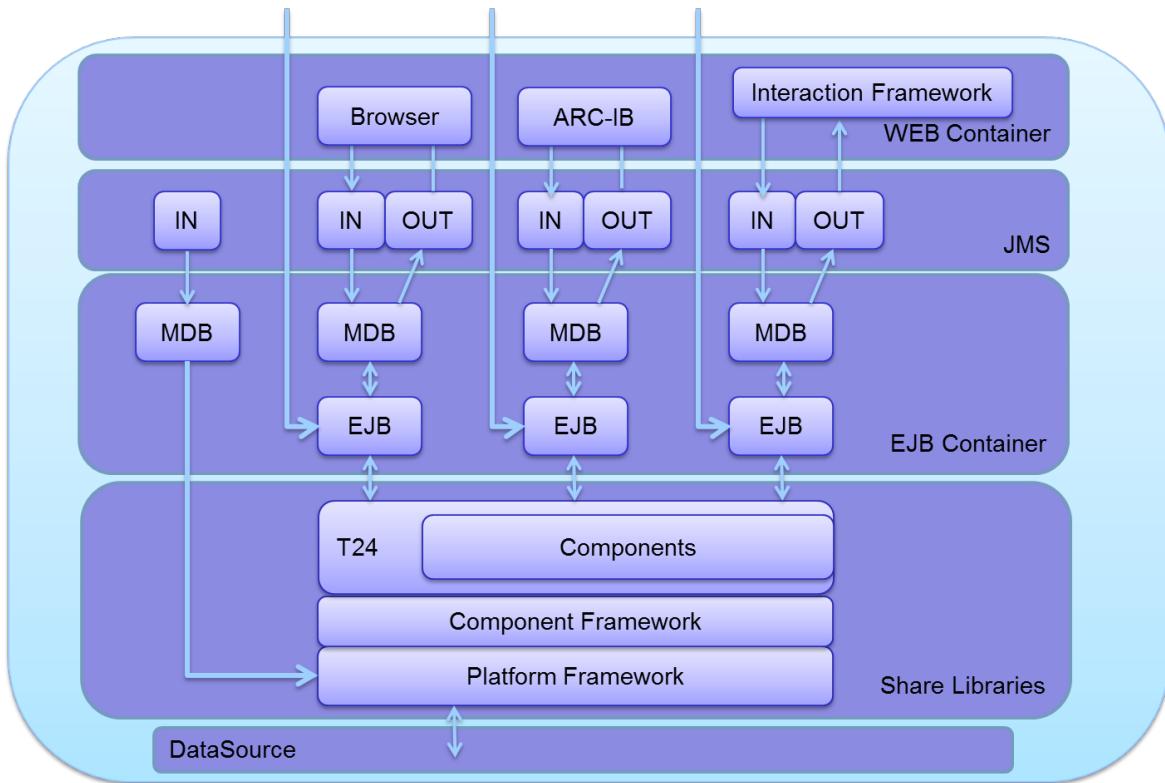
Table of Contents

Copyright.....	3
Errata and Comments.....	3
T24 Java deployment in WebSphere v7- v8 standalone.....	6
Infrastructure.....	6
WebSphere v7- v8 standalone installation.....	7
Application Server installation.....	7
Prerequisite.....	7
WebSphere v7 – v8 configuration for T24 solution with TAFJ.....	8
Script Mode.....	8
Creating a profile.....	8
Prerequisites.....	8
One step configuration.....	10
One step at a time configuration.....	15
After the deployment.....	34
TAFJ servlets.....	36
tDiag Servlet.....	37
tShow Servlet.....	38
Execute Servlet.....	39
Other TAFJEE functionalities.....	39
T24 Application deployment - BrowserWeb.war.....	40
Console mode.....	40
Websphere Properties for T24/TAFJ.....	43
Setup Multiple Servants (zOS).....	43
Set Data Source mutithread for DB2 (Only needed for XA Driver).....	44
Set Data Source Isolation for DB2.....	45
Kill clean thread of Appserv.....	46
Remove end point delivery message.....	47
Auto Reconnection database (Only needed for XA Driver).....	47
Validate existing pooled connections (optional).....	47

SETUP JInsight (not recommended for production use as this is a profile tool).....	48
SETUP Support Assistant (not recommended for production use as this is a profile tool).	48
Setup Load Balance in Queue with Multiple Servant.....	48
Avoid the thread cleaning of Websphere and avoid the disconnection of JDBC.....	50
Websphere Performance Properties.....	52
UAT / Production Memory setting.....	52
Set Performance Improvements for zOS.....	53
Statement Cache Size.....	55
Stored Procedure Called for Duplicate Key.....	55
Data sources > DB2 Universal JDBC Driver DataSource > Custom properties.....	55
Trouble Shooting.....	56
TAFJServices.war.....	56
Where is my Browser port?.....	58
Websphere is killing threads because the T24 code has a sleep in it.....	58
Websphere has J2CA0081E: Method cleanup failed while trying to execute method cleanup on ManagedConnection.....	58
Where is my Default Messaging port for the connection factory for T24Bus?.....	60
Tools.....	61
Debugging WAS through remote Eclipse.....	61
Health Center.....	61
JMX connection setup zOS.....	62
Appendix.....	63
Connect Browser to a Remote WAS.....	63
MQ manager and related object creation.....	63

T24 Java deployment in WebSphere v7- v8 standalone

Infrastructure





WebSphere v7- v8 standalone installation

Application Server installation.

For detailed information please refers to IBM Websphere official documentation.

http://pic.dhe.ibm.com/infocenter/rsahelp/v8/index.jsp?topic=%2Fcom.ibm.xtools.installation.rsaws.doc%2Ftopics%2Ft_install_was8.html

note : Install Websphere with IBM SDK 1.7.

Prerequisite

TAFJ has to be installed in a folder that we call in the document **TAFJ_HOME**.

T24 precompiled jar files have to be in a folder that we call in the document **T24_HOME**.

Before configure JBoss with TAFJ/T24, check your standalone installation is working.

The screenshot shows a Windows Command Prompt window titled "Command Prompt - tRun EX". The title bar also displays "GLOBUS Rev. 201502" and "SIGN.ON". The window content includes a copyright notice "Copyright (c) Temenos Systems Ltd 2015". Below this is a large block of dollar sign (\$) characters forming a grid pattern. At the bottom of the window, there is a log entry: "06 MAR 2015 17:59:20 USER [4145,] ACTION PLEASE ENTER YOUR SIGN ON NAME".



WebSphere v7 – v8 configuration for T24 solution with TAFJ

You can configure the following websphere resources either by using the administrative console or by running jython scripts. This document only covers script mode. For console mode, please refer to Websphere documentation.

Script Mode

The configuration can be run automatically on your server by using some jython scripts.

Note: These scripts can't be used on an existing configuration. To create a new configuration from scratch, first create a new profile and then run the configuration.

Creating a profile

You can easily create a new profile by using the tool **%TAFJ_HOME%\appserver\was\createStandalone.bat** .

Prerequisites

To run a jython script your server must be started and your PATH must contain:

- **%WAS_HOME%\bin** : the path to your websphere\bin directory
- **%TAFJ_HOME%\appserver\was\jython** : the path to the jython files

Edit the file **setUpEnv** and set the following properties i.e. :

```
set WAS_HOME= C:\IBM\WebSphere\AppServer
set HOST_NAME=localhost
set PROFILE_NAME=AppSrv01
set NODE_NAME=%PROFILE_NAME%Node01
set CELL_NAME=%NODE_NAME%Cell
set SERVER_NAME=server1
```

You can then launch **createStandalone.bat** and the targeted profile will be created in your **WAS_HOME\profiles** directory.

Edit the file **Create createStandalone** and choose the param **enableAdminSecurity** to add at the end of the command to the scripts:

```
-enableAdminSecurity false
Or
-enableAdminSecurity true -adminUserName tafj -adminPassword
tafj
```



Launch **createStandalone**

```
C:\Administrator:tah_Shell_R13_SP2
C:\Product\TAFJ\TAFJ.R13_SP2\appserver\was>createStandalone
INSTCONFSUCCESS: Success: Profile AppSrv01 now exists. Please consult C:\Product\AppSrv\WebSphere\8.5.5\AppSe
file.txt for more information about this profile.
ADMU0116I: Tool information is being logged in file
          C:\Product\AppSrv\WebSphere\8.5.5\AppServer\profiles\AppSrv01\logs\server1\startServer.log
ADMU0128I: Starting tool with the AppSrv01 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 27736
C:\Product\TAFJ\TAFJ.R13_SP2\appserver\was>
```

The screenshot shows a web browser window titled "WebSphere Integrated...". The address bar displays the URL <https://l38vgrm1:9043/ibm/console/unsecureLc>. The page itself is the "WebSphere Integrated Solutions Console" login screen. It features a graphic of three stylized human figures (blue, yellow, green) surrounding a large yellow key icon. The text "WebSphere software" is at the top left, and "WebSphere Integrated Solutions Console" is centered above the login fields. A "User ID:" label is followed by a text input field, and a "Log in" button is below it. At the bottom, there is a copyright notice: "Licensed Materials - Property of IBM (c) Copyright IBM Corp. 1997, 2011 All Rights Reserved. IBM, the IBM logo, ibm.com and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [Copyright and trademark information](#)". The IBM logo is also present at the bottom left.

One step configuration

To have a working deployment you need to configure:

- Environment variables and shared libraries (TAFJ and T24 libraries)
- JMS resources (Communication channels between applications)
- JDBC data sources (Database connectivity)
- Applications deployment (TAFJ ear file, BrowserWeb ...)

You can run all configurations: **environment configuration, JMS configuration, JDBC configuration, application deployments for T24 and TAFJ** by using a single script or by doing a one step at a time configuration.

Previously you have had to fill out the main configuration file: **%TAFJ_HOME %lappserver\wasljython\ tafj.py**.

This file contains the variables that are needed at each configuration step. If you need more information about these variables or if you want to adapt the scripts to your specific use case please refer to the next section of this document.

The below assumes you are using a local Oracle database with database name TAFJDB and no MQ Series message broker.

If you want to use another database or MQ, fill out the below properties in **tafj.py** and modify the file **T24Setup.py** appropriately.

```
#####
# Node details
#
cell_name="AppSrv01Node01Cell"
node_name="AppSrv01Node01"
server_name="server1"
#####
# Path details
#
tafjHome="D:\Temenos\Reference\Temenos\TAFJ"
t24Home="D:\Temenos\Reference\Temenos\T24\lib\t24lib"
#####
# database details
#
dbDriver="h2_1.3.175"
dbHostName="localhost"
dbPort="[port]"
dbName="T24"
dbUser="mbtafj"
dbPwd="mbtafj"
#dbR0HostName="[hostname]"
```



```
#dbR0Port="[port]"
#dbR0dbName="[sid]"
#dbR0dbUser="[user]"
#dbR0dbPwd="[password]"
#####
# database oracle specific
#
#use xdb.jar for Oracle 11g
#use xdb6.jar for Oracle 12c
#
oracleXDBjarName="xdb.jar"
#####
# database db2 specific
#
#use type 4
#use type 2
#
dbDriverType="4"
#####
# Browser details
#
# path of your BrowserWar file
browserWebHome="D:\Temenos\Reference\3rdParty\as\jBoss\jboss-
4.2.3.GA\server\T24\deploy"
#####
# MQ Details details
#
# DO NOT PUT "localhost" but the IP for MQ!
mqManager="MQT24"
mqServerName="[IP]"
mqPort="1414"
mqChannel="T24Channel"
```

To find your cell_name, go to WAS_HOME/{\$PROFILE_NAME}/config/cells. The directory under here is your cell name. If you go under this directory and then the nodes directory, you find your node name.

Multiple profiles

Please note that if you have multiple profiles defined and if you want to execute the scripts against a profile which is not the default one you will have to add to the command presented below the profile name.

-profileName %PROFILE_NAME%

Security enabled

If you have enabled the admin console security you will be prompt to enter your admin user and password when executing the script or you will have to add to the command presented below your admin user and password.

-user %ADMIN_USER_NAME% -password %ADMIN_USER_PASSWORD%
%

The script of the complete configuration is located at **%TAFJ_HOME%\appserver\was\python\T24Setup.py**

Edit the file and comment and uncomment what you need:

```
print "Setup Environment"
execfile('environmentConfiguration.py')

print "JMS Environment"
execfile('jmsConfiguration.py')
#or
#execfile('mqConfiguration.py')

print "Setup Database"
execfile('ORACLEConfiguration.py')
#or
#execfile('DB2Configuration.py')
#or
#execfile('MSSQLConfiguration.py')
#or
#execfile('H2Configuration.py')

print "Setup TAFJ EE (MDB & EJB)"
execfile('TAFJapplicationDeployment.py')

print "Setup Browser"
```



```
execfile('T24applicationDeployment.py')
```

Launch the following command from %TAFJ_HOME%\appserver\was\jython dir on the command line for one step installation.

```
%WAS_HOME%\bin\wsadmin -lang jython -f T24Setup.py -profileName  
%PROFILE_NAME%
```

```
\\ NOTE: Syntax errors in the policy files will cause the enterprise application FAIL to start.  
 Extreme care should be taken when editing these policy files. It is advised to use  
 the policytool provided by the JDK for editing the policy files  
 ({WAS_HOME}/java/jre/bin/policytool).  
  
grant codeBase "file:${application}" {  
};  
  
grant codeBase "file:${jars}" {  
};  
  
grant codeBase "file:${connectorComponent}" {  
};  
  
grant codeBase "file:${webComponent}" {  
};  
  
grant codeBase "file:${ejbComponent}" {  
};  
  
ADMA0516I: Installation of BrowserWeb_war started.  
ADMA0115W: Resource assignment of name jca:t24ConnectionFactory and type com.ibm.jbase.jremote.JConnectionFactory, with JN  
is not found within scope of module BrowserWeb with URI BrowserWeb.war,WEB-INF/web.xml deployed to target websphere:c  
rv01Node01,server=server1.  
ADMA5068I: The resource validation for application BrowserWeb_war completed successfully, but warnings occurred during  
ADMA5081I: Application and module versions are validated with versions of deployment targets.  
ADMA5053I: The application BrowserWeb_war is configured in the WebSphere Application Server repository.  
ADMA5051I: The application BrowserWeb_war is configured in the WebSphere Application Server repository.  
ADMA5081I: The bootstrap address for Client module is configured in the WebSphere Application Server repository.  
ADMA5053I: The library references for the installed optional package are created.  
ADMA5005I: The application BrowserWeb_war is configured in the WebSphere Application Server repository.  
ADMA5001I: The application binaries are saved in C:\Product\appSrv\Websphere\8.5.5\AppServer\profiles\AppSrv01\wstemp\  
s\AppSrv01Node01Cell\applications\BrowserWeb_war.ear\BrowserWeb_war.ear  
ADMA5005I: The application BrowserWeb_war is configured in the WebSphere Application Server repository.  
SEMA5005I: Successully updated the application BrowserWeb_war with the appContextIDForSecurity information.  
ADMA5005I: The application BrowserWeb_war is configured in the WebSphere Application Server repository.  
ADMA5005I: The application BrowserWeb_war is configured in the WebSphere Application Server repository.  
ADMA5131I: Activation plan created successfully.  
ADMA5011I: The cleanup of the temp directory for application BrowserWeb_war is complete.  
ADMA5013I: Application BrowserWeb_war installed successfully.  
BrowserWeb deployed  
  
C:\Product\TAFJ\TAFJ.R13_SP2\appserver\was\jython>
```

Note: The property **t24dpPwd** is not correctly set, you have to run the administrative console to re-enter it manually as shown below.

Security ▾ Global security ▾ Java Authentication and Authorization Service ▾ [J2C authentication data](#) ▾ IT24 database DB2

password: *****



The screenshot shows the 'Global security' configuration page. The left sidebar lists various administrative tasks under 'Security'. The main panel displays the 'JAAS - J2C authentication data' section for the 'App_T24_ALONE' profile. It shows an alias 'App_T24_ALONENode01/T24 database DB2' with a user ID 'EUROPE\user' and a password '*****'. A 'Description' field is empty. At the bottom are 'Apply', 'OK', 'Reset', and 'Cancel' buttons.

Click OK.

The screenshot shows the same 'Global security' configuration page after changes have been saved. A message box at the top right indicates that changes have been made to the local configuration and provides options to save directly to the master configuration or review changes before saving. Below the message, the 'Messages' section states that the server needs to be restarted for changes to take effect. The main panel shows the updated alias configuration and a 'Preferences' section where a new alias 'App_T24_ALONENode01/T24 database DB2' has been added for the user 'EUROPE\uncharpin'.

Click Apply then Click Save.



One step at a time configuration

This section will cover in details each configuration step presented above in the “One step configuration” section.

If you have executed the “One step configuration” you don’t need to execute the following scripts.

Environment configuration

The script is located at **%TAFJ_HOME%\appserver\was\jython\environmentConfiguration.py**

This script will set the following properties:

- TAFJ_HOME, ORACLE_TAFJ_JDBC_DRIVER_PATH,
T24_MODEL BANK_SHARED LIB, TAFJ shared libraries, -Dtafj.home

Before running it you have to define some parameters such as **node**, **cell** and **server name** to set the scope of the configuration and **tafjHome**, **t24Home** to specify the environment variable values.

To do so edit the file at **%TAFJ_HOME%\appserver\was\jython\tafj.py** and set the following properties i.e.:

```
cell_name="localhostNode01Cell"  
node_name="localhostNode01"  
server_name="server1"  
tafjHome="C:/product/appserver/websphere/TAFJ"  
t24Home="C:/product/appserver/websphere/T24"
```

To find your cell_name, go to WAS_HOME/{\$PROFILE_NAME}/config/cells. The directory under here is your cell name. If you go under this directory and then the nodes directory, you find your node name.

Multiple profiles

Please note that if you have multiple profiles defined and if you want to execute the scripts against a profile which is not the default one you will have to add to the command presented below the profile name.

-profileName %PROFILE_NAME%

Security enabled



If you have enabled the admin console security you will be prompted to enter your admin user and password when executing the script or you will have to add to the command presented below your admin user and password.

```
-user %ADMIN_USER_NAME% -password %ADMIN_USER_PASSWORD%
%
```

Launch the following command from the command line:

```
%WAS_HOME%\bin\wsadmin -lang jython -f
environmentConfiguration.py
```

```
C:\product\TAFJ\appserver\was>wsadmin -lang jython -f %CONF_PATH%\environmentConfiguration.py
WASX7209I: Connected to process "server1" on node localhostNode01 using SOAP connector; The type of process is: UnManagedProcess
TAFJ_HOME set to C:/product/appserver/websphere/TAFJ
ORACLE_TAFJ_JDBC_DRIVER_PATH set to C:/product/appserver/websphere/TAFJ/dbdrivers/oracle-11g
T24_MODELBANK_SHAREDLIB set to C:/product/appserver/websphere/T24
-Dtafj.home set to C:/product/appserver/websphere/TAFJ
T24 shared libraries created
TAFJ shared libraries created
Environment configuration saved

C:\product\TAFJ\appserver\was>
```

You can browse the administrative console to check that this step has been correctly done.

Environment □ WebSphere variables

Environment □ Shared Libraries

Server -> Server Type -> WebSphere application servers -> #profileName# -> (Server infrastructure) Java and Process Management ->[Process definition](#) -> [Java Virtual Machine](#)

Generic JVM arguments : -Dtafj.home= %TAFJ_HOME%

Exemple : -Dtafj.home= /opt/wasapps/TAFJ



The screenshot shows the Integrated Solutions Console (ISC) interface. On the left, the navigation tree includes categories like Guided Activities, Servers, Application Types, Services, Resources, Security, Environment, System Administration, Users and Groups, Monitoring and Tuning, Troubleshooting, Service Integration, and VODI. The 'Server' node is expanded, showing Server Types (WebSphere application servers, WebSphere MQ servers, Web servers), Application Types (WebSphere enterprise applications, Business-level applications, Assets), and Environment settings (Virtual hosts, Update global Web server plug-in configuration, WebSphere variables, Shared libraries, Replication domains, Naming). The main panel displays the 'Java Virtual Machine' configuration for a specific server. It has tabs for Configuration and Runtime. Under Configuration, there are sections for General Properties (Classpath, Boot Classpath, checkboxes for Verbose class loading, Verbose garbage collection, and Verbose JRI), System Properties (Initial heap size set to 128 MB, Maximum heap size set to 256 MB, Run HProf checkbox, HProf Arguments input field, Debug Mode checkbox, Debug arguments input field, Generic JVM arguments input field, Executable JAR file name input field, and Disable JIT checkbox), and Operating system name (set to z/OS). Buttons at the bottom include Apply, OK, Reset, and Cancel.



JMS Resources Configuration

The script is located at %TAFJ_HOME%\appserver\was\jython\jmsConfiguration.py

It will execute the following actions:

- Create a service integration and adding a bus member
- Create a JMS connection factory
- Create JMS Queues
- Create a JMS topic
- Create activation specifications

Before running it you have to define some parameters such as **node**, **cell** and **server name** to set the scope of the configuration in the %TAFJ_HOME%\appserver\was\jython\tafj.py file. You might have already done it previously in the environment configuration step.

Launch the following command in the command line (refer to Environment configuration for additional command parameters).

```
%WAS_HOME%\bin\wsadmin -lang jython -f jmsConfiguration.py
```

```
C:\product\TAFJ\appserver\was>wsadmin -lang jython -f %CONF_PATH%\jmsConfiguration.py
WASX7209I: Connected to process "server1" on node localhostNode01 using SOA
nector; The type of process is: UnManagedProcess
T24Bus created
t24ConnectionFactory created
t240FSQueue created
t240FSReplyQueue created
t24NEOQueue created
t24NEORReplyQueue created
t24ARCIBQueue created
t24ARCIBRReplyQueue created
t24SEATQueue created
t24SEATRReplyQueue created
t24EXECQueue created
tecEventsTopic created
t240FSQueue - t240FMessageMDB activation created
t24ARCIBQueue - t24ARCIBMessageMDB activation created
t24NEOQueue - t24NEOMessageMDB activation created
t24SEATQueue - t24SEATMessageMDB activation created
t24EXECQueue - t24EXECMessageMDB activation created
JMS configuration saved

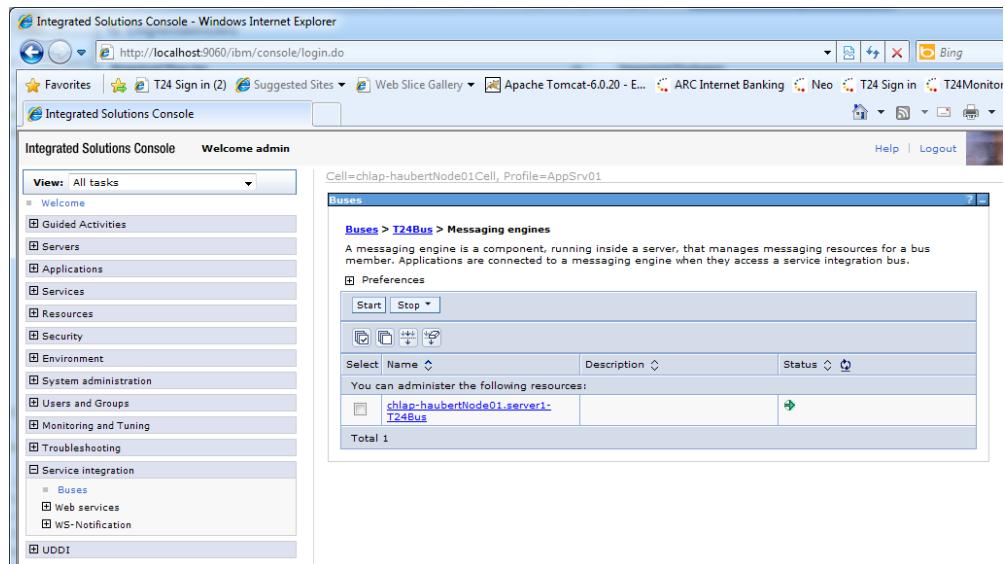
C:\product\TAFJ\appserver\was>
```

You can browse the administrative console to check this step has been correctly done.

Service Integration □ Buses □

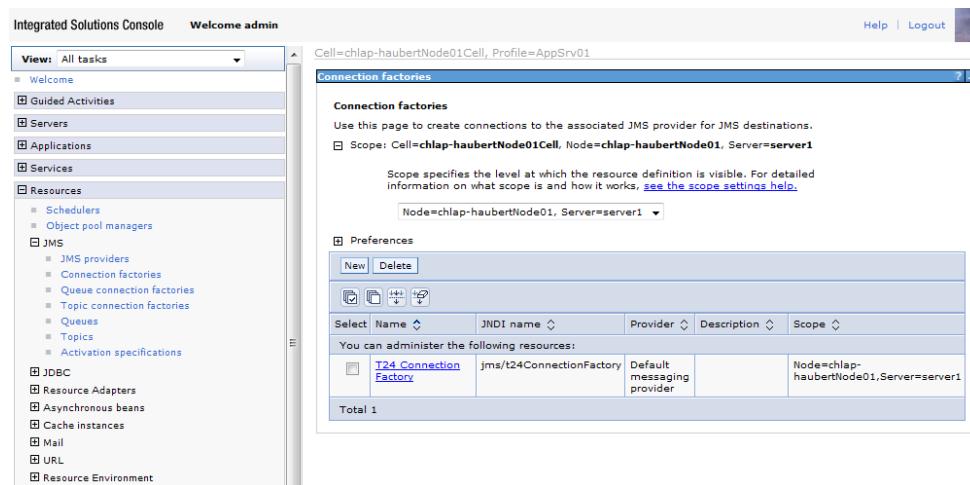
You should see: T24Bus

Message engines status (under **Buses->T24 Bus -> Messaging engines**) should automatically be  (Started).



Select	Name	Description	Status
<input type="checkbox"/>	chlap-hauberNode01.server1-T24Bus		 Started

Check connection factory was created:



Select	Name	JNDI name	Provider	Description	Scope
<input type="checkbox"/>	T24 Connection Factory	jms/t24ConnectionFactory	Default messaging provider		Node=chlap-hauberNode01.Server=server1

Check activation specs created:



The screenshot shows the WebSphere Admin Console interface. The left sidebar navigation includes 'System policy sets', 'Default policy set bindings', 'General provider policy set bindings', 'General client policy set bindings', 'Trust service' (with 'Token providers', 'Targets', and 'Trust service attachments'), 'Security cache', and 'Available messaging state'. Under 'Resources', there are sections for 'Schedulers', 'Object pool managers', 'JMS' (with 'JMS providers', 'Connection factories', 'Queue connection factories', 'Topic connection factories', 'Topics', and 'Activation specifications'), 'JDBC', 'Resource Adapters', 'Asynchronous beans', 'Cache instances', 'Mail', 'URL', 'Resource Environment', and 'Resource environment entries'. Under 'Security', there are sections for 'Environment' (with 'Virtual hosts', 'Update global Web server plug-in configuration', 'WebSphere variables', 'Shared libraries', 'SIP application routers', 'Replication domains', 'Naming', and 'OSGi bundle repositories'). The main content area displays 'Activation specifications' for JMS resources, listing the following:

Select	Name	JNDI name	Provider	Description	Scope
<input type="checkbox"/>	T24_ARCIB_Message_MDB	jms/124ARCIBMessageMDB	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_CALLAT_Message_MDB	jms/124CALLATMessageMDB	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_EXEC_Message_MDB	jms/124EXECMessageMDB	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_QFS_Message_MDB	jms/124QFSMessageMDB	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_SEAT_Message_MDB	jms/124SEATMessageMDB	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_TWS_Message_MDB	jms/124TWSMessageMDB	Default messaging provider		Node=L40R81P1Node02,Server=server1

Total 6

Check queues created:

The screenshot shows the WebSphere Admin Console interface, similar to the previous one but focusing on 'Queues'. The left sidebar navigation includes 'System policy sets', 'Default policy set bindings', 'General provider policy set bindings', 'General client policy set bindings', 'Trust service' (with 'Token providers', 'Targets', and 'Trust service attachments'), 'Security cache', and 'Available messaging state'. Under 'Resources', there are sections for 'Schedulers', 'Object pool managers', 'JMS' (with 'JMS providers', 'Connection factories', 'Queue connection factories', 'Topic connection factories', 'Topics', and 'Activation specifications'), 'JDBC', 'Resource Adapters', 'Asynchronous beans', 'Cache instances', 'Mail', 'URL', 'Resource Environment', and 'Resource environment entries'. Under 'Security', there are sections for 'Environment' (with 'Virtual hosts', 'Update global Web server plug-in configuration', 'WebSphere variables', 'Shared libraries', 'SIP application routers', 'Replication domains', 'Naming', and 'OSGi bundle repositories'). The main content area displays 'Queues' for JMS resources, listing the following:

Select	Name	JNDI name	Provider	Description	Scope
<input type="checkbox"/>	T24_ARCIB_Queue	jms/124ARCIBQueue	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_ARCIB_Reply_Queue	jms/124ARCIBReplyQueue	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_CALLAT_Queue	jms/124CALLATQueue	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_CALLAT_Reply_Queue	jms/124CALLATReplyQueue	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_EXEC_Queue	jms/124EXECQueue	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_QFS_Queue	jms/124QFSQueue	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_QFS_Reply_Queue	jms/124QFSReplyQueue	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_SEAT_Queue	jms/124SEATQueue	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_SEAT_Reply_Queue	jms/124SEATReplyQueue	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_TWS_Queue	jms/124TWSQueue	Default messaging provider		Node=L40R81P1Node02,Server=server1
<input type="checkbox"/>	T24_TWS_Reply_Queue	jms/124TWSReplyQueue	Default messaging provider		Node=L40R81P1Node02,Server=server1

Check topics created:



The screenshot shows the WebSphere Application Server V8.5.5 Administration Console interface. The left sidebar contains navigation links for System policy sets, Trust service, Resources (including JMS providers, JDBC, Resource Adapters, Cache instances, Mail, URL, Resource Environment, and Security), and Environment (Virtual hosts, WebSphere variables, shared libraries, SIEP application routers, Application domains, Namespaces, and OSGI bundle repositories). The main panel displays the 'Topics' configuration page. It includes a help section for 'Scope' and 'Field help'. A table lists a single topic entry:

Select	Name	JNDI name	Provider	Description	Scope
<input type="checkbox"/>	Tec_Events Topic	jms/tecEventsTopic	Default messaging provider		Node=L40R81P1Node02,Server=server1

Total 1



JDBC Resources configuration

T24 Oracle 11g database configuration

When you setup the datasource you have to use the drivers provide by the database. The %TAFJ_HOME%\DBDrivers is just a helper. We cannot guaranty the drivers we provide with TAFJ is working for all version of Database.

The script is located at %TAFJ_HOME% \appserver\was\jython\ORACLEConfiguration.py

It will execute the following actions:

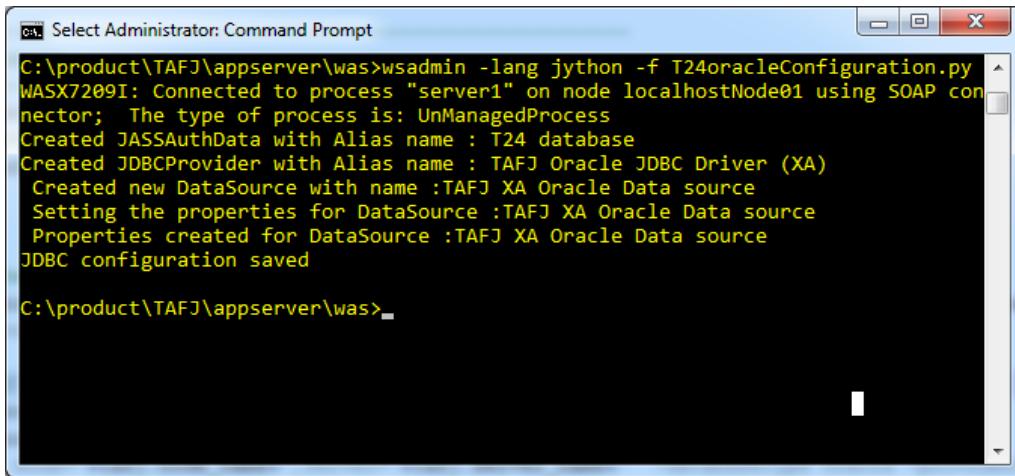
- Create the T24 Manager Database Credentials
- Create a JDBC Provider for TAFJ
- Create a DataSource for TAFJ

Before running it you have to define some parameters such as **node**, **cell** and **server name** to set the scope of the configuration in the %TAFJ_HOME%\appserver\was\jython\ tafj.py file. You must also define the following properties to identify your Oracle database and provide a **user id and password** for authentication i.e. :

```
dbDriver="oracle-12c"
dbHostName="172.16.22.102"
dbPort="1521"
dbName="MB2013"
dbUser="mbtafj"
dbPwd="mbtafj"
```

Launch the following command in the command line (refer to Environment configuration for additional command parameters).

```
%WAS_HOME%\bin\ wsadmin -lang python -f ORACLEConfiguration.py
```



```
C:\product\TAFJ\appserver\was>wsadmin -lang jython -f T24oracleConfiguration.py
WASX7209I: Connected to process "server1" on node localhostNode01 using SOAP connector; The type of process is: UnManagedProcess
Created JASSAuthData with Alias name : T24 database
Created JDBCProvider with Alias name : TAFJ Oracle JDBC Driver (XA)
Created new DataSource with name :TAFJ XA Oracle Data source
Setting the properties for DataSource :TAFJ XA Oracle Data source
Properties created for DataSource :TAFJ XA Oracle Data source
JDBC configuration saved

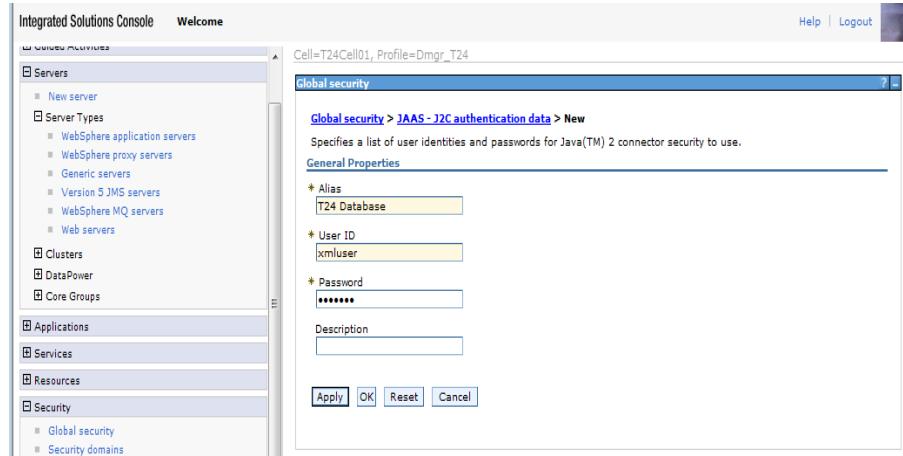
C:\product\TAFJ\appserver\was>
```

You can browse the administrative console to check this step has been correctly done.

Note: The property **dbPwd** is not correctly set, you have to run the administrative console to re-enter it manually as shown below.

Security □ Global security □ Java Authentication and Authorization Service □ J2C authentication data □ T24 database

password: *****



The screenshot shows the 'Global security > JAAS - J2C authentication data > New' dialog box. The 'General Properties' section contains the following fields:

- Alias:** T24 Database
- User ID:** xmluser
- Password:** *****
- Description:** (empty)

At the bottom of the dialog are buttons for **Apply**, **OK**, **Reset**, and **Cancel**.

Click OK.



The screenshot shows the Integrated Solutions Console interface. On the left, there's a navigation sidebar with options like Welcome, Servers, Server Types, Applications, New Application, Application Types (WebSphere enterprise applications, Business-level applications, Assets), Services, Resources, Security (Global security, Security domains, Administrative Authorization Groups, SSL certificate and key management, Security auditing, Bus security), Environment, System administration, and Users and Groups. The main panel is titled 'Global security' under 'Cell=localhostNode01Cell, Profile=App_T24_ALONE'. It shows a 'Messages' section with a warning about changes made to local configuration. Below that is a 'Global security > JAAS - J2C authentication data' section, which specifies a list of user identities and passwords for Java(TM) 2 connector security. A checkbox 'Prefix new alias names with the node name of the cell (for compatibility with earlier releases)' is checked. There are 'Apply' and 'Save' buttons. The 'Preferences' section shows a table with one row: Alias: App_T24_ALONENode01/T24_database, User ID: tafj, and Description: . The total count is 1.

Click Apply then Click Save.

T24 DB2 database configuration

The script is located at %TAFJ_HOME%\appserver\was\jython\ DB2Configuration.py

It will execute the following actions:

- Create the T24 DB2 Database Credentials
- Create a JDBC Provider for TAFJ
- Create a DataSource for TAFJ

Before running it you have to define some parameters such as **node**, **cell** and **server name** to set the scope of the configuration in the %TAFJ_HOME%\appserver\was\jython\ tafj.py file. You must also define the following properties to identify your DB2 database and provide a **user id** and **password** for authentication i.e. :

```
dbDriver="db2_v10.1"
dbHostName="localhost"
dbPort="50000"
dbDriverType="4"
dbName="MB2013"
dbUser="mbtafj"
dbPwd="mbtafj"
```

Launch the following command in the command line (refer to Environment configuration for additional command parameters).

```
%WAS_HOME%\bin\ wsadmin -lang jython -f DB2Configuration.py
```



```
C:\install\python>wsadmin -lang python -f T24DB2Configuration.py
WASX7209I: Connected to process "server1" on node App_T24_ALONENode01 using SOAP
connector; The type of process is: UnManagedProcess
Created JASSAuthData with Alias name : T24 database DB2
Created JDBCProvider with Alias name : DB2 Universal JDBC Driver Provider (XA)
Created new DataSource with name :TAFJ XA DB2 Data source
Setting the properties for DataSource :TAFJ XA DB2 Data source
Properties created for DataSource :TAFJ XA DB2 Data source

C:\install\python>
```

You can browse the administrative console to check this step has been correctly done.

Note: The property **dbPwd** is not correctly set, you have to run the administrative console to re-enter it manually as shown below.

Security ▶ Global security ▶ Java Authentication and Authorization Service ▶ J2C authentication data ▶ T24 database DB2

password: *****

Integrated Solutions Console Welcome admin

Cell=localhostNode01Cell, Profile=App_T24_ALONE

Global security > JAAS - J2C authentication data > App_T24_ALONENode01/T24 database DB2

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias: App_T24_ALONENode01/T24 database DB2

* User ID: EUROPEUSER

* Password: *****

Description:

Apply OK Reset Cancel

Click OK.



The screenshot shows the Integrated Solutions Console interface. On the left, there's a navigation tree with categories like Welcome, Guided Activities, Servers, Applications, Services, Resources, Security (which is expanded to show Global security, Security domains, etc.), Environment, System administration, Users and Groups, Monitoring and Tuning, Troubleshooting, Service integration, and UDDI. The main panel is titled 'Global security' under 'JAAS - J2C authentication data'. It contains a message box stating: 'Changes have been made to your local configuration. You can: • Save directly to the master configuration. • Review changes before saving or discarding.' Below this is a note: 'The server may need to be restarted for these changes to take effect.' There's a checkbox 'Prefix new alias names with the node name of the cell (for compatibility with earlier releases)' followed by an 'Apply' button. A 'Preferences' section shows a table with one row: 'Alias' (App_T24_ALONENode01/T24_database), 'User ID' (EUROPE\nccharpin), and 'Description' (DB2). The table has buttons 'New' and 'Delete' at the top.

Click Apply then Click Save.

T24 MSSQL database configuration

The script is located at %TAFJ_HOME%\appserver\was\jython\ MSSQLConfiguration.py

It will execute the following actions:

- Create the T24 MSSQL Database Credentials
- Create a JDBC Provider for TAFJ
- Create a DataSource for TAFJ

Before running it you have to define some parameters such as **node**, **cell** and **server name** to set the scope of the configuration in the %TAFJ_HOME%\appserver\was\jython\ tafj.py file. You must also define the following properties to identify your DB2 database and provide a **user id and password** for authentication i.e. :

```
dbDriver="sqljdbc_1.2"
dbServerName="localhost"
dbPort="1433"
dbName="R14DB"
dbUser="R14DB"
dbPwd="R14DB"
```

Launch the following command in the command line (refer to Environment configuration for additional command parameters).

```
%WAS_HOME%\bin\ wsadmin -lang jython -f MSSQLConfiguration.py
```



```
<ggowri@hmldevblp7a1:/storage1/home2/ggowri/IBM/WebSphere/Appserver/profiles/t24_db2/bin> wsadmin.sh -lang jython -f /sto
FJ/appserver/was/jython/MSSQLConfigurationtest.py
WASX7209I: Connected to process "server1" on node hmldevblp7a1Node02 using SOAP connector; The type of process is: UnMan
TAFJ shared libraries created
MICROSOFT JDBC DRIVER PATH set to /storage1/home2/ggowri/IBM/WebSphere/Appserver/TAFJ/dbdrivers/sqljdbc_1.2
MICROSOFT JDBC DRIVER_NATIVEPATH set to /storage1/home2/ggowri/IBM/WebSphere/Appserver/TAFJ/dbdrivers/sqljdbc_1.2
Created JASSAuthData with Alias name : T24 database MSSQL
Created JDBC Provider with the name :Microsoft SQL Server JDBC Driver (XA)
Created JDBC Provider with the name :Microsoft SQL Server JDBC Driver
Created JDBC DataSource with the name :TAFJ XA MSSQL Data source
Setting the properties for DataSource :TAFJ XA MSSQL Data source
Created JDBC DataSource with the name :TAFJ LOCK NON-XA MSSQL Data source
Setting the properties for DataSource :TAFJ LOCK NON-XA MSSQL Data source
MSSQL JDBC configuration saved
```

You can browse the administrative console to check this step has been correctly done.

The screenshot shows the 'JDBC providers' page in the IBM WebSphere Administrative Console. The page title is 'JDBC providers'. A descriptive text block explains that the JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. It includes a link to a guided activity and a general information link. Below this, a scope section shows 'Scope: Cell=hmldevblp7a1Node02Cell, Node=hmldevblp7a1Node02, Server=server1'. A note below the scope says 'Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.' A dropdown menu shows 'Node=hmldevblp7a1Node02, Server=server1'. A 'Preferences' section contains 'New...', 'Delete', and other icons. A table lists two JDBC providers:

Select	Name	Scope	Description
<input type="checkbox"/>	Microsoft SQL Server JDBC Driver	Node=hmldevblp7a1Node02,Server=server1	Microsoft SQL Server JDBC Driver. This provider is configurable in version 6.1.0.15 and later nodes.
<input type="checkbox"/>	Microsoft SQL Server JDBC Driver (XA)	Node=hmldevblp7a1Node02,Server=server1	Microsoft SQL Server JDBC Driver (XA). This provider is configurable in version 6.1.0.15 and later nodes.

Total 2



Data sources

Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=hmldevblp7a1Node02Cell, Node=hmldevblp7a1Node02, Server=server1

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=hmldevblp7a1Node02, Server=server1

Preferences

New... Delete Test connection Manage state...

Select	Name	JNDI name	Scope	Provider	Description	Category
<input type="checkbox"/>	TAFJ LOCK NON-XA MSSQL Data source	jdbc/t24LockingDataSource	Node=hmldevblp7a1Node02,Server=server1	Microsoft SQL Server JDBC Driver	NON-XA MSSQL Locking Data source	
<input type="checkbox"/>	TAFJ XA MSSQL Data source	jdbc/t24DataSource	Node=hmldevblp7a1Node02,Server=server1	Microsoft SQL Server JDBC Driver (XA)	XA MSSQL Data source	

Total 2

WebSphere Variables

WebSphere Variables

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

Scope: Cell=hmldevblp7a1Node02Cell, Node=hmldevblp7a1Node02, Server=server1

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Node=hmldevblp7a1Node02, Server=server1

Preferences

New... Delete

Select	Name	Value	Scope
<input type="checkbox"/>	MICROSOFT JDBC DRIVER_NATIVEPATH	\${TAFJ_HOME}/dbdrivers/sqljdbc_1.2	Node=hmldevblp7a1Node02,Server=server1
<input type="checkbox"/>	MICROSOFT JDBC_DRIVER_PATH	\${TAFJ_HOME}/dbdrivers/sqljdbc_1.2	Node=hmldevblp7a1Node02,Server=server1

Note: The property **dbPwd** is not correctly set, you have to run the administrative console to re-enter it manually as shown below.

Security □ **Global security** □ **Java Authentication and Authorization Service** □
J2C authentication data □ **T24 database MSSQL**

password: *****



[Global security](#) > [JAAS - J2C authentication data](#) > **hmldevblp7a1Node02/T24 database MSSQL**

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias

* User ID

* Password

Description

Click OK.

Global security

Messages

⚠ Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

[Global security](#) > [JAAS - J2C authentication data](#)

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Preferences

<input type="button" value="New..."/>	<input type="button" value="Delete"/>		
<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	<input type="checkbox"/>		
<input type="checkbox"/>	<input type="checkbox"/>		
Select	Alias ▲	User ID ▲	Description ▲
You can administer the following resources:			
<input type="checkbox"/>	hmldevblp7a1Node02/T24 database MSSQL	R14DB	
Total 1			

Click Apply then Click Save.

TAFJ Application deployment



TAFJ application deployment can be run automatically.

The script is located at %TAFJ_HOME%
\\appserver\\was\\jython\\TAFJapplicationDeployment.py

It will execute the following actions:

- Deploy TAFJJEE_EAR.ear application

Before running it you have to define some parameters such as **node**, **cell** and **server name** to set the scope of the configuration in the %TAFJ_HOME%\\appserver\\was\\jython\\ tafj.py file. You must also define the **tafjHome** as root folder to locate the applications to be deployed i.e. :

tafjHome="C:/product/appserver/websphere/TAFJ"

Launch the following command in the command line (refer to Environment configuration for additional command parameters).

wsadmin -lang jython -f TAFJapplicationDeployment.py

```
C:\product\TAFJ\appserver\was>wsadmin -lang jython -f TAFJapplicationDeployment.py
WASX7209I: Connected to process "server1" on node localhostNode01 using SOAP connector; The type of process is: UnManagedProcess
TAFJJEE_EAR will be deployed from path C:/product/appserver/websphere/TAFJ/appserver/TAFJJEE_EAR.ear
ADMA5016I: Installation of TAFJJEE_EAR started.
ADMA5067I: Resource validation for application TAFJJEE_EAR completed successfully.
ADMA5058I: Application and module versions are validated with versions of deployment targets.
ADMA5005I: The application TAFJJEE_EAR is configured in the WebSphere Application Server repository.
ADMA5053I: The library references for the installed optional package are created.
ADMA5005I: The application TAFJJEE_EAR is configured in the WebSphere Application Server repository.
ADMA5001I: The application binaries are saved in C:/product/appserver/websphere/profiles/App_T24.ALONE/wstemp/Script12ec8ba7fa3/workspace/cells/localhostNode01/cell/applications/TAFJJEE_EAR.ear|TAFJJEE_EAR.ear
ADMA5005I: The application TAFJJEE_EAR is configured in the WebSphere Application Server repository.
SECJ0400I: Successfully updated the application TAFJJEE_EAR with the appContextIDForSecurity information.
ADMA5005I: The application TAFJJEE_EAR is configured in the WebSphere Application Server repository.
ADMA5113I: Activation plan created successfully.
ADMA5011I: The cleanup of the temp directory for application TAFJJEE_EAR is complete.
ADMA5013I: Application TAFJJEE_EAR installed successfully.
TAFJEE_EAR deployed
```

.....

```
c:\ Administrator: Command Prompt
ADM5016I: Installation of push-server_war started.
ADM50159W: Duplicate root context(/) was found on the same node localhostNode01
and same host default_host
ADM5068I: The resource validation for application push-server_war completed su
ccessfully, but warnings occurred during validation.
ADM5058I: Application and module versions are validated with versions of deplo
ment targets.
ADM5005I: The application push-server_war is configured in the WebSphere Appli
cation Server repository.
ADM5053I: The library references for the installed optional package are create
.
ADM5005I: The application push-server_war is configured in the WebSphere Appli
cation Server repository.
ADM5001I: The application binaries are saved in C:\product\appserver\websphere
profiles\App_T24_ALONE\wstemp\Script12ec8ba7fa3\workspace\cells\localhostNode01
cell\applications\push-server_war.ear\push-server_war.ear
ADM5005I: The application push-server_war is configured in the WebSphere Appli
cation Server repository.
SECJ0400I: Successfully updated the application push-server_war with the appCon
extIDForSecurity information.
ADM5005I: The application push-server_war is configured in the WebSphere Appli
cation Server repository.
ADM5113I: Activation plan created successfully.
ADM5011I: The cleanup of the temp directory for application push-server_war is
complete.
ADM5013I: Application push-server_war installed successfully.
push-server deployed
TAFJEE_EAR_MONITOR will be deployed from path C:/product/appserver/websphere/TA
FJ/appserver/
ADM5016I: Installation of TAFJJEE_EAR_Monitor started.
ADM5067I: Resource validation for application TAFJJEE_EAR_Monitor completed su
ccessfully.
ADM5058I: Application and module versions are validated with versions of deplo
ment targets.
ADM5005I: The application TAFJJEE_EAR_Monitor is configured in the WebSphere A
pplication Server repository.
ADM5053I: The library references for the installed optional package are create
.
ADM5005I: The application TAFJJEE_EAR_Monitor is configured in the WebSphere A
pplication Server repository.
ADM5001I: The application binaries are saved in C:\product\appserver\websphere
profiles\App_T24_ALONE\wstemp\Script12ec8ba7fa3\workspace\cells\localhostNode01
cell\applications\TAFJJEE_EAR_Monitor.ear\TAFJJEE_EAR_Monitor.ear
ADM5005I: The application TAFJJEE_EAR_Monitor is configured in the WebSphere A
pplication Server repository.
SECJ0400I: Successfully updated the application TAFJJEE_EAR_Monitor with the ap
ContextIDForSecurity information.
ADM5005I: The application TAFJJEE_EAR_Monitor is configured in the WebSphere A
pplication Server repository.
ADM5113I: Activation plan created successfully.
ADM5011I: The cleanup of the temp directory for application TAFJJEE_EAR_Monito
is complete.
ADM5013I: Application TAFJJEE_EAR_Monitor installed successfully.
TAFJEE_EAR_MONITOR deployed
TAFJ application deployment saved

C:\product\TAFJ\appserver\was>_
```

You have to stop and restart your server to start the applications.

Type the following command in the command line:

```
stopServer <your server name>
```



```
C:\ Select Administrator: Command Prompt
C:\product\TAFJ\appserver\was\jython>stopServer server1
ADMU0116I: Tool information is being logged in file
          C:\product\AppServer\WebSphere\profiles\App_T24_ALONE\logs\server1\stopServer.log
ADMU0128I: Starting tool with the App_T24_ALONE profile
ADMU3100I: Reading configuration for server: server1
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.

C:\product\TAFJ\appserver\was\jython>
```

Type the following command in the command line:

startServer <your server name>

```
C:\ Select Administrator: Command Prompt
C:\product\TAFJ\appserver\was\jython>startServer server1
ADMU0116I: Tool information is being logged in file
          C:\product\AppServer\WebSphere\profiles\App_T24_ALONE\logs\server1\startServer.log
ADMU0128I: Starting tool with the App_T24_ALONE profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 1772

C:\product\TAFJ\appserver\was\jython>
```

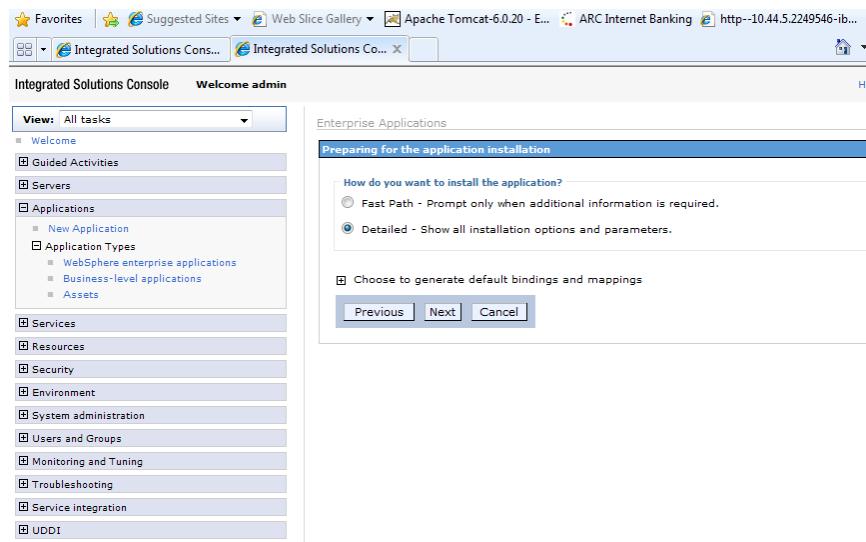
You can then browse the administrative console to check they have been correctly deployed and started.

Console Mode

You could also manually deploy or redeploy TAFJJEE_EAR.

Applications □ New Application □ New Enterprise Application

Select local file deployment option and the **TAFJJEE_EAR.ear** file under % TAF_HOME %\appserver folder and click on next



Select Detailed - Show all installation options and parameters.

Click Next.

Step 1: Let default values and click Next.

Step 2: Let default values and click Next.

Step 3: Let default values and click Next.



Step 4: add TAFJ shared libraries and T24 shared libraries

The screenshot shows the 'Install New Application' wizard in the Integrated Solutions Console. The left sidebar has a tree view with 'View' (All tasks), 'Welcome', 'Guided Activities', 'Servers' (selected), 'Server Types', 'Clusters', 'DataPower', 'Core Groups', 'Applications' (New Application selected), 'Application Types', 'Services', and 'Resources'. The main panel title is 'Cell=T24Cell01, Profile=Dmgr_T24' and 'Install New Application'. It says 'Specify options for installing enterprise applications and modules.' and 'Step 1 Select installation options' (disabled). The current step is 'Step 4: Map shared libraries'. It says 'Map shared libraries' and 'Specify shared libraries that the application or individual modules reference. These libraries must be defined in the configuration at the appropriate scope.' A table titled 'Reference shared libraries' shows two rows:

Select	Application	URI	Shared Libraries
<input type="checkbox"/>	TAFJJEE_EAR	META-INF/application.xml	T24 shared libraries TAFJ shared libraries

Below the table are steps 5 through 9: Step 5 Map shared library relationships, Step 6 Initialize parameters for servlets, Step 7 Bind listeners for message-driven beans, Step 8 Provide JNDI names for beans, and Step 9 Bind EJB.

Finish the wizard letting the default value and clicking next.

Validate at the end of the installation processing that **TAFJJEE_EAR.ear** application was well installed.

Click Save.

After the deployment

To avoid websphere classloading issue as websphere uses internally axis, a specific classloading policy should be applied for the **WAR class loader policy of TAFJJEE_EAR**.

Select “Class loader for each war file in application”.



The screenshot shows the WebSphere Integrated Solutions Console interface. The left sidebar navigation includes 'Welcome', 'Guided Activities', 'Servers', 'Applications' (selected), 'Services', 'Resources', 'Security', 'Environment', 'System administration', 'Users and Groups', 'Monitoring and Tuning', 'Troubleshooting', 'Service integration', and 'UDDI'. The main content area displays the 'Enterprise Applications' configuration for 'TAFJEE_EAR > Class loader'. It shows the 'General Properties' tab with 'Class reloading options' (checkbox for 'Override class reloading settings for Web and EJB modules' and a 'Polling interval for updated files' input field) and 'Class loader order' (radio button selected for 'Classes loaded with local class loader first (parent last)'). The 'Configuration' tab is also visible. A 'Help' panel on the right provides links for 'Field help', 'Page help', and 'More information about this page'.

For the module **TAFJServices.war**, axis libraries have to be loaded from the application and not from websphere libraries.

Select “Classes loaded with local classloader first (parent last).”

The screenshot shows the WebSphere Integrated Solutions Console interface. The left sidebar navigation is identical to the previous screenshot. The main content area displays the 'Enterprise Applications' configuration for 'TAFJEE_EAR > Manage Modules > TAFJServices.war'. It shows the 'General Properties' tab with fields for 'URI' (TAFJServices.war), 'Alternate deployment descriptor' (empty), 'Starting weight' (10000), and 'Class loader order' (radio button selected for 'Classes loaded with local class loader first (parent last)'). The 'Additional Properties' tab is also visible. A 'Help' panel on the right provides links for 'Field help', 'Page help', and 'More information about this page'.



TAFJ servlets

TAFJJEE_EAR.ear file will set MDB (TAFJJEE_MDB.jar) and EJB (TAFJJEE_EJB.jar) to read message from JMS Queues, call T24 and publish response in reply queues.

It also contains a war file to deploy helper servlet.

<http://localhost:8080/TAFJEE>

The screenshot shows a Microsoft Internet Explorer window with the title 'TAFJEE - Home'. The address bar displays 'localhost:8080/TAFJEE/index.html'. The page content includes the TEMENOS logo and tagline 'The Banking Software Company'. Below this, a 'Welcome!' section states: 'Welcome to TAFJ servlets. If you can see this page you have successfully deployed TAFJ Web Application.' Under the 'Diagnostic' heading, there are links for 'tDiag' (Get details about your TAFJ environment) and 'tShow' (Get routine compilation details). The 'Execution' section contains links for 'Execute servlet' (Post message to ExecQueue), 'Entry points' (Learn how to interact with TAFJEE), and 'DBTools' (Execute DBTools command). The 'Troubleshooting' section lists 'Technical monitor' (Monitor your TAFJEE application), 'Log files' (Change log level and view log files), and 'Como files' (View como files). At the bottom, it says 'Webapp version:DEV.201505.0'.

For more information please refer to the **TAFJ-AS TAFJ** documentation.



tDiag Servlet

TAFJJEE_WAR_TAFJ contains a servlet which could be used to get details about TAFJ installation

<http://localhost:8080/TAFJEE/tDiag>

i.e.

The screenshot shows a web browser window with the URL <http://localhost:8080/TAFJEE/tDiag>. The page title is "tDiag servlet".

TAFJ installation details

tafj home directory	D:\Temenos\Reference\3rdParty\as\jboss-4.2.3.GA\..\..\..\Temenos\TAFJ
tafj version	DEV_201504.0
tafj properties file	D:\Temenos\Reference\3rdParty\as\jboss-4.2.3.GA\..\..\..\Temenos\TAFJ\conf\MB.properties
tafj trace properties file	D:\Temenos\Reference\3rdParty\as\jboss-4.2.3.GA\..\..\..\Temenos\TAFJ\conf\TAFJTrace.properties
tafj log directory	D:\Temenos\Reference\3rdParty\as\jboss-4.2.3.GA\..\..\..\Temenos\TAFJ\log
T24 log directory	D:\Temenos\Reference\3rdParty\as\jboss-4.2.3.GA\..\..\..\Temenos\TAFJ\log_T24

Environment details

java home	D:\Temenos\Reference\3rdParty\java\jdk1.7.0_60-64\jre
java vendor	Oracle Corporation
java version	1.7.0_60
os arch	amd64
os name	Windows 7
hostname	wkshome01
ip address	10.42.201.58
runtime separator	Double Byte SysSeparator
temn.tafj.appserver.name	JBOSS

TAFJ Runtime properties

temn.tafj.locking.mode	JDBC
------------------------	------

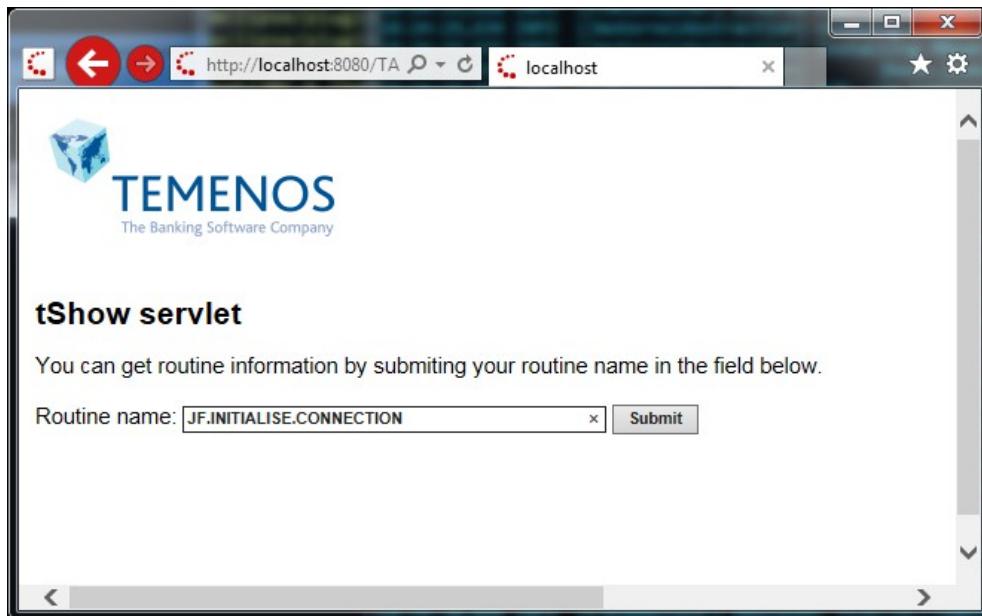
tShow Servlet

TAFJJEE_WAR_TAFJ contains a servlet which could be used to get compilation details about a specific routine

<http://localhost:8080/TAFJEE/tShow>

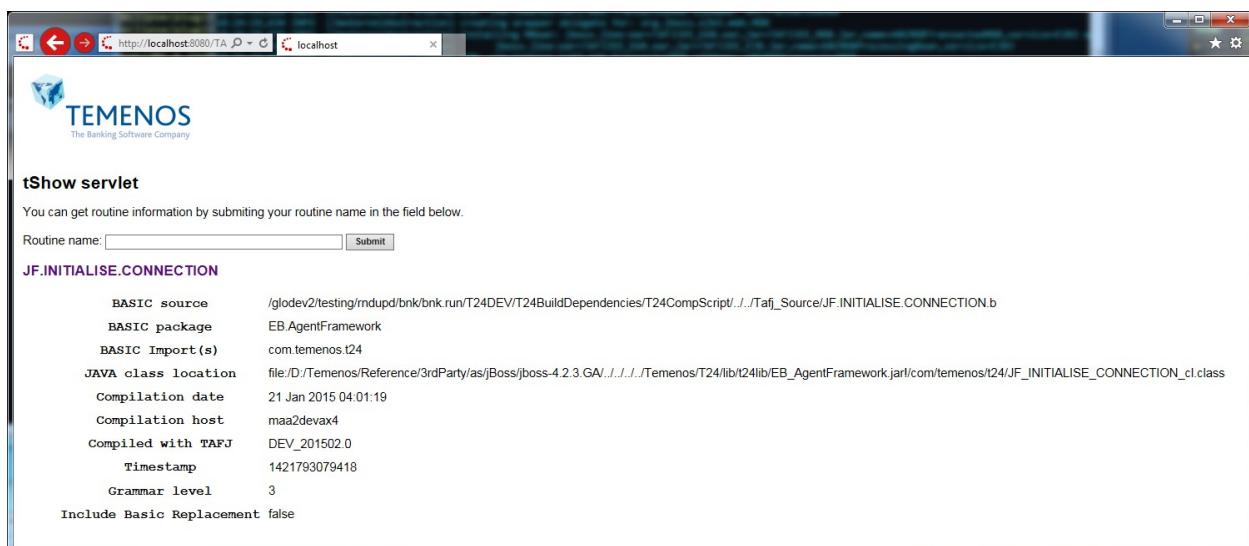
i.e.

to get details about JF.INITIALISE.CONNECTION



You can get routine information by submitting your routine name in the field below.

Routine name:



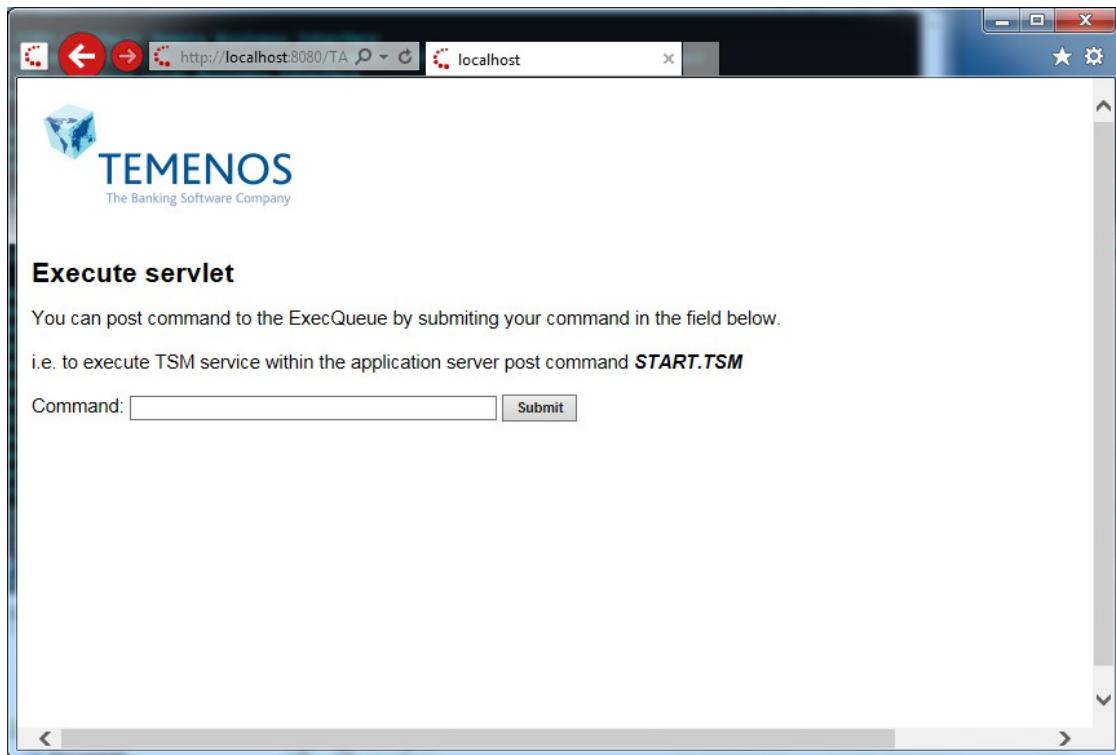
Parameter	Value
BASIC source	/glodev2/testing/rndupd/bnk/bnk.run/T24DEV/T24BuildDependencies/T24CompScript/..../Tafj_Source/JF_INITIALISE.CONNECTION.b
BASIC package	EB.AgentFramework
BASIC Import(s)	com.temenos.t24
JAVA class location	file:/D:/Temenos/Reference/3rdParty/as/jBoss/jboss-4.2.3.GA/../../../../Temenos/T24/lib/t24lib/EB_AgentFramework.jar!/com/temenos/t24/JF_INITIALISE_CONNECTION_ci.class
Compilation date	21 Jan 2015 04:01:19
Compilation host	maa2devax4
Compiled with TAFJ	DEV_201502.0
Timestamp	1421793079418
Grammar level	3
Include Basic Replacement	false



Execute Servlet

TAFJJEE_WAR_TAFJ contains a servlet which could be used to post message to the JMS queue t24ExeqQueue.

<http://localhost:8080/TAFJEE/Execute>



i.e.

to post START.TSM submit START.TSM in the form.

Remark:

The following properties need to be set correctly to run TAFJ within an application server context.

temn.tafj.runtime.phantom.as.process = false

Other TAFJEE functionalities

TAFJEE application offers many other functionalities like monitoring, changing log level, como viewer... please refer to TAFJ AS documentation for detailed information.

T24 Application deployment - BrowserWeb.war

T24 application – BrowserWeb.war deployment can be run automatically.

The script is located at %TAFJ_HOME%\appserver\was\jython\T24applicationDeployment.py

It will execute the following action:

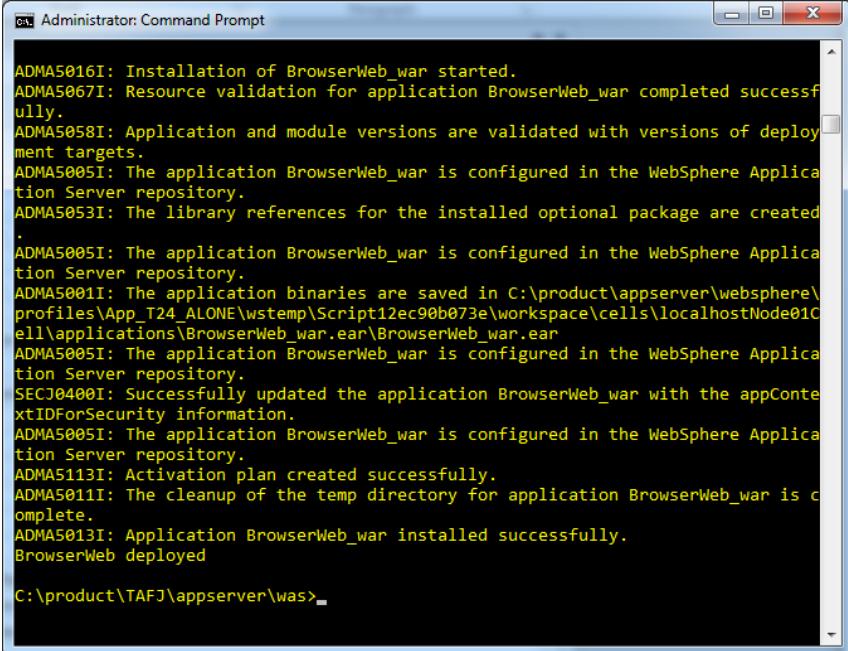
- Deploy Browserweb.war

Before running it you have to define some parameters such as **node**, **cell** and **server name** to set the scope of the configuration in the %TAFJ_HOME%\appserver\was\jython\tafj.py file. You must also define the **browserWebHome** as root folder to locate the Browserweb.war to be deployed i.e. :

```
browserWebHome="C:/product/appserver/websphere/TAFJ/appserver"
```

Launch the following command in the command line (refer to Environment configuration for additional command parameters).

```
wsadmin -lang python -f T24applicationDeployment.py
```



```

Administrator: Command Prompt
[...]
ADMA5016I: Installation of BrowserWeb_war started.
ADMA5067I: Resource validation for application BrowserWeb_war completed successfully.
ADMA5058I: Application and module versions are validated with versions of deployment targets.
ADMA5005I: The application BrowserWeb_war is configured in the WebSphere Application Server repository.
ADMA5053I: The library references for the installed optional package are created.
.
ADMA5005I: The application BrowserWeb_war is configured in the WebSphere Application Server repository.
ADMA5001I: The application binaries are saved in C:\product\appserver\websphere\profiles\App_T24_ALONE\wstemp\Script12ec90b073e\workspace\cells\localhostNode01Cell\applications\BrowserWeb_war.ear\BrowserWeb_war.ear
ADMA5005I: The application BrowserWeb_war is configured in the WebSphere Application Server repository.
SEC30400I: Successfully updated the application BrowserWeb_war with the appContentForSecurity information.
ADMA5005I: The application BrowserWeb_war is configured in the WebSphere Application Server repository.
ADMA5113I: Activation plan created successfully.
ADMA5011I: The cleanup of the temp directory for application BrowserWeb_war is complete.
ADMA5013I: Application BrowserWeb_war installed successfully.
BrowserWeb deployed
C:\product\TAFJ\appserver\was>_

```

You have to stop and restart your server to start the application. You can then browse the administrative console to check it has been correctly deployed and started.

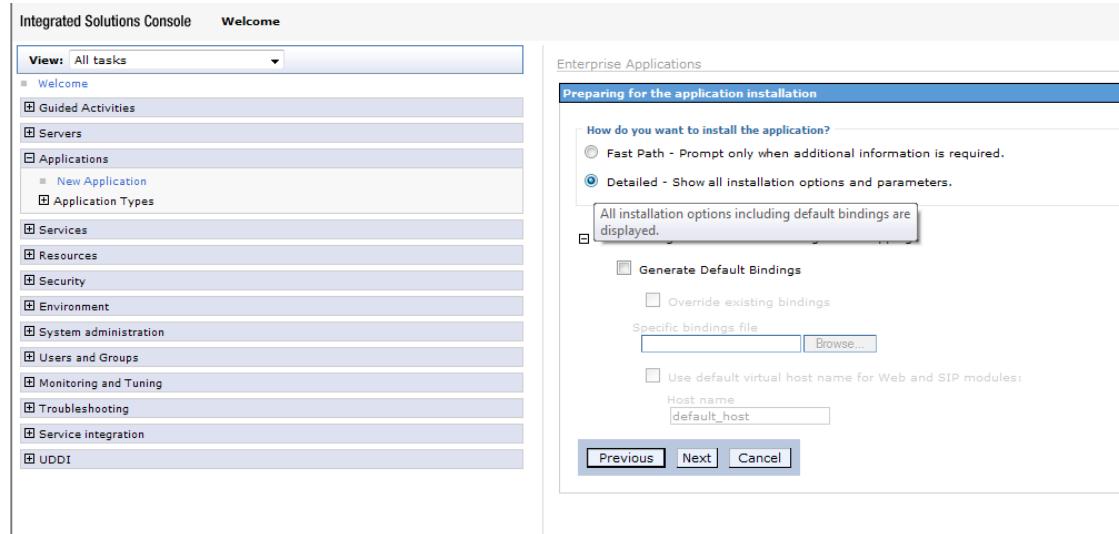


Console mode

You could also manually deploy or redeploy BrowserWeb.war.

With the admin console, under Applications\New Application\New Enterprise Application

Select local file deployment option and the browser.war file click on next



Select Detailed - Show all installation options and parameters.

Click Next.

Step 1: Let default values and click Next.

Step 2: Let default values and click Next.

Step 3: Let default values and click Next.

Step 4: Let default values and click Next.

Step 5: Let default values and click Next.

Step 6:

bind: jms/jmsConnectionFactory to bind: queue/t24OFSQueue bind: queue/t24OFSReplyQueue to	jms/t24ConnectionFactory to jms/t24OFSQueue jms/t24OFSReplyQueue
--	---



Select	Module	EJB	URI	Resource Reference	Target Resource JNDI Name
<input type="checkbox"/>	BrowserWeb		BrowserWeb.war,WEB-INF/web.xml	jms/jmsConnectionFactory	jms/t24ConnectionFactory Browse...

Select	Module	EJB	URI	Resource Reference	Target Resource JNDI Name
<input type="checkbox"/>	BrowserWeb		BrowserWeb.war,WEB-INF/web.xml	queue/t24OFSQueue	jms/t24OFSQueue Browse...

Step 7: Let default values and click Next.

Step 8:

Context Root : /BrowserWeb

Web module	URI	Context Root
BrowserWeb	BrowserWeb.war,WEB-INF/web.xml	/BrowserWeb

Step 9: Let default values and click Next.

Click Next and Finish

In the Enterprise Applications page, select BrowserWeb application and click on start.

Browser the URL : http://IP_Add:9080/BrowserWeb/servlet/BrowserServlet

Where IP_Add is the IP address of the WebSphere Server.



Websphere Properties for T24/TAFJ

Setup Multiple Servants (zOS)

Number of servants setup

The screenshot shows the IBM WebSphere Application Server Administration Console interface. The left sidebar navigation menu includes 'View: All tasks', 'Welcome', 'Guided Activities', 'Servers' (selected), 'New server', 'Server Types' (WebSphere application servers, WebSphere proxy servers, Generic servers, Version 5 JMS servers, WebSphere MQ servers, Web servers), 'Clusters', 'DataPower', 'Core Groups', 'Applications', 'Jobs', 'Services', 'Resources', 'Security', 'Environment', 'System administration', 'Users and Groups', 'Monitoring and Tuning', 'Troubleshooting', 'Service integration', and 'UDDI'. The main content area displays the 'Application servers > w8sr01a > Server Instance' configuration page. It shows the 'General Properties' section with the 'Multiple Instances Enabled' checkbox checked, and 'Minimum Number of Instances' and 'Maximum Number of Instances' both set to 2. Below these fields are 'Apply', 'OK', 'Reset', and 'Cancel' buttons. A 'Help' panel on the right provides field and page help information.

You need to set the property below so threads will go evenly to each servant for an IC.COB



The screenshot shows the 'Custom properties' page for the w8sr01a server. The left sidebar shows various navigation options like 'Servers', 'Clusters', 'Applications', etc. The main panel displays a table with one row:

Select	Name	Value	Description
<input type="checkbox"/>	globalConnectionTypeOverride	true	

Total 1

Set Global transaction to unshared (Only needed for XA Driver)

With TAFJ and T24 we do not want our connection shared across threads. Set the following :

[**Data sources > TAFJ XA DB2 Data source > Connection pools > Custom properties**](#)

globalConnectionTypeOverride=unshared

The screenshot shows the 'Custom properties' dialog for the globalConnectionTypeOverride entry. The dialog has the following fields:

Select	Name	Value
<input type="checkbox"/>	globalConnectionTypeOverride	unshared

General Properties

* Name: globalConnectionTypeOverride
* Value: unshared
Description: Set by Marc on 08/06 in order

Buttons: Apply, OK, Reset, Cancel

Set Data Source mutithread for DB2 (Only needed for XA Driver)

Enable multithreaded access detection:

[**Data sources > TAFJ XA DB2 Data source > WebSphere Application Server data source properties**](#)

General Properties

Statement cache size

statements

- Enable multithreaded access detection
- Enable database reauthentication
- Log missing transaction context
- Non-transactional data source

Set Data Source Isolation for DB2

We want DB2 to be in READ_COMMITTED mode (no dirty reads)

[**Data sources > TAFJ XA DB2 Data source > Custom properties \(page 2\)**](#)

webSphereDefaultIsolationLevel = "2"

<input type="checkbox"/>	webSphereDefaultIsolationLevel	2	Specifies a default transaction isolation level for new connections. Resource References and Access Intents override this value. To configure a default transaction isolation level, use the constants defined by JDBC: 1 (READ UNCOMMITTED), 2 (READ COMMITTED), 4 (REPEATABLE READ), 8 (SERIALIZABLE).
--------------------------	--	---	--

General Properties

* **Scope**

Required

Name

Value

Description

Specifies a default transaction isolation level for new connections. Resource References and Access Intents override this value. To configure a default transaction isolation level, use the constants defined by JDBC: 1 (READ UNCOMMITTED), 2 (READ COMMITTED), 4 (REPEATABLE READ), 8 (SERIALIZABLE).

Type

Kill clean thread of Appserv

T24 has SLEEPS in it. Websphere will detect this and kill the thread unless this property is set.

[Application servers > w1sr01 > Administration > Custom properties](#)

server_region_stalled_thread_threshold_percent=100 instead of 0

<input type="checkbox"/>	<u>server_region_stalled_thread_threshold_percent</u>	100	Set this value to 100 in order to avoid the hung thread detection, Marc (08/06)
--------------------------	---	-----	---

Remove end point delivery message**Stop endpoint if message delivery fails = uncheck**

[***Activation specifications***](#) > "[**T24 Execute Message MDB MQ**](#)" > **Advanced properties**

Additional Fail JMS method calls if the queue manager is quiescing Stop endpoint if message delivery fails**Number of sequential delivery failures before suspending endpoint****Auto Reconnection database (Only needed for XA Driver)****Enable database reauthentication**

[**Data sources**](#) > [**TAFJ XA DB2 Data source**](#) > [**WebSphere Application Server data source properties**](#)

General Properties

Statement cache size
 statements

 Enable multithreaded access detection Enable database reauthentication Log missing transaction context Non-transactional data source

- . . .

Validate existing pooled connections (optional)

Connection validation properties

Validate new connections

Number of retries
100

Retry interval
5 seconds

Validate existing pooled connections

Retry interval
3 seconds

Validation options

Query
`SELECT CURRENT SQLID FROM SYSIBM.SYSDUMMY1`

SETUP JInsight (not recommended for production use as this is a profile tool)

[Application servers > w1sr01 > Process definition > Servant > Environment Entries](#)

LIBPATH=/WebSphere/jinsight

General Properties

* Name
LIBPATH

* Value
/WebSphere/jinsight

Description
LIBPATH adjustment required

Buttons: Apply | OK | Reset | Cancel

[Application servers > w1sr01 > Process definition > Servant > Java Virtual Machine](#)

-agentlib:jinsight-zOS-64=port=12345

Debug arguments
`-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7777`

Generic JVM arguments
`-agentlib:jinsight-zOS-64=port=12345 -Dtafj.home=/WebSphere/1/AppServer/TAFJ`

Executable JAR file name
[Empty input field]

SETUP Support Assistant (not recommended for production use as this is a profile tool)

Setup Load Balance in Queue with Multiple Servant

[Application servers > w1sr01 > Custom properties](#)

server_region_stalled_thread_threshold_percent = 100

server work distribution algorithm = 1

server use wlm to queue work = 0



General Properties

* Name

* Value

Description

Avoid the thread cleaning of Websphere and avoid the disconnection of JDBC

If you are getting the error below, consider this property.

Error: com.ibm.websphere.ce.cm.ObjectClosedException: DSRA9110E:
Statement is closed.

[Application servers > tesr01a > ORB service > z/OS additional settings](#)

Container Settings

- [Session management](#)
- + [SIP Container Settings](#)
- + [Web Container Settings](#)
- + [Portlet Container Settings](#)
- + [EJB Container Settings](#)
- [Container Services](#)
 - [Application profiling service](#)
 - [Transaction Service](#)
 - [Dynamic cache service](#)
 - [Compensation service](#)
 - [Internationalization Service](#)
 - [Default Java Persistence API settings](#)
 - [Object pool service](#)
 - [ORB service](#)
 - [Startup beans service](#)
- + [Business Process Services](#)

[Application servers](#) > [tesr01a](#) > [ORB service](#) > z/OS additional settings

Workload manager timeout = 0

General Properties

* **ORB listener keep alive**
 seconds

* **ORB SSL listener keep alive**
 seconds

* **Workload manager timeout**
 seconds

Workload profile
 ▾

Websphere Performance Properties

UAT / Production Memory setting

For a UAT and production environment you must refine your memory setting depending on your expected number of sessions and tSA.

A session / tSA memory impact vary depending on the job done, it will be at least 20MB and could be up to 60MB.

When planning to run for example 80 interactive sessions and 40 tSA on a server, you could dimension the **max memory for sessions** by applying:

120 sessions * average 50MB = 6G max heap size.

You will have **-Xmx6G**

A good practice could be to set initial heap size to same value **-Xms6G**

If you are interested to monitor that more precisely, you could use a monitoring tool like Visual VM.

Take a heapdump when running the expected jobs, isolate the jSession objects and compute the retained size.

This is for one session.

Class Name	Instances [%]	Instances	Size	Retained ▾
com.temenos.tafj.common.jSession		1 (0%)	691	21,026,378 (19.2%)

This for 16 sessions.

Instances: 16 Instance size: 691 Total size: 11,056 Retained size: 376,874,718	376,874,718 (74%)
--	-------------------

You could also simply use TAFJ technical monitor to follow the memory evolution graph over the time and refine your setup.

Memory errors

Java.lang.OutOfMemoryError: Java heap space

Increase **-Xmx** max heap size parameter

Java.lang.OutOfMemoryError: PermGen space

Increase -XX:MaxPermSize max perm gen parameter

JVM parameters

Snapshot of the heap to analyse the content when getting a OOM error.

-XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/some/path/

Garbage collection logs, detailed information about Garbage Collection, could be applied as there is a low overhead, display the amount of memory released

-XX:+PrintGC or -verbose:gc

Print messages at garbage collection, simple logging mode, i.e.

[GC 370562K->208870K(964096K), 0.0138438 secs]

[Full GC 174246K->81336K(853504K), 0.7733941 secs]

-XX:+PrintGCDetails

Same as above but print more details at garbage collection, differs depending on GC algorithm.

-Xloggc:<file> equivalent to -XX:+PrintGC -XX:+PrintGCTimeStamps

Log GC verbose output to specified file with time and date information

Set Performance Improvements for zOS

Note: If you do not do the below, you will not see scalability with TAFJ (will scale to only around 67% CPU usage with 16-20 threads depending on what you are doing). The following params will help scale TAFJ much better on a WAS zOS implementation. Results might vary on other OS platforms.

Application Servers> server1>(under Java and Process Management)>Process Definition>Java Virtual Machine

Change initial heap size to something large like 8Gig.



Set up Garbage Collection to gencon

```
-Dtafj.home=${TAFJ_HOME} -Xtrace:none -Dcom.ibm.xml.xlpx.jaxb.opt.level=3  
-Xgcpolicy:gencon -Xnoclassgc -Xlp1M
```

From doc “TEMENOS T24z_Technical Report”, “-Xlp1M” means:

- 10 GB for the JVM with Large Frames (1 MB Frame instead of 4 K Frames)

Statement Cache Size

Set cache size high enough so number of prepares in DB2 decreases.

Stored Procedure Called for Duplicate Key

This especially needs to be done for the t24LockingDatasource!

From doc "TEMENOS T24z_Technical Report":

3.3.2 Sqlcode -803

During IC.COB we found SQLCAMESSAGE stored procedure called each time a duplicate key was inserted. This created increase in class 1 elapsed time. In the Websphere data source we found retrieveMessagesFromServerOnGetMessage property was set to true. We set it to false, the SQLCAMESSAGE stored procedure was not called anymore.

[**Data sources > DB2 Universal JDBC Driver DataSource > Custom properties**](#)

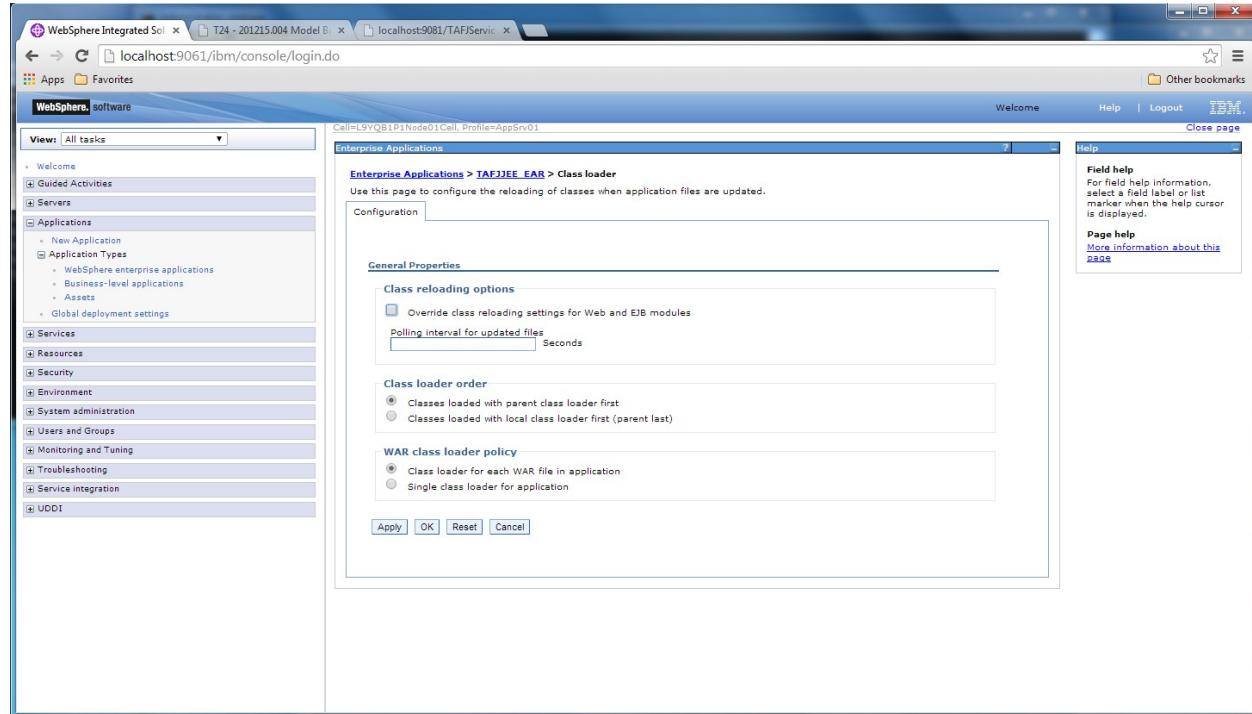
[retrieveMessagesFromServerOnGetMessage=false](#)

Trouble Shooting

TAFJServices.war

To avoid websphere classloading issue as websphere uses internally axis, a specific classloading policy should be applied for the **WAR class loader policy of TAFJJEE_EAR**.

Select “Class loader for each war file in application”.



For the module **TAFJServices.war**, axis libraries have to be loaded from the application and not from websphere libraries.

Select “Classes loaded with local classloader first (parent last).



The screenshot shows the WebSphere Integrated Solutions Console interface. The left sidebar navigation bar includes links for Welcome, Guided Activities, Servers, Applications (with sub-options like New Application, Application Types, WebSphere enterprise applications, Business-level applications, Assets, Global deployment settings), Services, Resources, Security, Environment, System administration, Users and Groups, Monitoring and Tuning, Troubleshooting, Service integration, and UDDI.

The main content area displays the "Enterprise Applications" configuration page for the application "TAFJServices.war". The URL field is set to "TAFJServices.war". The "Starting weight" is set to "10000". The "Class loader order" dropdown is set to "Classes loaded with local class loader first (parent last)". The right side of the screen shows additional properties and help links.



Where is my Browser port?

- It is the first line here (in this case the port is 9080 where the Browser servlet is?)

Select	Host Name	Port
□		9080
□		80
□		9443
□		5060
□		5061
□		443

Websphere is killing threads because the T24 code has a sleep in it.

Application Servers->server->Custom properties (under Administration)->New

Name: server_region_stalled_thread_threshold_percent

Value: 100

Websphere has J2CA0081E: Method cleanup failed while trying to execute method cleanup on ManagedConnection

Your threads are sharing connections. You want a dedicated connection per thread. This property has no effect for Oracle on windows, or DB2 on windows

Datasources->Your datasource name->custom properties

Add a property:

Name: globalConnectionTypeOverride

Value : unshared



Where is my Default Messaging port for the connection factory for T24Bus?

Under Buses/T24Bus/Bootstrap members. This allows you to connect with a JMS client.

The screenshot shows the Integrated Solutions Console interface in Mozilla Firefox. The URL in the address bar is <http://10.44.5.224:9546.ibm/console/login.do>. The left sidebar navigation tree includes Guided Activities, Servers, Applications, Services, Resources (with JMS, JDBC, Resource Adapters, etc.), Security, Environment, System administration, Users and Groups, Monitoring and Tuning, Troubleshooting, and Service integration (with Buses selected). The main content area displays the 'Buses' section under 'T24Bus > Bootstrap members'. It shows a table with one row:

Name	Type	SIB Service	Host	Ports
cisr01a	Bus member	Enabled	ukhml-t24z17.europe.temenosgroup.com	9550,9551

Total 1

On the right side of the page, there are 'Field help', 'Page help', and 'Command Assistance' sections.



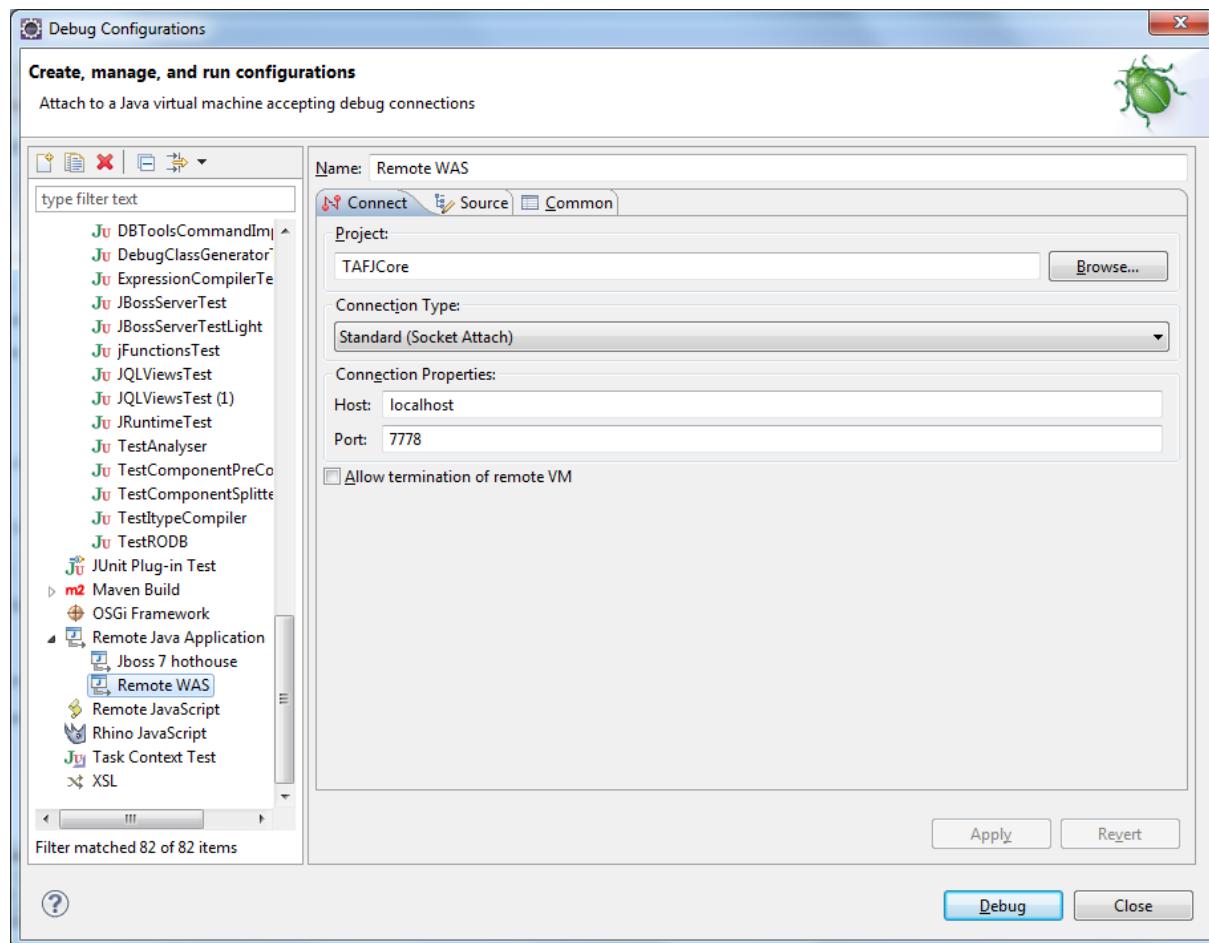
Tools

Debugging WAS through remote Eclipse

Put the following as JVM startup parameters.

```
-Xdebug -Xnoagent -Djava.compiler=NONE
-Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=7778
```

Define an Eclipse Remote Java Application configuration.



Health Center

(Note that fix pack 7.0.0-WS-WASSDK-WinX64-FP0000011.pak must be applied for this to work, otherwise your server will crash on startup with an invalid JVM option. To correct it, you must modify server.xml)



Download IBM Support Assistant Workbench and under Update->Find New->Tools Add On menu, choose JVM Based Tools and then select Health Center. Next, start Health Center through the below screen on Websphere adding the –Xhealthcenter option to the Generic JVM args.

The screenshot shows the 'Java Virtual Machine' configuration page within the 'Integrated Solutions Console'. The 'Runtime' tab is active. Key visible fields include:

- General Properties:** Classpath, Root Classpath.
- Additional Properties:** Custom properties.
- Verbose Options:** Verbose class loading, Verbose garbage collection, Verbose J2EE.
- Memory Settings:** Initial heap size (256 MB), Maximum heap size (1024 MB).
- Diagnostic Tools:** Run HProf, HProf Arguments.
- Debug Arguments:** Agentid (jdp-transport=ptl_socket,server,suspend=n,address=7777), Generic J2EE arguments (-Xhealthcenter: /opt/ibm/wasapps/TAFJ/TAFJ -Dcom.ibm.websphere.management.jmxremote.port=9999 -Dcom.sun.management.jmxremote.authenticate=false -Djava.rmi.server.hostname=10.3.42.3), Executable JAR file name.
- Operating System:** Disable JIT, Operating system name (aIX).

At the bottom are 'Apply', 'OK', 'Reset', and 'Cancel' buttons.

JMX connection setup zOS

Add to the JVM the following properties

```
-Dtajf.home=/temenos/tajf -Xtrace:none -Dcom.ibm.xml.xlpx.jaxb.opti.level=3  
-Xgcpolicy:gencon -Xnoclassgc -Xlp1M -Djavax.management.builder.initial=  
-Dcom.sun.management.jmxremote -Djava.rmi.server.hostname=10.3.42.3  
-Xdump:java:events=throw,filter=java/lang/NullPointerException[jVarImpl.verifyFileDescriptor  
]
```

And In management.properties (common settings for all JVMs)

```
com.sun.management.jmxremote.port=9999  
com.sun.management.jmxremote.authenticate=false  
com.sun.management.jmxremote.ssl=false
```

Appendix

Connect Browser to a Remote WAS

File : BrowserWeb.war/WEB-INF/web.xml

```
<resource-ref id="ResourceRef_jmsConnectionFactory">
    <description>Used to get connections to JMS for OFS
queues</description>
    <res-ref-name>iiop://\[IP\]:\[PORT\_IIOP\]/jmsConnectionFactory</res-ref-
name>
    <res-type>javax.jms.ConnectionFactory</res-type>
    <res-auth>Container</res-auth>
</resource-ref>
```

MQ manager and related object creation

The below is the script used to create the MQ manager and its associated object to be bound with Websphere. The scripts are available at the %TAFJ_HOME%/appserver/was folder.

```
# UNIX Only
#
# execute the commands to create The Queue Manager , the Channel and the mandatory
queues.
#
# Add to mqm group all user need access to the queues including user running the
Application Server.

# define variable for Queue Manager Name.
export MQNAME=P2CBRMQ1

# Creation of the Queue manager.
crtmqm -q $MQNAME

# the Queue manager.Start Queue
strmqm $MQNAME

# run console for the queue manager
runmqsc $MQNAME

# In the console copy these commands
DEFINE LISTENER('LISTENER.TCP') TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
START LISTENER('LISTENER.TCP')
DEFINE CHANNEL('SYSTEM.ADMIN.SVRCONN') CHLTYPE(SVRCNN)
ALTER QMGR CHLAUTH(DISABLED)
```

REFRESH SECURITY

```
DEFINE QLOCAL('t24BROWSERQueue') SHARE
DEFINE QLOCAL('t24BROWSERReplyQueue') SHARE
DEFINE QLOCAL('t24OFSQueue') SHARE
DEFINE QLOCAL('t24OFSReplyQueue') SHARE
DEFINE QLOCAL('t24TCIBQueue') SHARE
DEFINE QLOCAL('t24TCIBReplyQueue') SHARE
DEFINE QLOCAL('t24TCIBCORPQueue') SHARE
DEFINE QLOCAL('t24TCIBCORPReplyQueue') SHARE
DEFINE QLOCAL('t24TWSQueue') SHARE
DEFINE QLOCAL('t24TWSReplyQueue') SHARE
DEFINE QLOCAL('t24CALLATQueue') SHARE
DEFINE QLOCAL('t24CALLATReplyQueue') SHARE
DEFINE QLOCAL('t24ARCIBQueue') SHARE
DEFINE QLOCAL('t24ARCIBReplyQueue') SHARE
DEFINE QLOCAL('t24SEATQueue') SHARE
DEFINE QLOCAL('t24SEATReplyQueue') SHARE
DEFINE QLOCAL('t24EXECQueue') SHARE
DEFINE TOPIC('tecEventsTopic') TOPICSTR('/') SHARE
```