T24 Outbound Adapter for IBM Integration Bus

Table of Contents

Introduction

Preface

This document is intended to those who need to set up IBM Integration Bus in integrating T24 with external system and to send messages from an external system to T24.

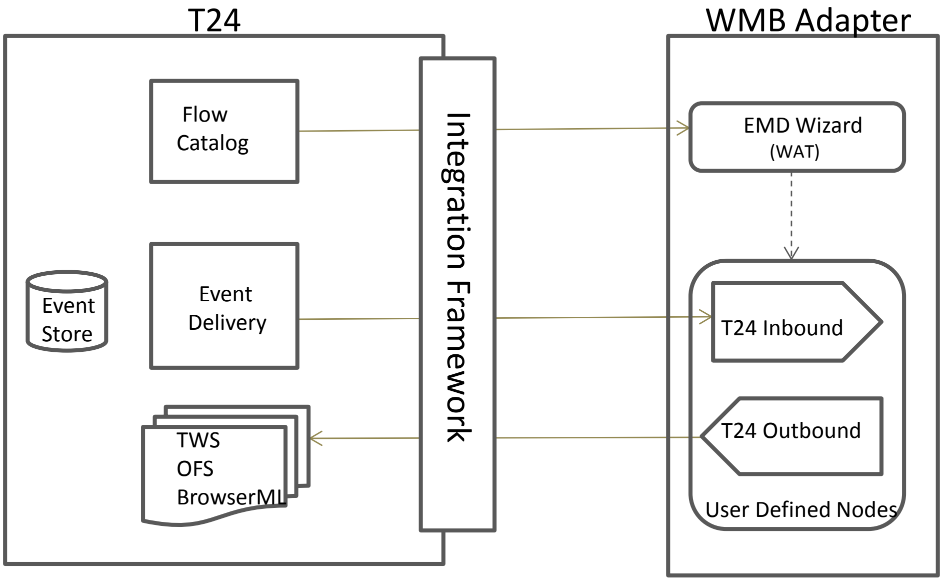
Overview

IBM® Integration Bus is an enterprise service bus (ESB) providing connectivity and universal data transformation for service-oriented architecture (SOA) and non-SOA environments. Now businesses of any size can eliminate point-to-point connections and batch processing regardless of platform, protocol or data format.

To integrate T24 with 3rd party system, T24 Integration Framework provides a flow catalog component along with event delivery component and TWS & OFS components which enables Outbound and outbound integration.

This IBM Integration Bus thus provides an opportunity to integrate T24 with 3rd party systems. For this integration an Adapter is required. This document deals about the T24 Outbound adapter that facilitates the communication from IBM Integration Bus to T24.

The following diagram explains the various components of the T24 Adapters for IBM Integration Bus and T24 Integration Framework.



Prerequisites

Component	Version
T24	R14 with WS Product
IBM Integration Bus (IIB)	9.0
TAFC or TAFJ	R14
T24WMBConnector.rar	V1.0
com.temenos.adapter.wmb.udn_<Version>.jar	V1.0
T24WMBCustomNode.par	V1.0
JDK	1.6
jBoss (For TAFJ)	7.2

Note: Please refer the Appendix ‘Setting up T24 and jboss for TAFJ’ if the runtime is TAFJ.

Assumptions

The document is highly technical in nature and requires knowledge in TWS and IBM products namely IIB and MQ. MQ is used for demo purpose

Installation:

1. Ensure the T24WMBConnector.rar is copied to a location that can be accessible. This is the artefact that allows the IIB Designer to introspect T24.
2. Copy and paste T24 Custom Node (com.temenos.adapter.wmb.udn_<Version>.jar) into the plugins folder of the Integration Toolkit Designer. This is the plugin used during the design time of the flow.
3. Add the T24WMBCustomNode.par into the library path of IIB using the command mqsichangebroker. This is a runtime artefact that allows the IIB flow to connect to T24 and poll for events

   Syntax:

   Mqsichangebroker BrokerName –l <location of the library file>

   Ex.

   Mqsichangebroker T24_Integ_Node –l “D:\WMBLibrary”

Getting Started

T24 Outbound Adapter for IIB receives messages from external systems and sends it to T24 for processing. This Adapter

- Expose operations for native messages. Formats supported are
  - OFS, OFSML, Batch OFS, Batch OFSML

    OFSML – follows a simpler modified object model. This model is followed while coding within Visual Studio.

    Batch OFS and Batch OFSML – standard OFS or OFSML messages sent in an array with each array element containing a single message. This enables a batch of messages to be sent.
- Expose Business services and operations in T24
  - Business services are defined in EB.SERVICE. Business services compose of business operations which are in turn defined in PW.ACTIVITY.
  - A business operation is either a transaction (in simple terms a VERSION + FUNCTION) or a query (ENQUIRY)

T24 Outbound Adapter has to be set up to send messages to T24 and the setup is a 2 step process:

1. Performing Metadata Discovery.
2. Deploying Message Flow

The first step is applicable for ServiceXML only. For other type of request it is enough to create the Message Flow and deploy it to the Message Broker

This document explains the 3 steps in detail.

Before setting up T24 Outbound Adapter for IIB, ensure that

1. TAFC Agent is started or the Application server, where the web service to connect to T24 resides, is started.
2. OFS record in OFS.SOURCE table in T24 is available with attribute field set to PREAUTHENTICATED.
3. A queue manager is created in IBM MQ with the required queues.
4. A Broker which is a set of execution processes that hosts one or more message flows to route, transform, and enrich in messages, is created.

Note: The section “*Performing Metadata Discovery*“ is applicable for ServiceXML Type of Request.

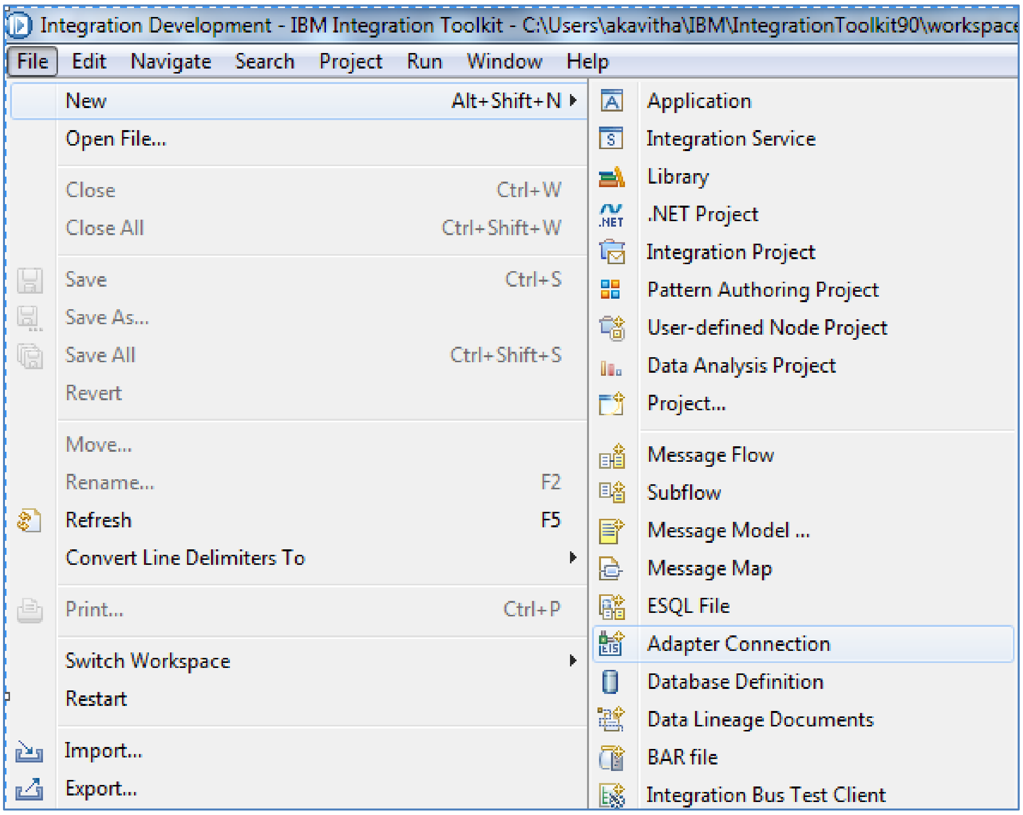
Performing Metadata Discovery:

WMB provides Adapter Connection wizard, a tool that is used to create services. The Adapter Connection wizard establishes a connection to the server, discovers services (based on search criteria that is provided), and generates business objects, interfaces, and import or export files, based on the services that are discovered.

The T24WMBConnector.rar is the metadata discovery adapter that is to be used in the Adapter Connection Wizard to connect to T24 and get the metadata. The metadata discovery wizard has the introspection capabilities for T24 integration framework flow catalogue for business events.

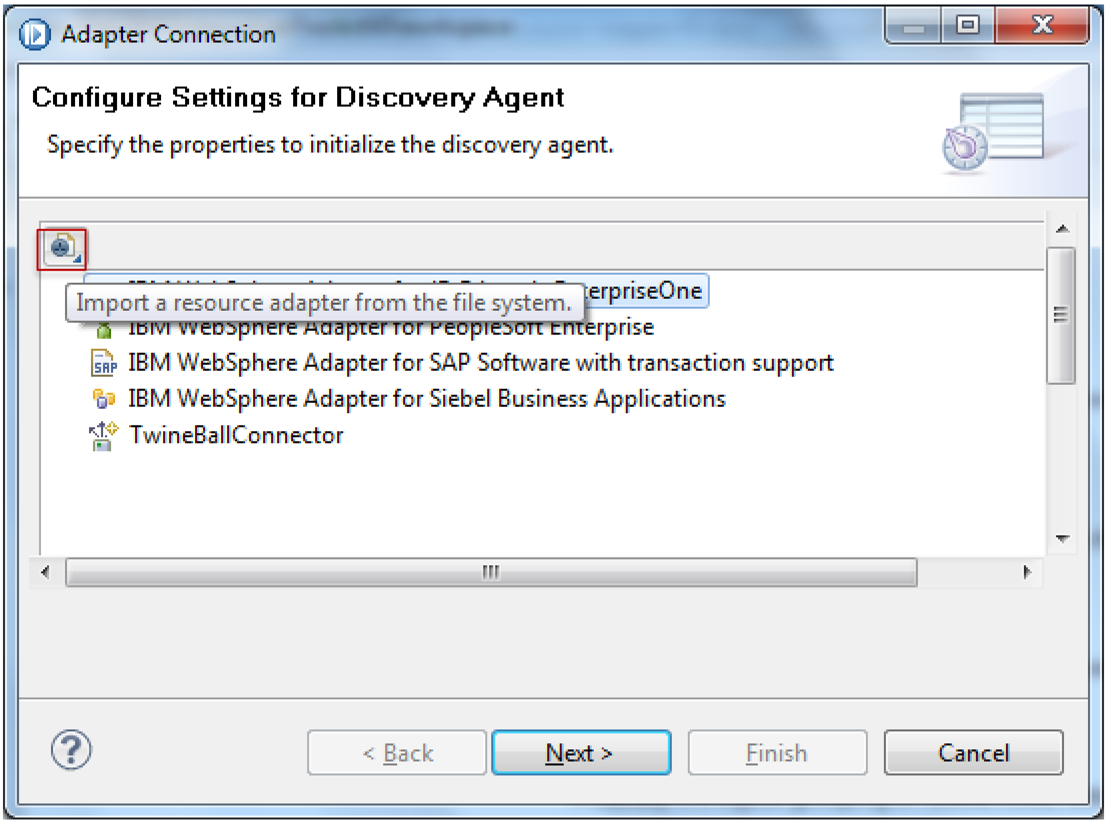
To perform Metadata discovery:

1. Launch IBM Websphere Message Broker Toolkit.
2. Create a new Adapter Connector Project (File → New → Adapter Connection)



The user is taken to the “Configure Settings for Discovery Agent” dialog box.

3. Click on the highlighted portion to add the T24WMBConnector.rar



Configure Settings for Discovery Agent

Specify the properties to initialize the discovery agent.

Import a resource adapter from the file system.

erpriseOne

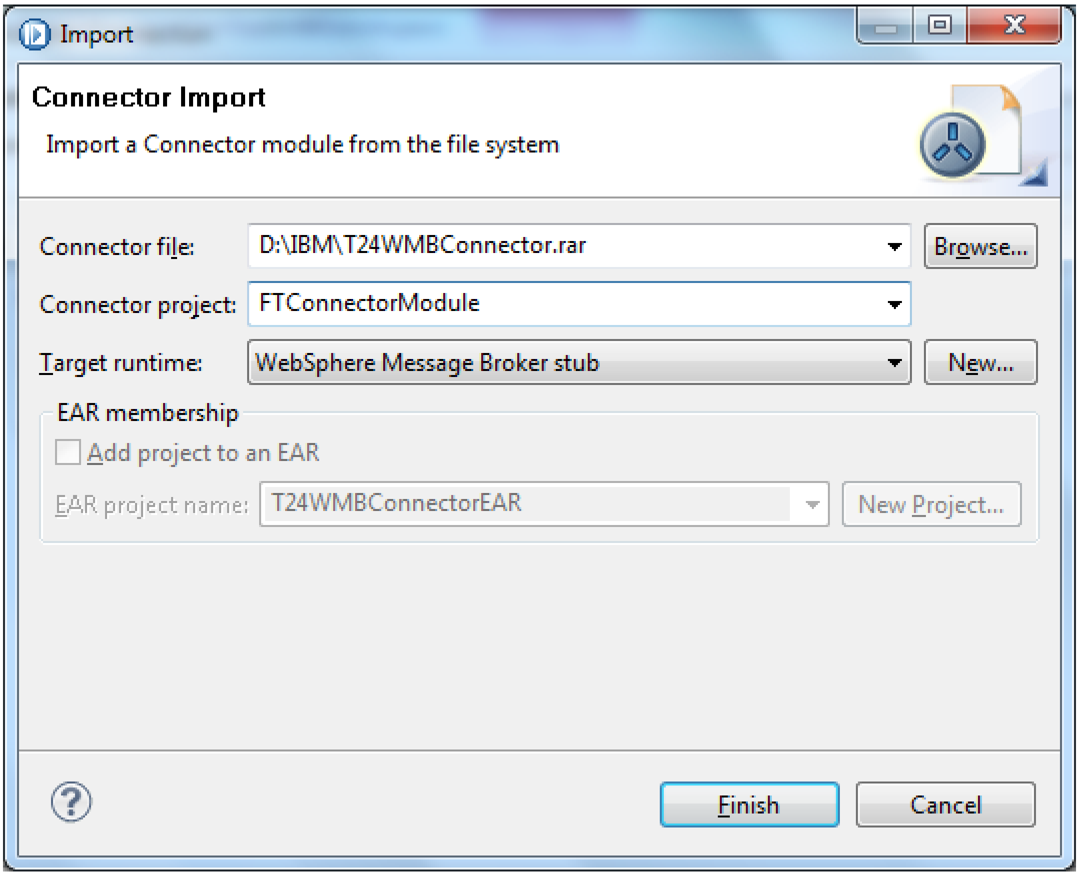
IBM WebSphere Adapter for PeopleSoft Enterprise

IBM WebSphere Adapter for SAP Software with transaction support

IBM WebSphere Adapter for Siebel Business Applications

TwineBallConnector

Search and select the Connector Module (here T24WMBConnector.rar). Once the Connector module is located, specify the name for the Connector Module (the name using which the connector is identified later)



Import

Connector Import

Import a Connector module from the file system

Connector file: D:\IBM\T24WMBConnector.rar

Connector project: FTConnectorModule

Target runtime: WebSphere Message Broker stub

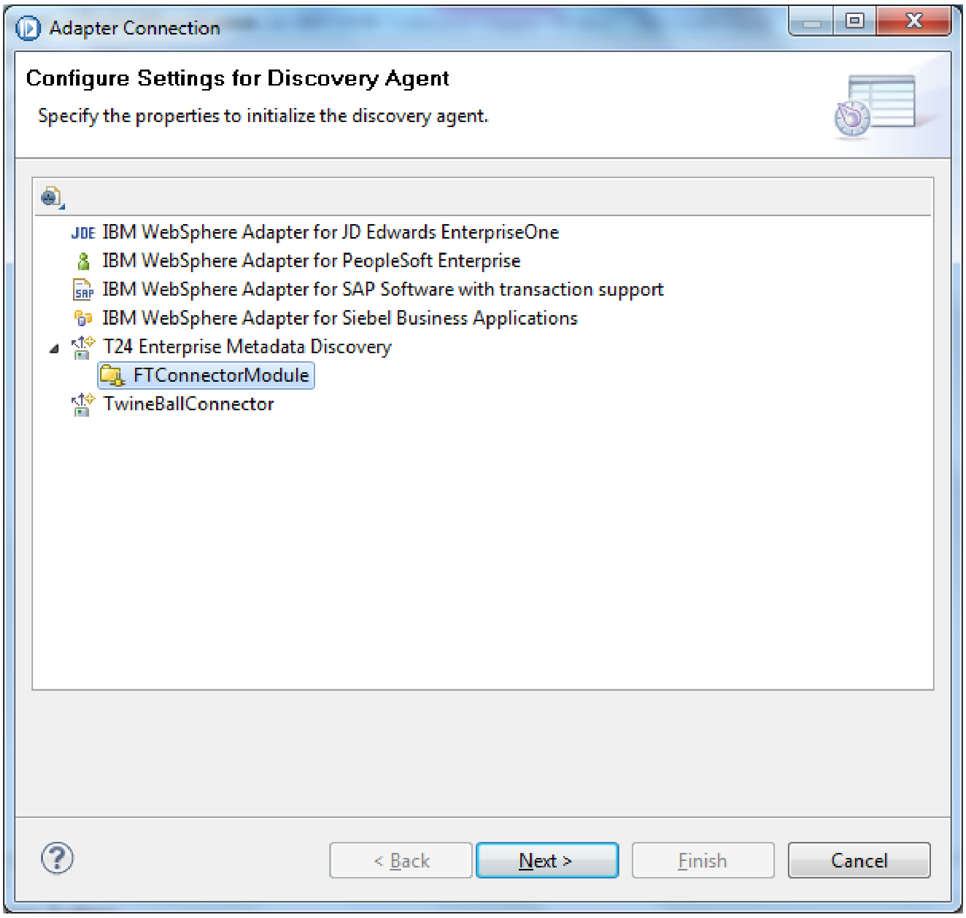
EAR membership

□ Add project to an EAR

EAR project name: T24WMBConnectorEAR

4. Click Finish.

Now the connector module gets displayed in the list of connectors



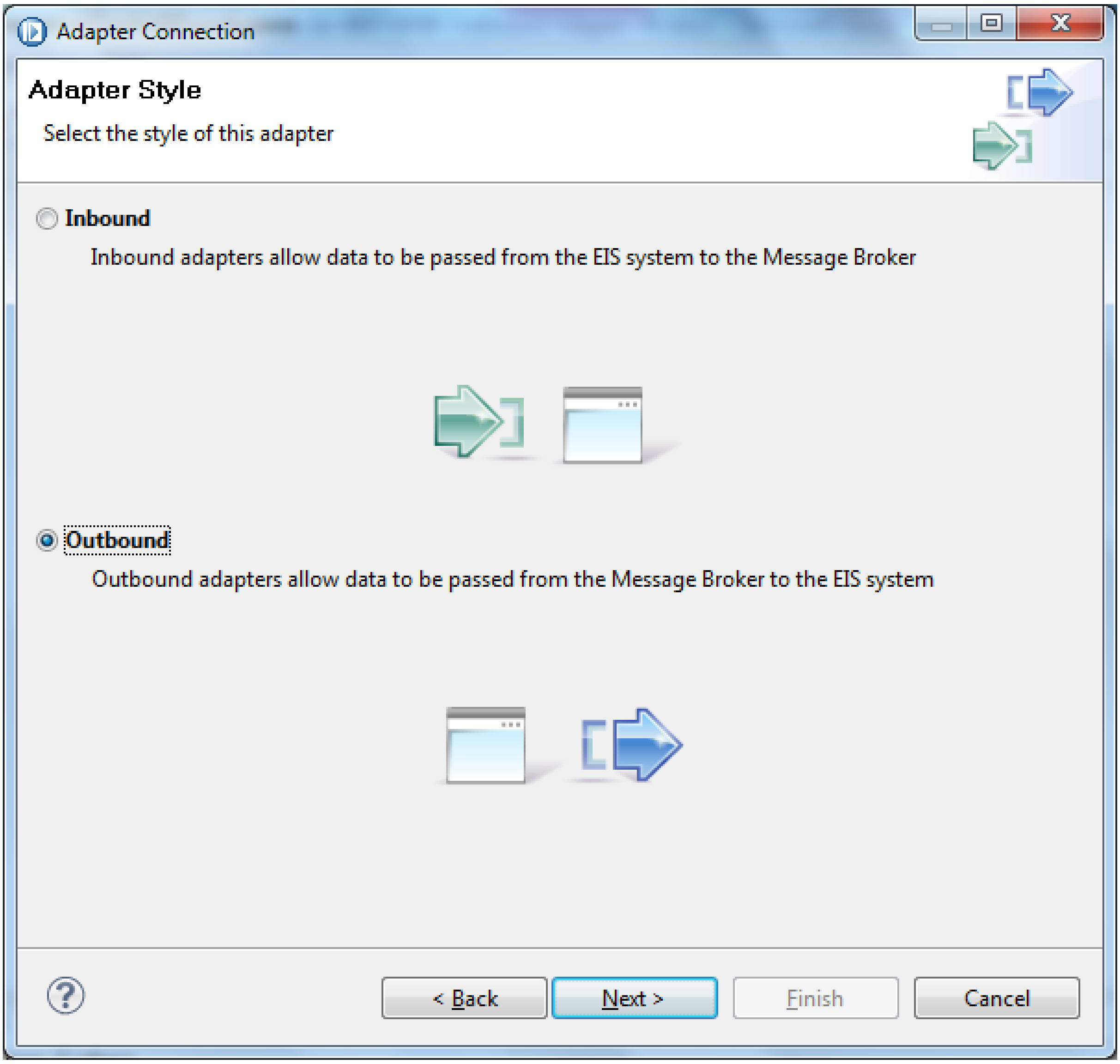
Configure Settings for Discovery Agent

Specify the properties to initialize the discovery agent.

- IBM WebSphere Adapter for JD Edwards EnterpriseOne
- IBM WebSphere Adapter for PeopleSoft Enterprise
- IBM WebSphere Adapter for SAP Software with transaction support
- IBM WebSphere Adapter for Siebel Business Applications
- T24 Enterprise Metadata Discovery
  - FTConnectorModule
- TwineBallConnector

Select the connector module and continue with the Wizard for metadata discovery.

5. The next step is to select the style of the Adapter. In the context of this document “Outbound” needs to be selected.



Adapter Connection

Adapter Style

Select the style of this adapter

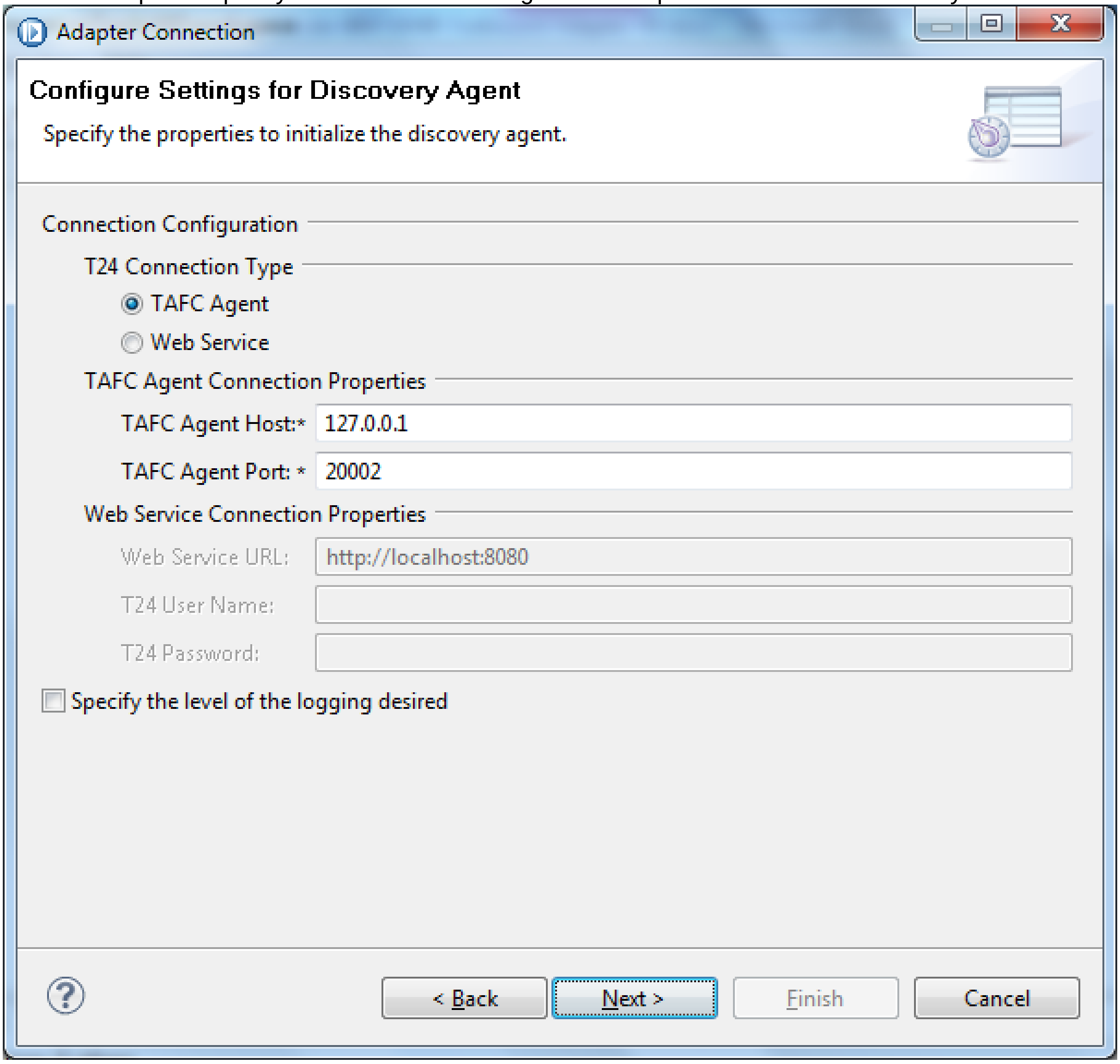
○ **Inbound**

Inbound adapters allow data to be passed from the EIS system to the Message Broker

● **Outbound**

Outbound adapters allow data to be passed from the Message Broker to the EIS system

6. The next step is to specify the connection settings to T24 to perform metadata discovery.



Configure Settings for Discovery Agent

Specify the properties to initialize the discovery agent.

Connection Configuration

T24 Connection Type

● TAFC Agent

○ Web Service

TAFC Agent Connection Properties

TAFC Agent Host:* 127.0.0.1

TAFC Agent Port: * 20002

Web Service Connection Properties

Web Service URL: http://localhost:8080

T24 User Name:

T24 Password:

□ Specify the level of the logging desired

There are two types of connection is supported, classic Agent based connectivity and the Web Service based connectivity especially for TAFJ users.

As the connection mechanism is specified as TAFC agent, the user has to provide the Host and Port number of the TAFC Agent.

The logging configuration mentioned here is the design time logging and currently this feature is diabled.

To use the web service based connectivity option the following components has to be packed as a web archive and then to be deployed in an application server like jBoss.

- IntegrationLandscapeService
- IntegrationFlowService
- CatalogService

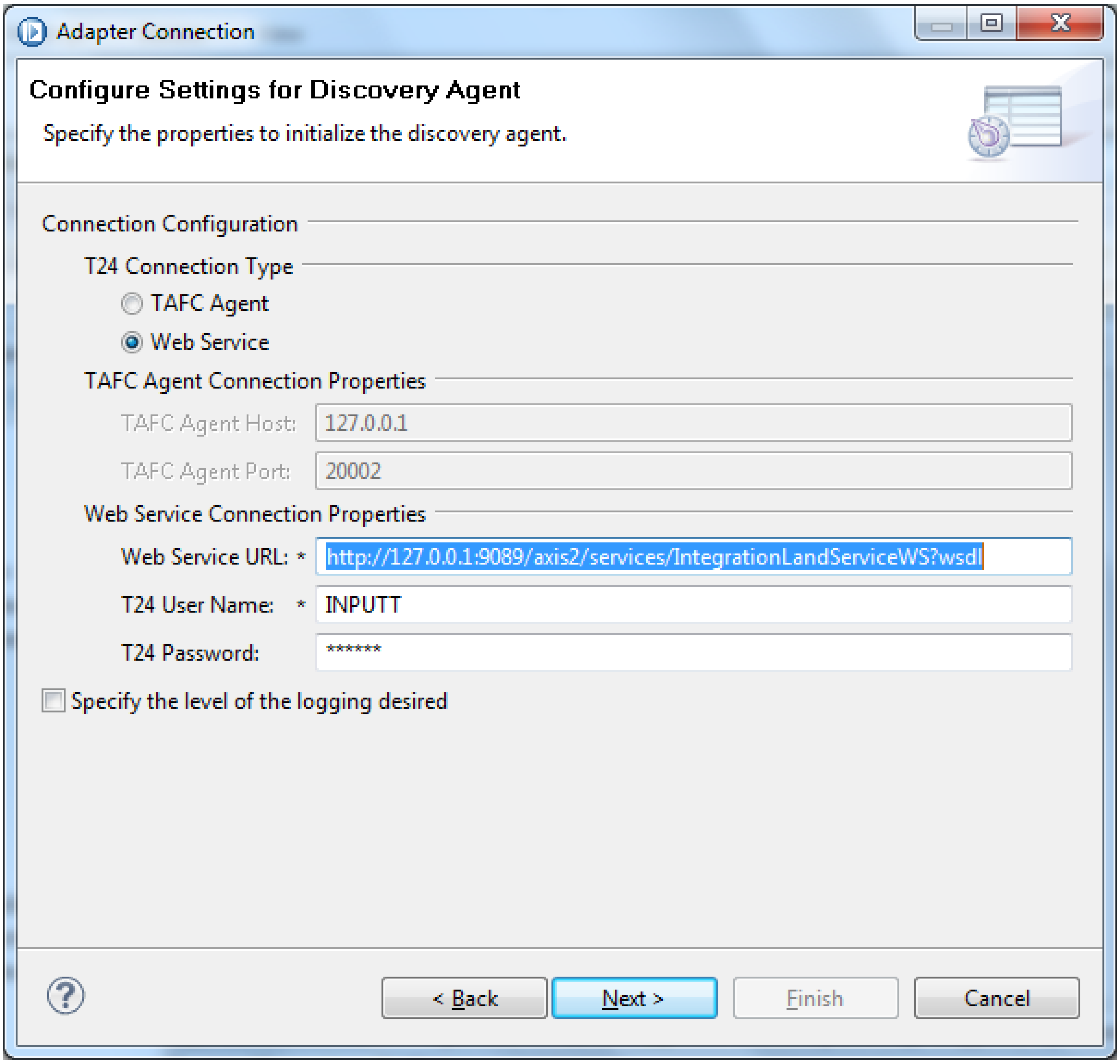
The Integration Landscape Service URL has to be provided along with the user credentials to connect to T24.

The URL will be in this form:

http://<Host Name>:<Port Number>/<context Name>/ services/IntegrationLandServiceWS?wsdl

For ex:

http://127.0.0.1:9089/axis2/services/IntegrationLandServiceWS?wsdl



Configure Settings for Discovery Agent

Specify the properties to initialize the discovery agent.

Connection Configuration

T24 Connection Type

○ TAFC Agent

● Web Service

TAFC Agent Connection Properties

TAFC Agent Host: 127.0.0.1

TAFC Agent Port: 20002

Web Service Connection Properties

Web Service URL: * http://127.0.0.1:9089/axis2/services/IntegrationLandServiceWS?wsdl

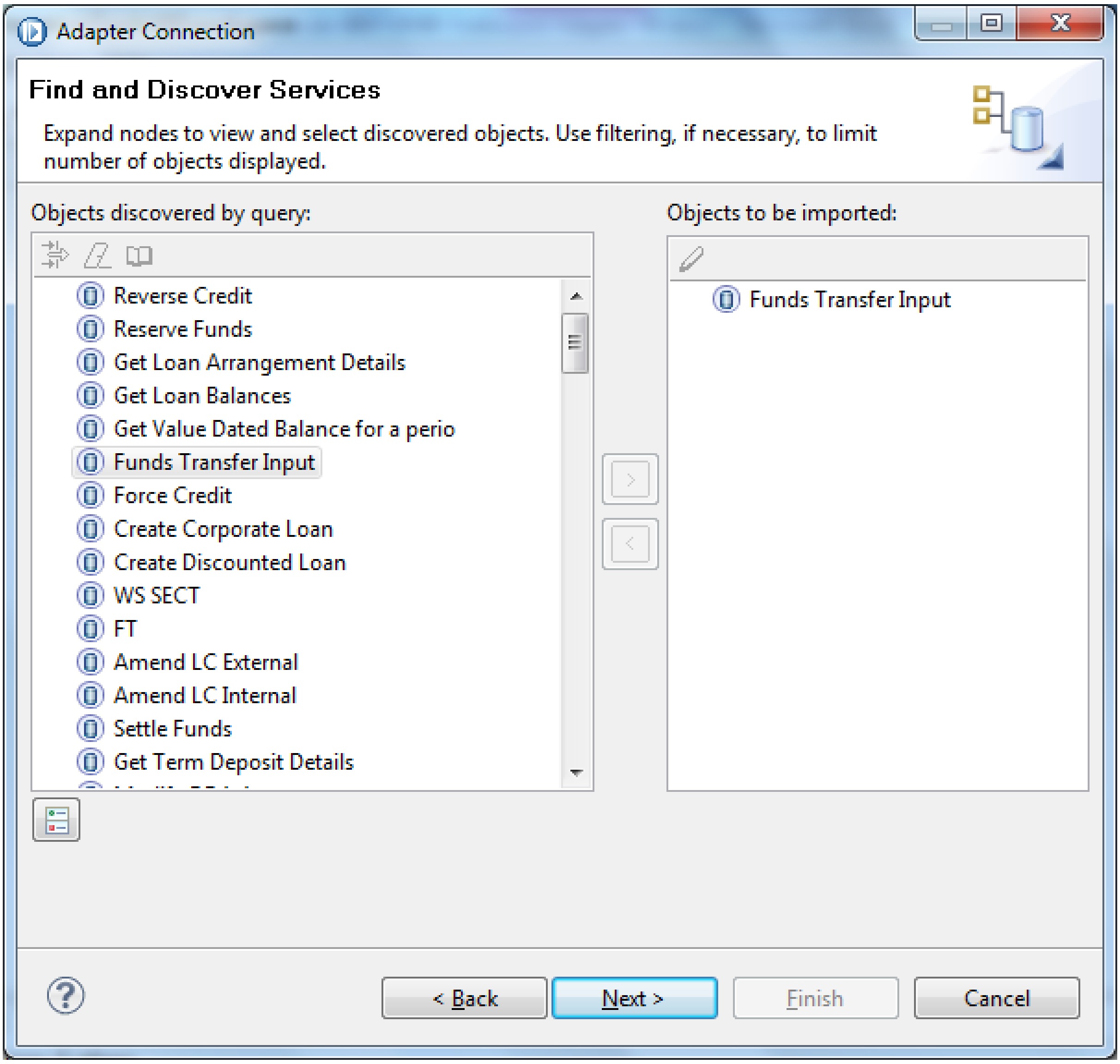
T24 User Name: * INPUTT

T24 Password: ******

□ Specify the level of the logging desired

Note: Refer the Deploying Component Service User Guide to know how to deploy a component in a J2EE application server.

7. On proceeding to the next screen, the list of services available in T24 gets listed. Select the required services and click next.



Adapter Connection

Find and Discover Services

Expand nodes to view and select discovered objects. Use filtering, if necessary, to limit number of objects displayed.

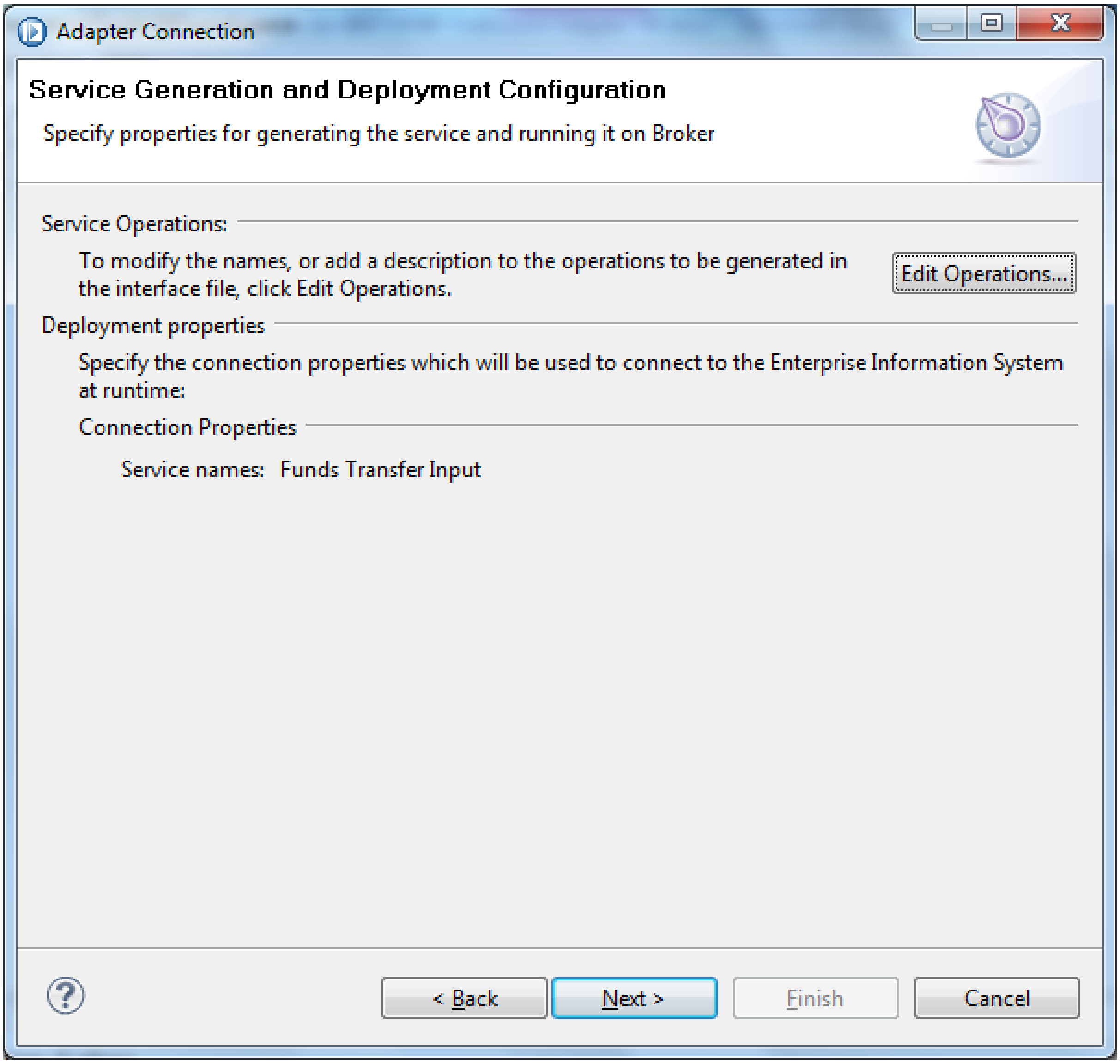
Objects discovered by query:

- Reverse Credit
- Reserve Funds
- Get Loan Arrangement Details
- Get Loan Balances
- Get Value Dated Balance for a perio
- Funds Transfer Input
- Force Credit
- Create Corporate Loan
- Create Discounted Loan
- WS SECT
- FT
- Amend LC External
- Amend LC Internal
- Settle Funds
- Get Term Deposit Details

Objects to be imported:

- Funds Transfer Input

8. The next screen lists the services selected.



Service Generation and Deployment Configuration

Specify properties for generating the service and running it on Broker

Service Operations:

To modify the names, or add a description to the operations to be generated in the interface file, click Edit Operations.

Deployment properties

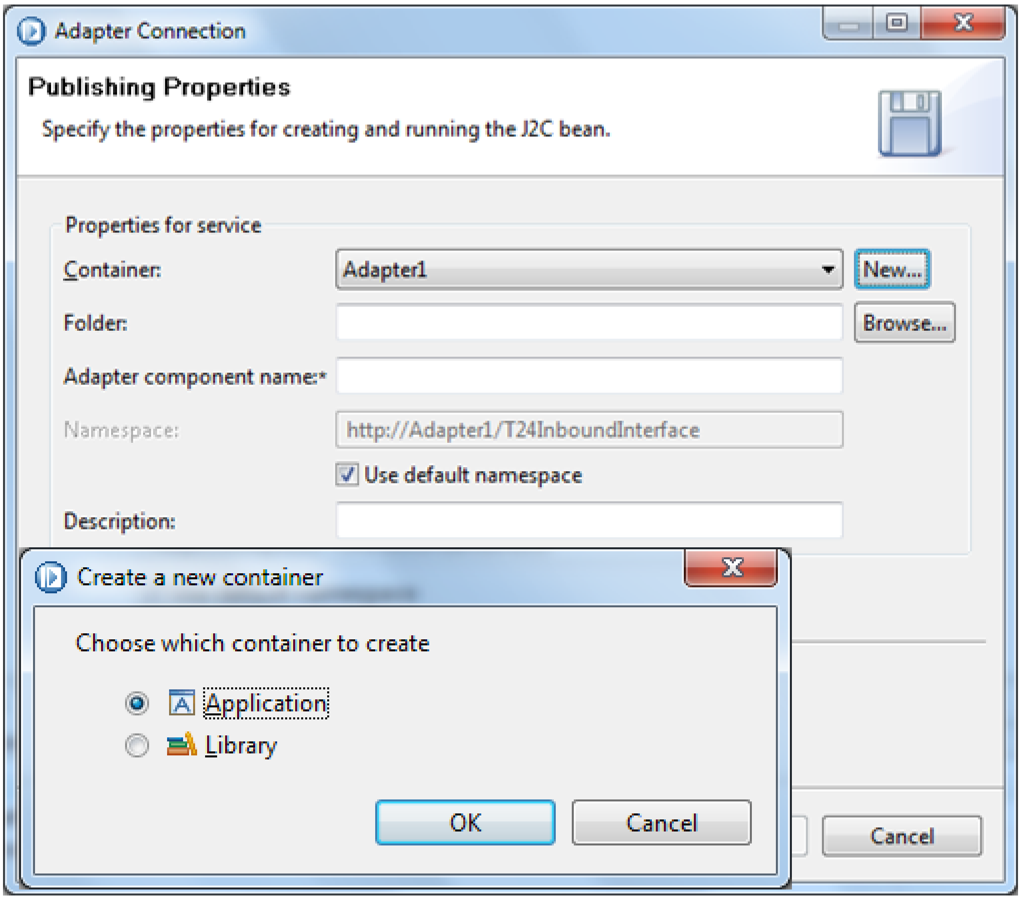
Specify the connection properties which will be used to connect to the Enterprise Information System at runtime:

Connection Properties

Service names: Funds Transfer Input

Click next to continue.

Now the wizard moves to the next step to configure the publishing properties. Here the user has to specify the container for the adapter connection properties. The container could be an Application or a library based on the requirement whether to reuse the adapter connection or not. To create a new container click on New and select the type of container.



Adapter Connection

Publishing Properties

Specify the properties for creating and running the J2C bean.

Properties for service

Container: Adapter1 [New...]

Folder: [Browse...]

Adapter component name:*

Namespace: http://Adapter1/T24InboundInterface

☑ Use default namespace

Description:

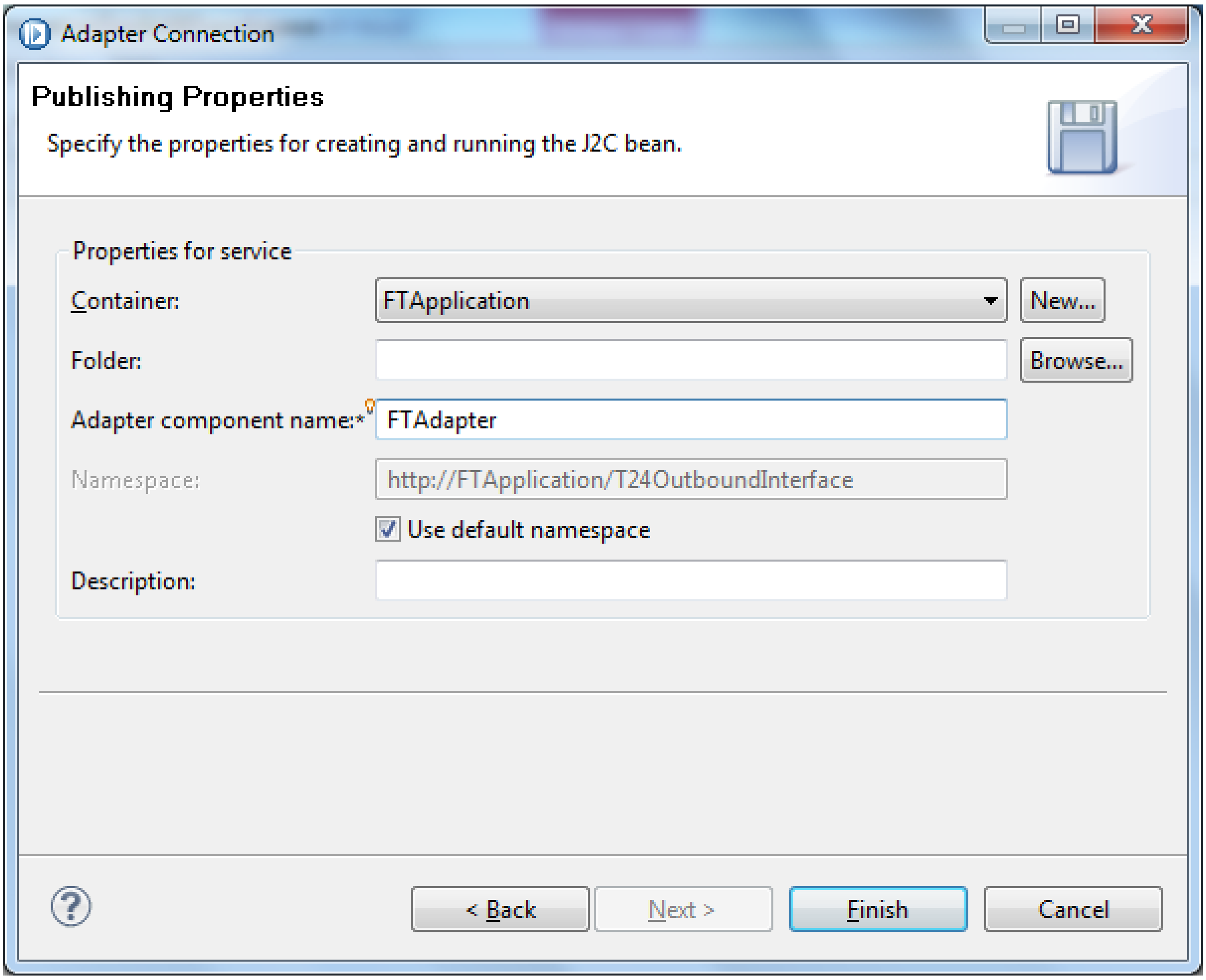
Create a new container

Choose which container to create

● Application

○ Library

1. Enter the Project Name in the dialog box that opens and click finish.
2. Then in the publishing properties dialog, specify a name for the Adapter and click finish.


Adapter Connection

Publishing Properties

Specify the properties for creating and running the J2C bean.

Properties for service

Container: FTApplication

Folder:

Adapter component name:* FTAdapter

Namespace: http://FTApplication/T24OutboundInterface

☑ Use default namespace

Description:

Note: Ensure that the folder name is left blank.

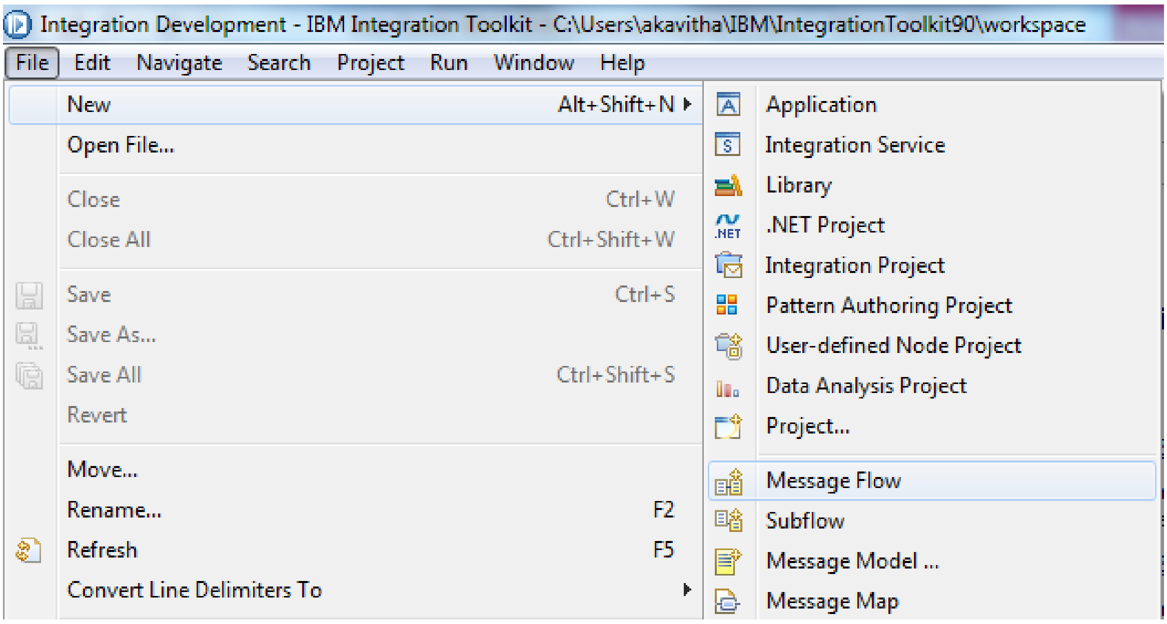
Creating a Message Flow

When a WebSphere Business Integration adapter uses a IIB as its integration broker, it uses WebSphere MQ message flows to process and route business object messages representing data or requests being sent by business applications to one another. A single message flow, defined for each queue, processes all messages placed on that queue. Using the MQ Integrator Broker Control Center (for WebSphere MQ Integrator Broker or WebSphere MQ Integrator) or the Integration Toolkit (for WebSphere Business Integration Message Broker), the user can build message flows from message-processing primitives to allow processing decisions to be made on either the message header or the message content. That is, the message flow can specify different processing steps for each type of message it is expected to handle.

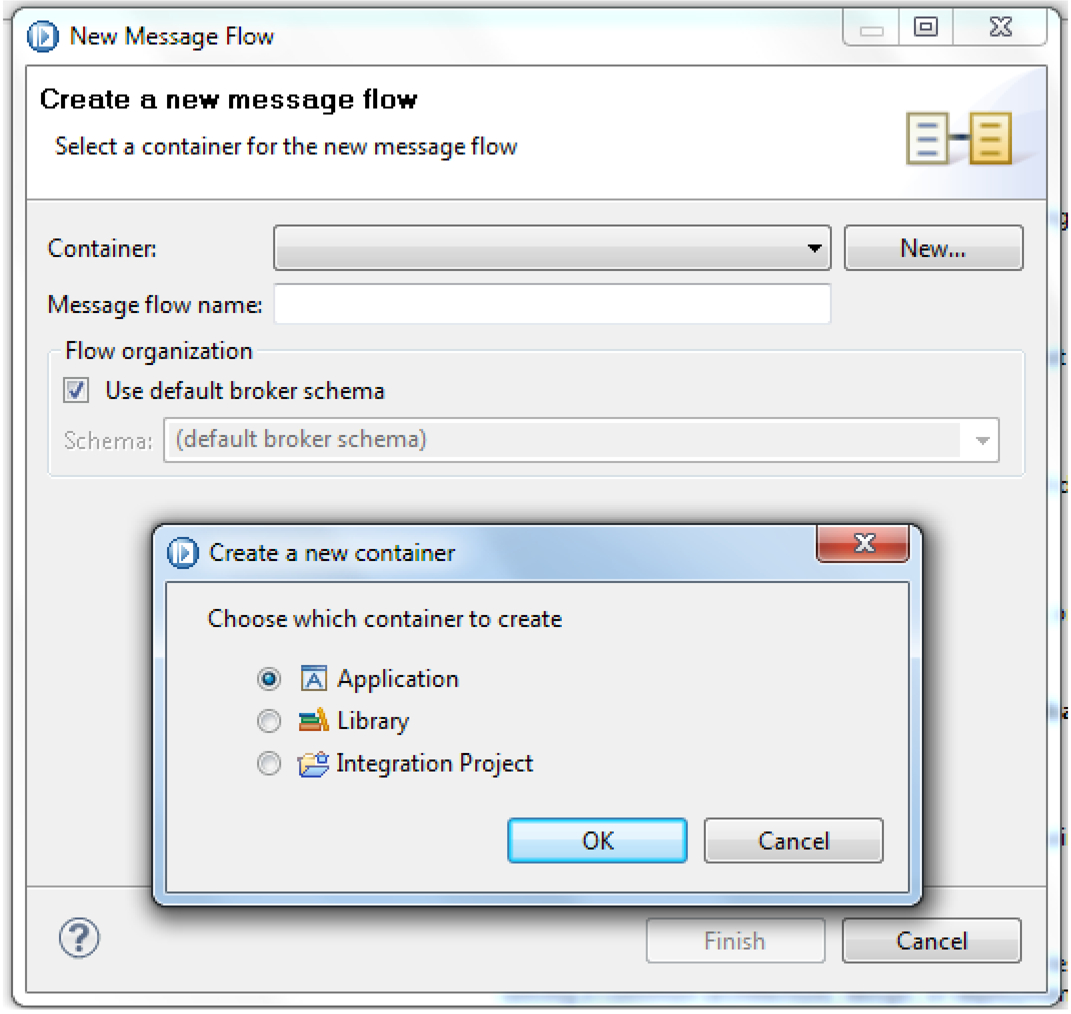
Here the process of creating and deploying Message flows in IIB is explained.

To define a message flow

1. Change the perspective to “Integration Development” perspective and create a new Message Flow.



2. This opens the New Message Flow Dialog Window. To add the flow to a new container, click new. The message flow has to be added to an application or to a library. Here application is used as a container for the flow.



New Message Flow

Create a new message flow

Select a container for the new message flow

Container: [▾] New...

Message flow name:

Flow organization

☑ Use default broker schema

Schema: (default broker schema)

Create a new container

Choose which container to create

● Application

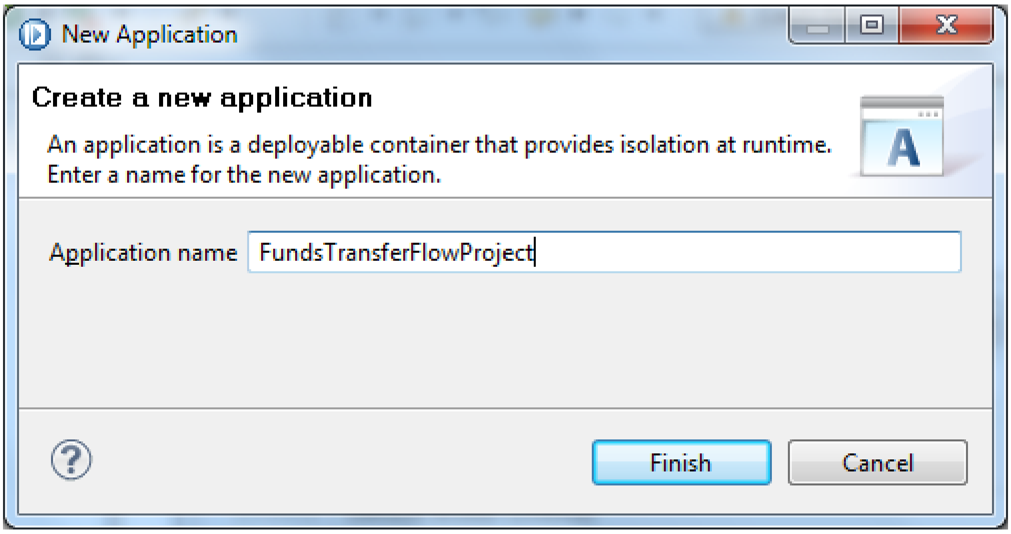
○ Library

○ Integration Project

OK Cancel

Finish Cancel

3. Specify a name for the new application and click finish.



New Application

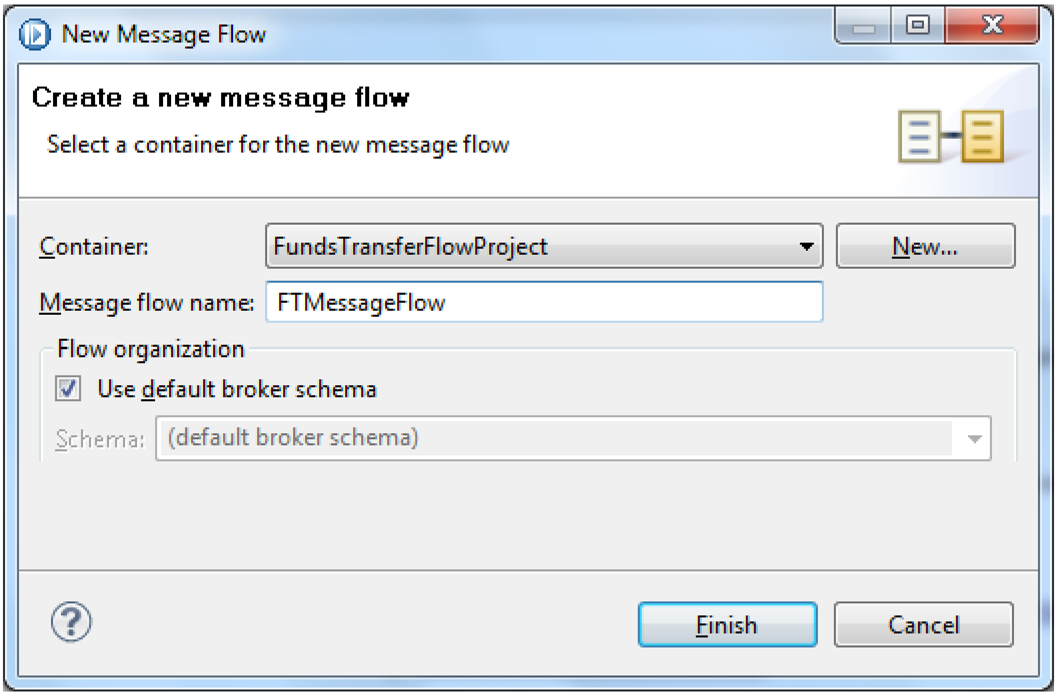
Create a new application

An application is a deployable container that provides isolation at runtime.
Enter a name for the new application.

Application name: FundsTransferFlowProject

Finish Cancel

4. Specify a name for the flow.



New Message Flow

Create a new message flow

Select a container for the new message flow

Container: FundsTransferFlowProject

Message flow name: FTMessageFlow

Flow organization

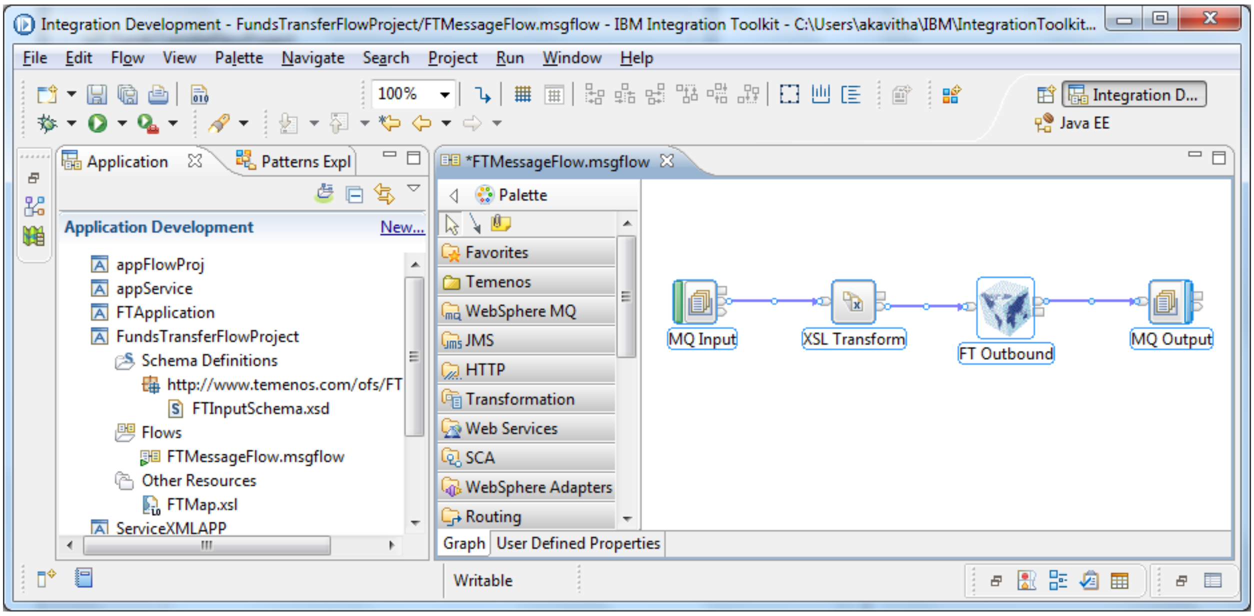
☑ Use default broker schema

Schema: (default broker schema)

5. Click finish.

Once the flow is created, the flow designer is available to the user to design the flow of the message and to apply any transformations.

6. To design the flow drag and drop the T24 Outbound Adapter Node from the service palatte. As the message source and destination is MQ, add the MQ Input and Output Node to the designer. This allows the message from T24 is directly delivered to a MQ Queue. The XSL Transform node is required in case the input is a XML format, but the T24 Outbound Adapter requires a different XMl Structure.



Integration Development - FundsTransferFlowProject/FTMessageFlow.msgflow - IBM Integration Toolkit - C:\Users\akavitha\IBM\IntegrationToolkit...

Application Development

- appFlowProj
- appService
- FTApplication
- FundsTransferFlowProject
  - Schema Definitions
    - http://www.temenos.com/ofs/FT
      - FTInputSchema.xsd
  - Flows
    - FTMessageFlow.msgflow
  - Other Resources
    - FTMap.xsl
- ServiceXMLAPP

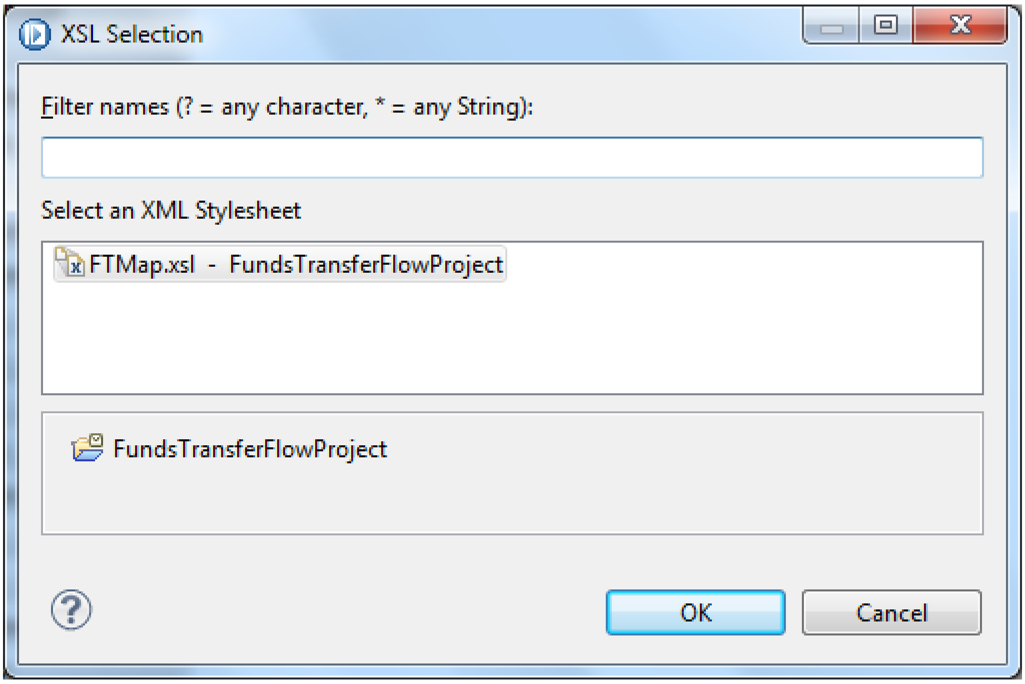
*FTMessageFlow.msgflow

XSL Transform Node Configuration:

XSL Transform node is used to transform an Input XML to a different XML based on the schema structure. As this example shows the configuration of ServiceXML, the XSL Transform node is used to transform the input XML to ServiceXML Type of request.

To use XSL Transform the user has to define a stylesheet that maps the Source and Destination XML Schema. Then the Style Sheet and the Source Schema structure have to be imported to the flow project.

To configure XSL Transform Node, double click the XSL Transform Node. This will open a dialog box as shown below:


XSL Selection
Filter names (? = any character, * = any String):
Select an XML Stylesheet
FTMap.xsl - FundsTransferFlowProject
FundsTransferFlowProject
OK
Cancel

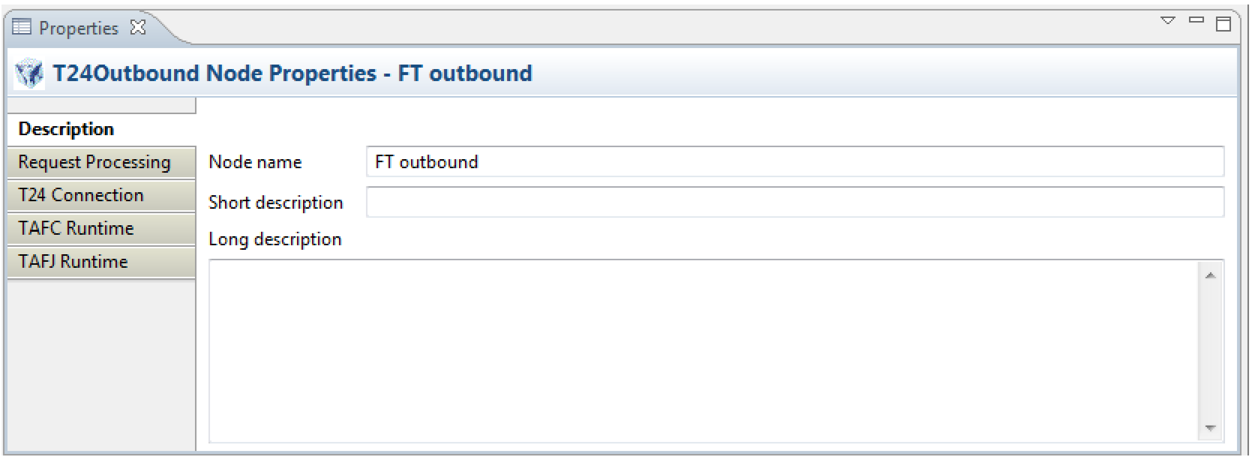
Select the imported style sheet and click Ok.

T24 Outbound Node Configuration:

The T24 Outbound node has a set of properties to be configured. The property page can be accessed by double clicking the Node.

Description:

The description tab of the property page is used to specify the name of the node and an optional description.



Properties

T24Outbound Node Properties - FT outbound

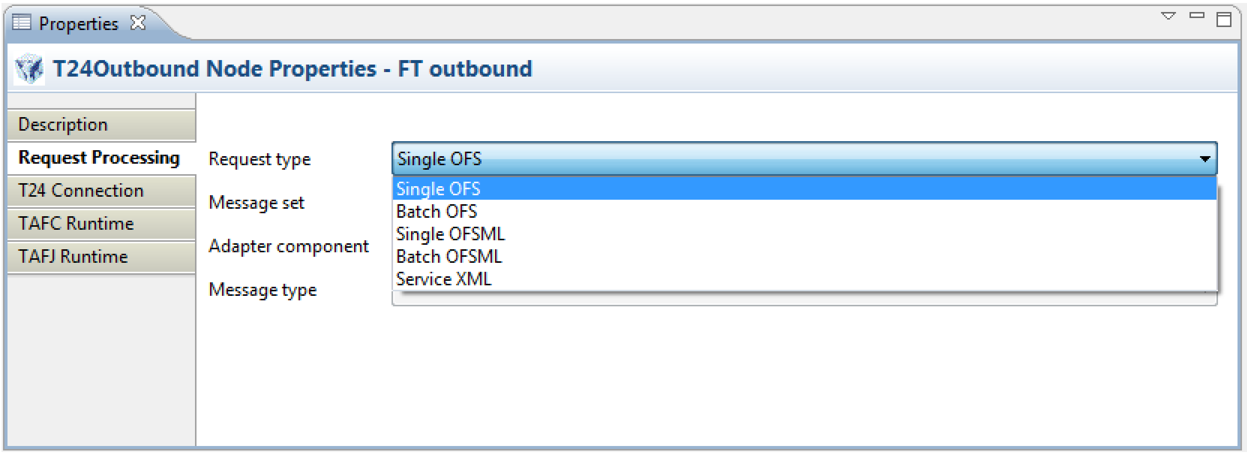
Node name: FT outbound

Short description:

Long description:

Request Processing

The Request Processing tab is used to specify the outbound operation that this flow is going to perform.



Properties

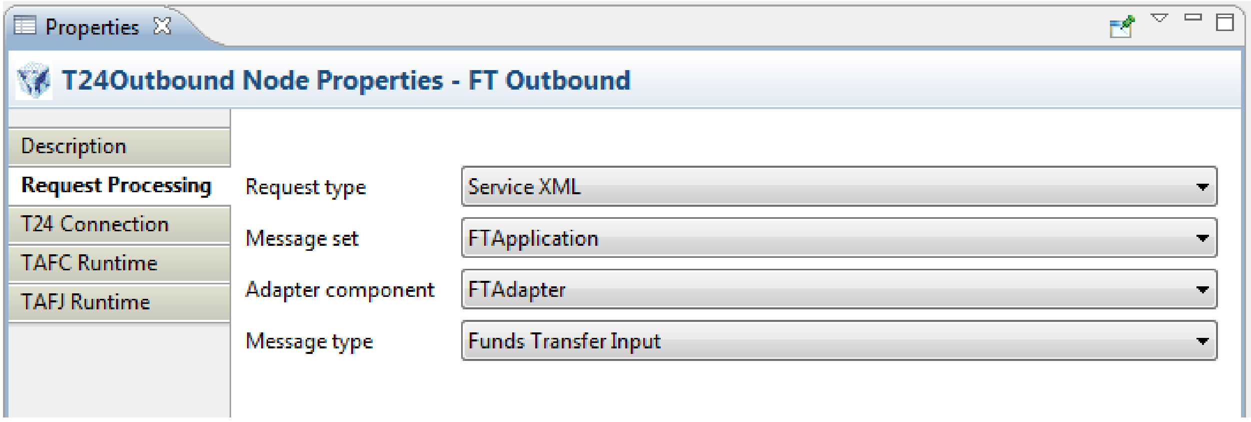
T24Outbound Node Properties - FT outbound

Request type: Single OFS

Message set

Adapter component

Message type



Properties

T24Outbound Node Properties - FT Outbound

Request type: Service XML

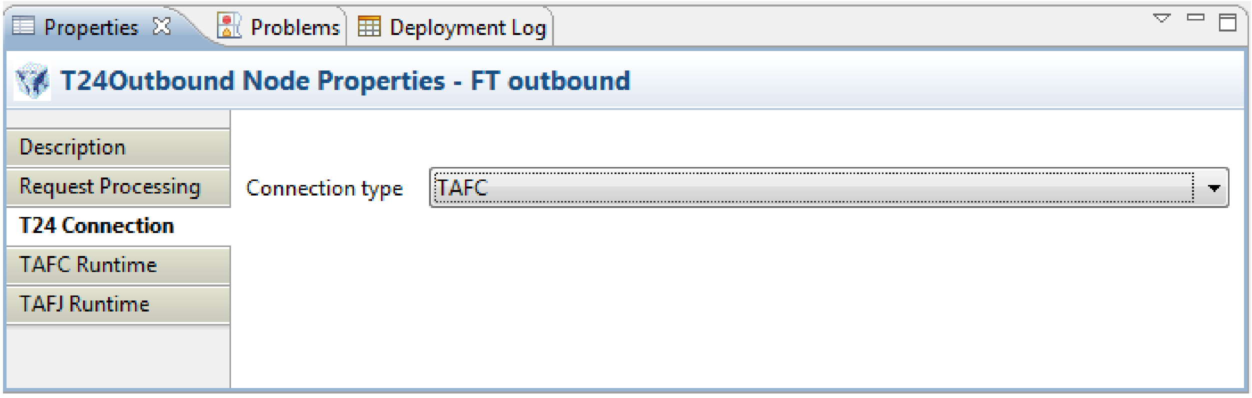
Message set: FTApplication

Adapter component: FTAdapter

Message type: Funds Transfer Input

T24 Connection:

The T24 Connection tab allows the user to specify the connection type.The connection type can be set as TAFC or TAFJ based on the runtime specified.

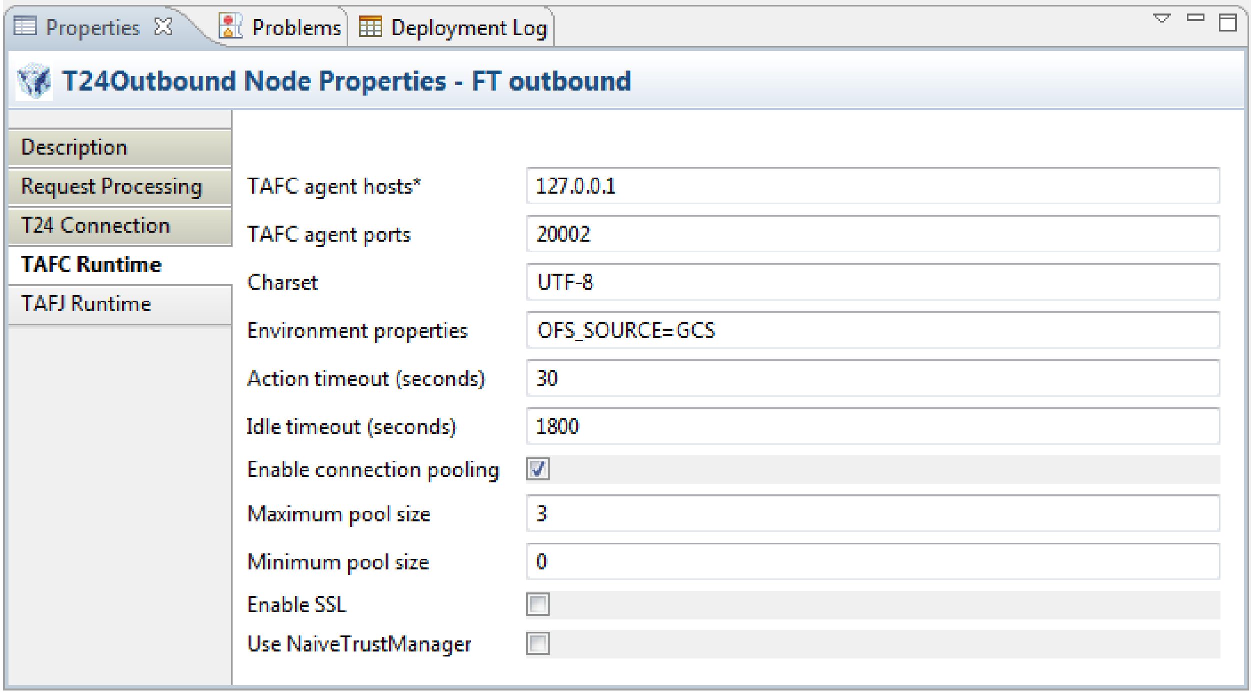


T24Outbound Node Properties - FT outbound

Connection type: TAFC

TAFC Runtime:

The TAFC Runtime tab allows the user to specify the settings that is used at runtime when TAFC is mentioned as the connection type in the T24 Connection Tab.



T24Outbound Node Properties - FT outbound

TAFC agent hosts*	127.0.0.1
TAFC agent ports	20002
Charset	UTF-8
Environment properties	OFS_SOURCE=GCS
Action timeout (seconds)	30
Idle timeout (seconds)	1800
Enable connection pooling	☑
Maximum pool size	3
Minimum pool size	0
Enable SSL	☐
Use NaiveTrustManager	☐

TAFC Agent Hosts: Comma separated list of server addresses hosting TAFC Agent e.g. 10.44.1.100, 10.44.1.101

TAFC Agent Ports: This property specifies TCP Port number where TAFC Agent is listening. e.g. 20002, 20003

Charset: This property specifies the character set of the remote EIS on which TAFC Agent is running. T24RA will use this value when encoding and decoding character data between the server and the client. This setting should only be used when connecting to non I18N T24 configurations. It is set to UTF-8 by default.

Environment Properties: This property specifies OFS Source record ID to communicate with T24.

ActionTimeout: This property configures the length of time the TAFC Agent will wait for a request to return a result before forcibly stopping the subroutine call and exiting. This setting is very important to avoid T24 deadlocks as all T24 locks will be released when the action is stopped, thus allowing processing in other connections to continue.

IdleTimeout: This is the maximum time out period, till when the connection can be idle in the pool before being removed to free resources.

Enable Connection Pooling: This property specifies whether the connection to T24 is pooled or not. Values are true or false.

Maximum Pool Size: This specifies the maximum number of connections that can be available in pool.

Minimum Pool Size: This specifies the minimum number of connections to be used.

Enable SSL: This is set to true to enable Secure connection This option should only be enabled when the remote TAFC Agent instance is also configured to use SSL encryption.

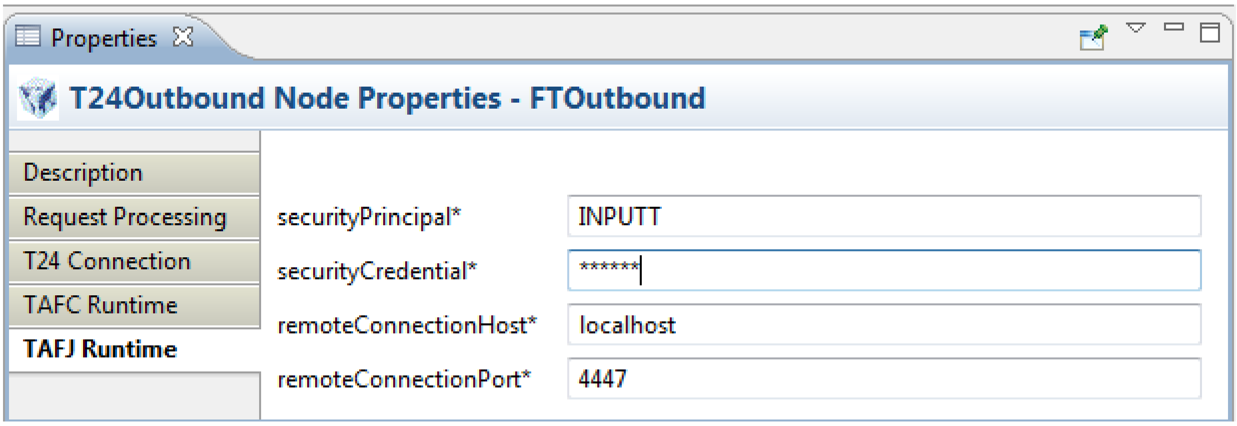
Use Naïve Trust Manager: This property is set to true when enabling secure connections via SSL

TAFJ Runtime:

The TAFJ Runtime tab allows the user to specify the settings that is used at runtime when TAFJ is mentioned as the connection type in the T24 Connection Tab.

1. **SecurityPrincipal and SecutityCredential**: These properties is used to authenticate the runtime connection with JBoss and for JBoss to pass on the principal to T24.
2. **Remote Connection Host**: The name or IP Address of the machine where jBoss is running.
3. **Remote Connection Port**: This is the port number used for remote ejb lookup.

   Remote connection Host and Port resolve the ejb lookup for t24-IntegrationFrameworkService-ejb.jar deployed in the jboss
   Note: Please refer the Appendix ‘Setting up T24 and jboss for TAFJ’ if the runtime is TAFJ.

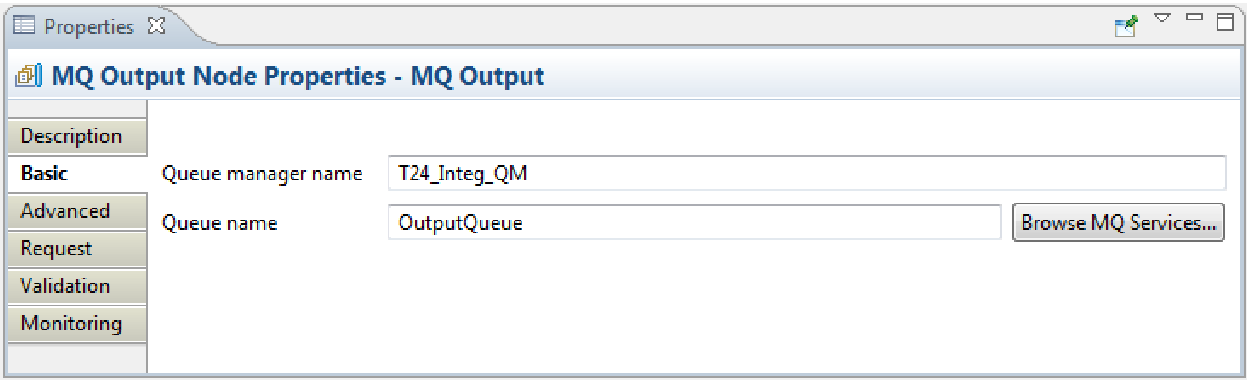


T24Outbound Node Properties - FTOutbound

securityPrincipal* INPUTT
securityCredential* ******
remoteConnectionHost* localhost
remoteConnectionPort* 4447

MQ Node Configuration:

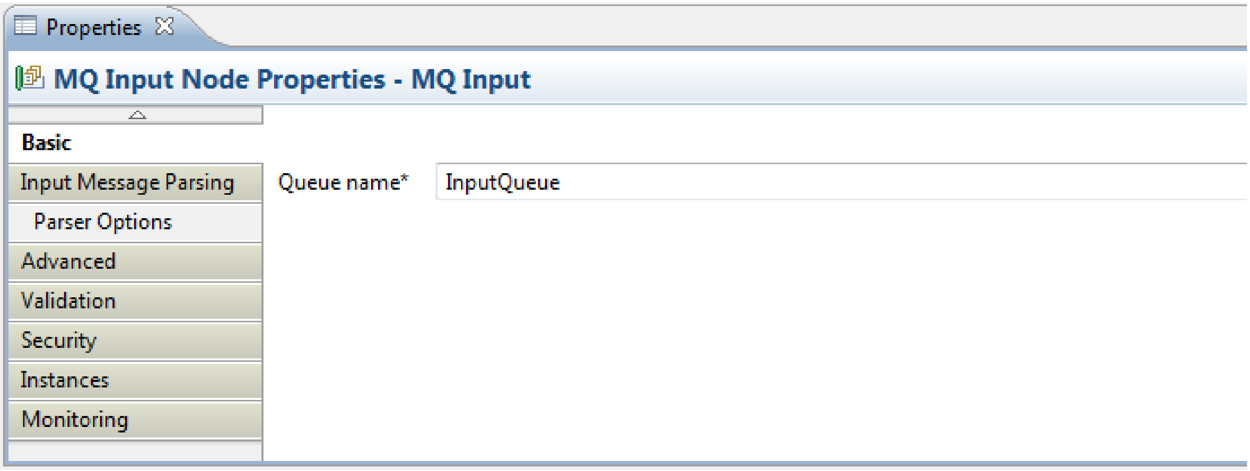
The MQ Output Node has to be configured to specify the MQ Queue Manager name and the Queue Name to which the event will be delivered.



Queue manager name T24_Integ_QM

Queue name OutputQueue

The MQ Input node has to be configured to specify the MQ from where the request has to be taken for processing



Queue name* InputQueue

In the MQ Input Node the Input Message Parsing option has to be configured based on the Input Type as:

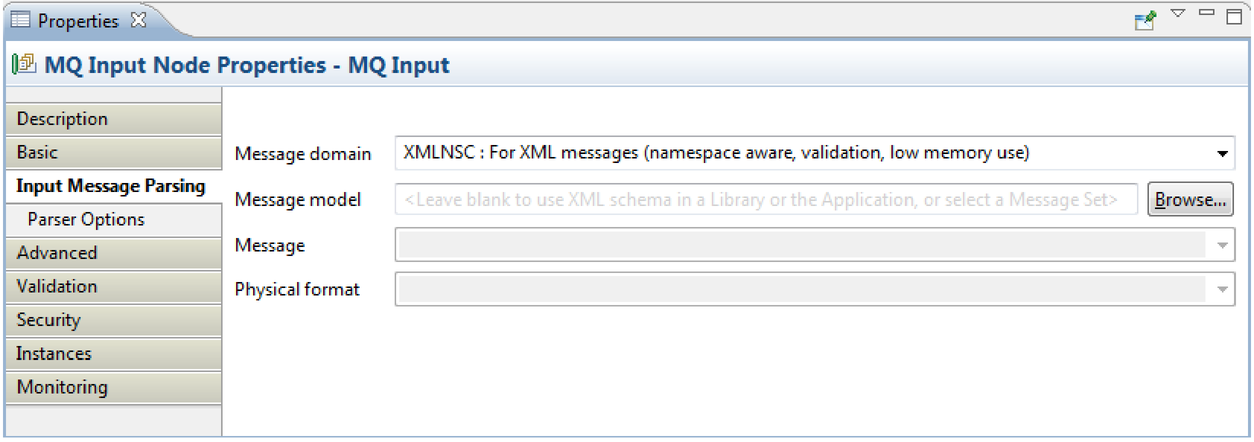
OFS: Message Domain should be “BLOB : For messages with an unspecified format”

OFSML: Message Domain should be “XMLNS : For XML messages (namespace aware)”

Batch OFS: Message Domain should be “XMLNS : For XML messages (namespace aware)”

Batch OFSML: Message Domain should be “XMLNS : For XML messages (namespace aware)”

Service XML: Message Domain should be “XMLNSC : For XML messages (namespace aware, validation, low memory use)”.



Message domain: XMLNSC : For XML messages (namespace aware, validation, low memory use)

Message model: <Leave blank to use XML schema in a Library or the Application, or select a Message Set>

Message:

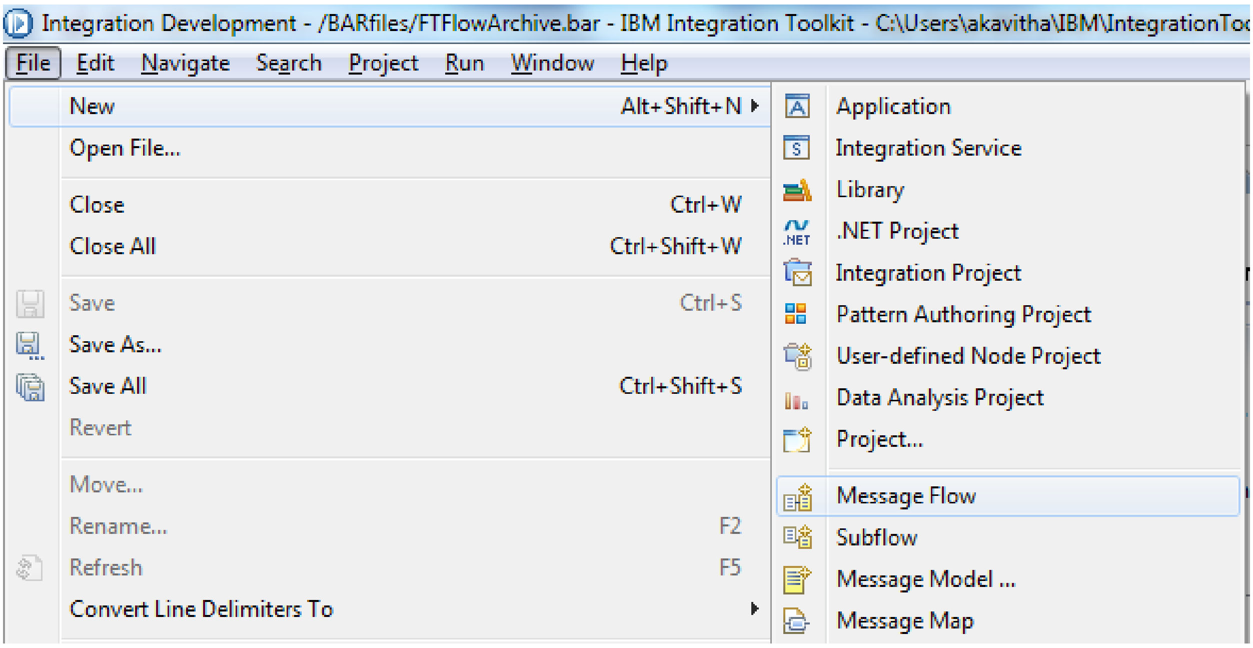
Physical format:

Deploying Message Flow in a Message Broker

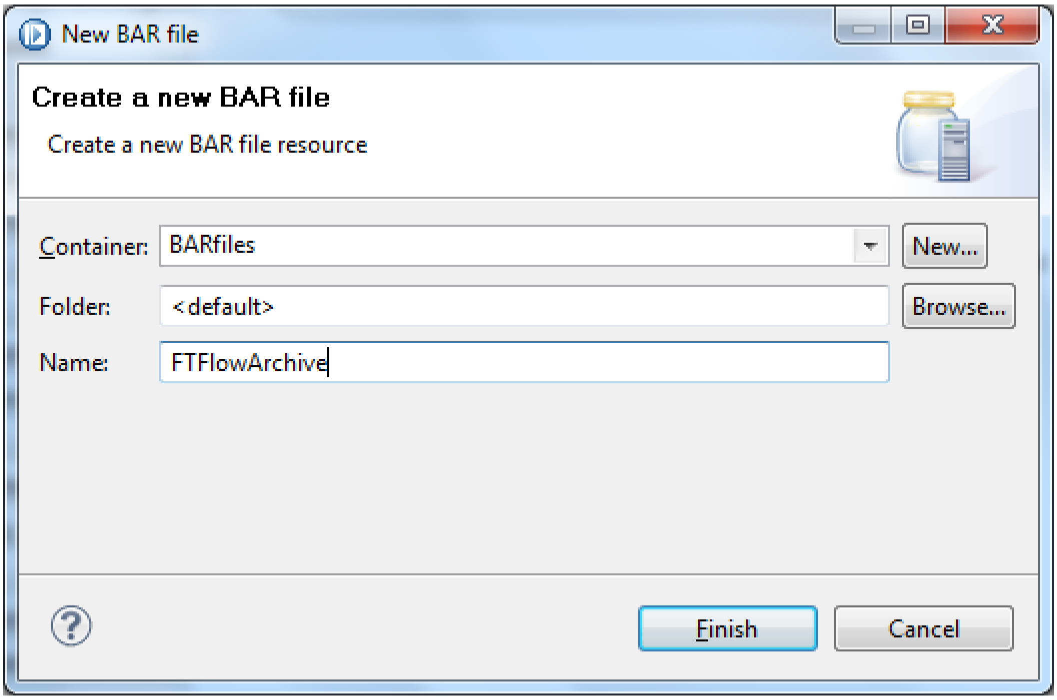
When a WebSphere Business Integration adapter uses a message broker as its integration broker, it uses WebSphere MQ message flows to process and route business object messages representing data or requests being sent by business applications to one another. A single message flow, defined for each queue, processes all messages placed on that queue.

Here the process of deploying Message flows in a Message Broker is explained.

1. Once the flow is configured, the user has to create the Message Broker Archive. To do so, use the menu option File → New → BAR File.



2. The Message Broker Archive project is usually added to a container. Here the container BARfiles is used.



New BAR file

Create a new BAR file

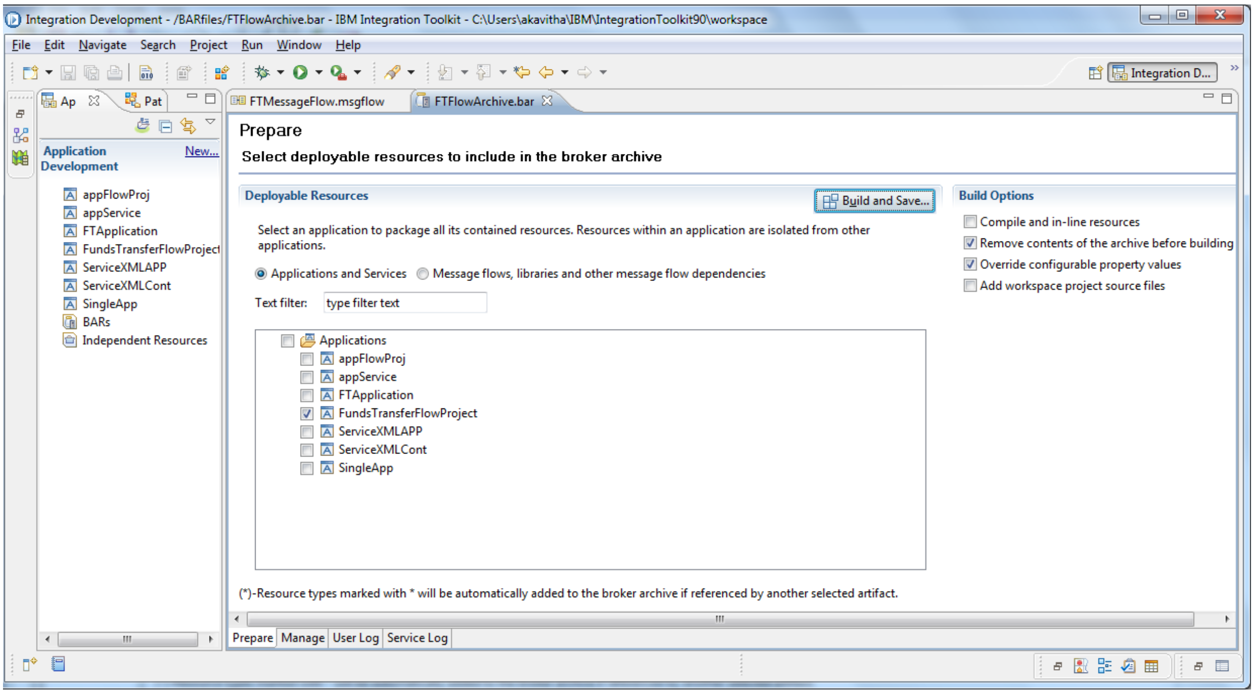
Create a new BAR file resource

Container: BARfiles

Folder: <default>

Name: FTFlowArchive

3. The user is now taken to the prepare screen of the Message Broker Archive inorder to select the Message Flow that are to be added to archive.



Integration Development - /BARfiles/FTFlowArchive.bar - IBM Integration Toolkit - C:\Users\akavitha\IBM\IntegrationToolkit90\workspace

Prepare

Select deployable resources to include in the broker archive

Deployable Resources

Select an application to package all its contained resources. Resources within an application are isolated from other applications.

● Applications and Services ○ Message flows, libraries and other message flow dependencies

Text filter: type filter text

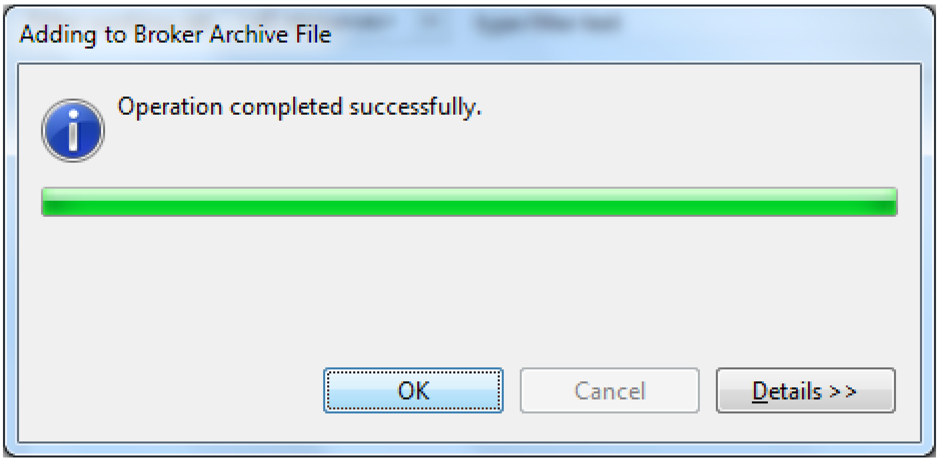
- □ Applications
  - □ appFlowProj
  - □ appService
  - □ FTApplication
  - ☑ FundsTransferFlowProject
  - □ ServiceXMLAPP
  - □ ServiceXMLCont
  - □ SingleApp

(*)-Resource types marked with * will be automatically added to the broker archive if referenced by another selected artifact.

Build Options

- □ Compile and in-line resources
- ☑ Remove contents of the archive before building
- ☑ Override configurable property values
- □ Add workspace project source files

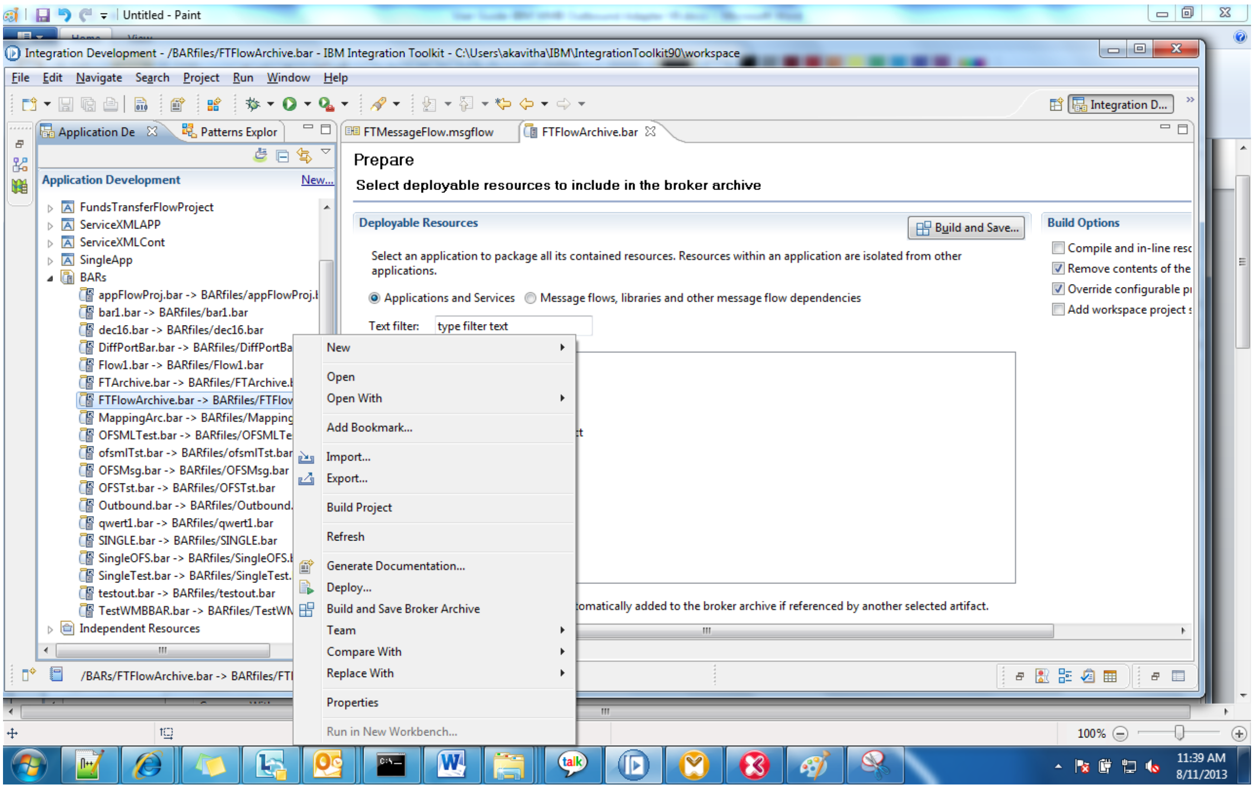
Once the required contents are selected, the broker archive has to be built using the “Build and Save” option.



Adding to Broker Archive File

Operation completed successfully.

4. The built Broker Archive now is ready to be deployed into a Broker. The deployment process is very simple. Right click the broker archive file and select deploy.



Integration Development - /BARfiles/FTFlowArchive.bar - IBM Integration Toolkit - C:\Users\akavitha\IBM\IntegrationToolkit90\workspace

Prepare

Select deployable resources to include in the broker archive

Deployable Resources

Select an application to package all its contained resources. Resources within an application are isolated from other applications.

● Applications and Services ○ Message flows, libraries and other message flow dependencies

Text filter: type filter text

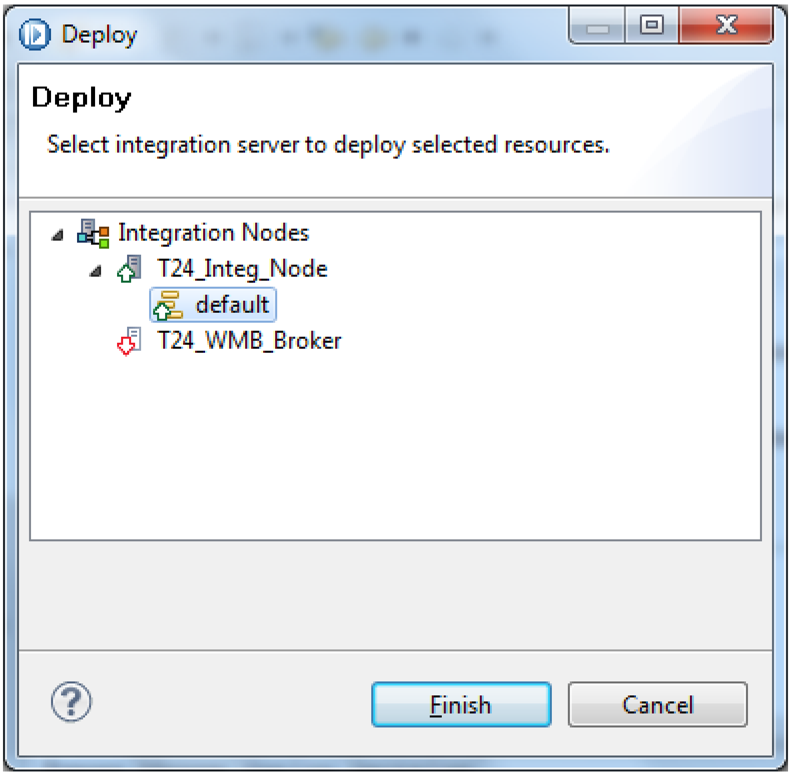
…omatically added to the broker archive if referenced by another selected artifact.

Build Options

□ Compile and in-line resc
☑ Remove contents of the
☑ Override configurable pi
□ Add workspace project s

Untitled - Paint
File
Edit
Navigate
Search
Project
Run
Window
Help
Integration D...
Application De
Patterns Explor
FTMessageFlow.msgflow
FTFlowArchive.bar
Application Development
New...
FundsTransferFlowProject
ServiceXMLAPP
ServiceXMLCont
SingleApp
BARs
appFlowProj.bar -> BARfiles/appFlowProj.b
bar1.bar -> BARfiles/bar1.bar
dec16.bar -> BARfiles/dec16.bar
DiffPortBar.bar -> BARfiles/DiffPortBa
Flow1.bar -> BARfiles/Flow1.bar
FTArchive.bar -> BARfiles/FTArchive.b
FTFlowArchive.bar -> BARfiles/FTFlov
MappingArc.bar -> BARfiles/Mapping
OFSMLTest.bar -> BARfiles/OFSMLTe
ofsmlTst.bar -> BARfiles/ofsmlTst.bar
OFSMsg.bar -> BARfiles/OFSMsg.bar
OFSTst.bar -> BARfiles/OFSTst.bar
Outbound.bar -> BARfiles/Outbound.
qwert1.bar -> BARfiles/qwert1.bar
SINGLE.bar -> BARfiles/SINGLE.bar
SingleOFS.bar -> BARfiles/SingleOFS.b
SingleTest.bar -> BARfiles/SingleTest.
testout.bar -> BARfiles/testout.bar
TestWMBBAR.bar -> BARfiles/TestWN
Independent Resources
/BARs/FTFlowArchive.bar -> BARfiles/FTI
Build and Save...
New
Open
Open With
Add Bookmark...
Import...
Export...
Build Project
Refresh
Generate Documentation...
Deploy...
Build and Save Broker Archive
Team
Compare With
Replace With
Properties
Run in New Workbench...
100%
11:39 AM
8/11/2013

5. Then select the Broker under which the broker archive has to be deployed.

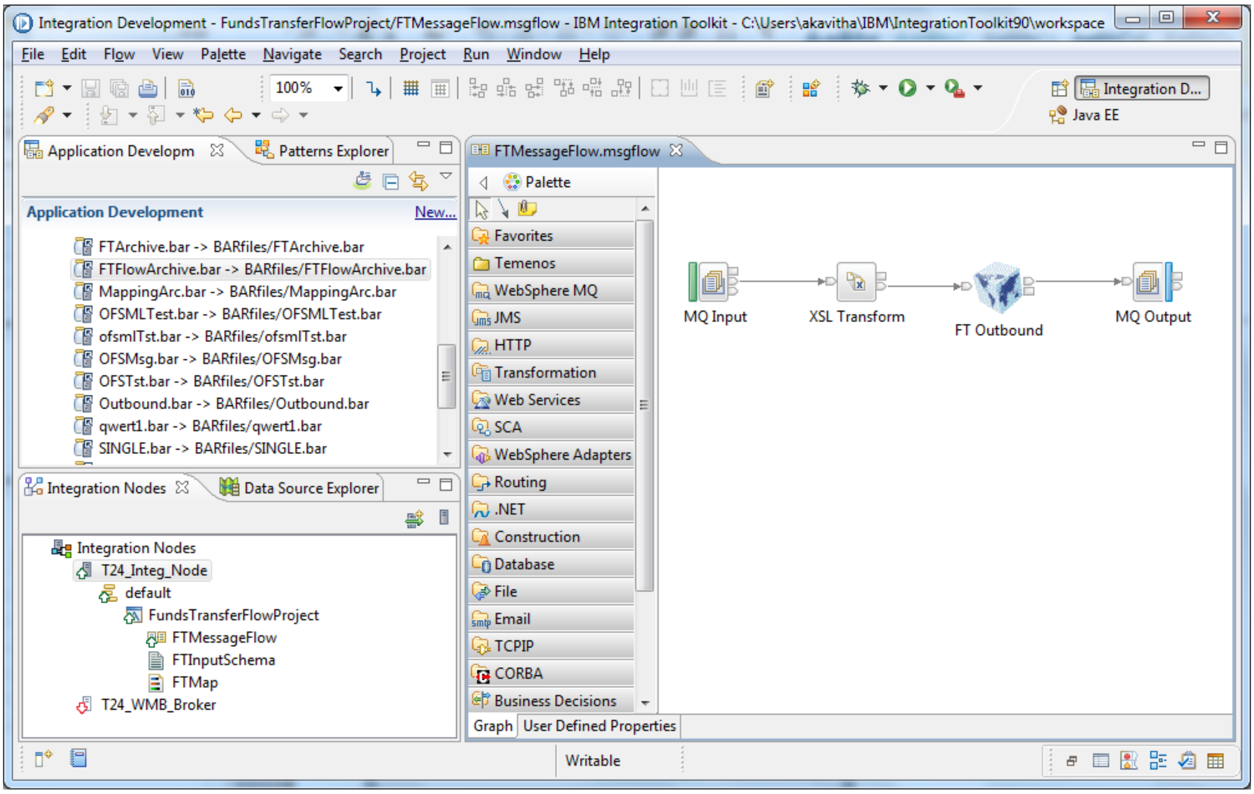


Deploy

Select integration server to deploy selected resources.

- Integration Nodes
  - T24_Integ_Node
    - default
  - T24_WMB_Broker

6. Click finish. This starts the deploy process. Once the deployment is completed successful the progress information dialog box gets closed and the deployed can be seen under the Integration Node.



Integration Development - FundsTransferFlowProject/FTMessageFlow.msgflow - IBM Integration Toolkit - C:\Users\akavitha\IBM\IntegrationToolkit90\workspace

Application Development

- FTArchive.bar -> BARfiles/FTArchive.bar
- FTFlowArchive.bar -> BARfiles/FTFlowArchive.bar
- MappingArc.bar -> BARfiles/MappingArc.bar
- OFSMLTest.bar -> BARfiles/OFSMLTest.bar
- ofsmlTst.bar -> BARfiles/ofsmlTst.bar
- OFSMsg.bar -> BARfiles/OFSMsg.bar
- OFSTst.bar -> BARfiles/OFSTst.bar
- Outbound.bar -> BARfiles/Outbound.bar
- qwert1.bar -> BARfiles/qwert1.bar
- SINGLE.bar -> BARfiles/SINGLE.bar

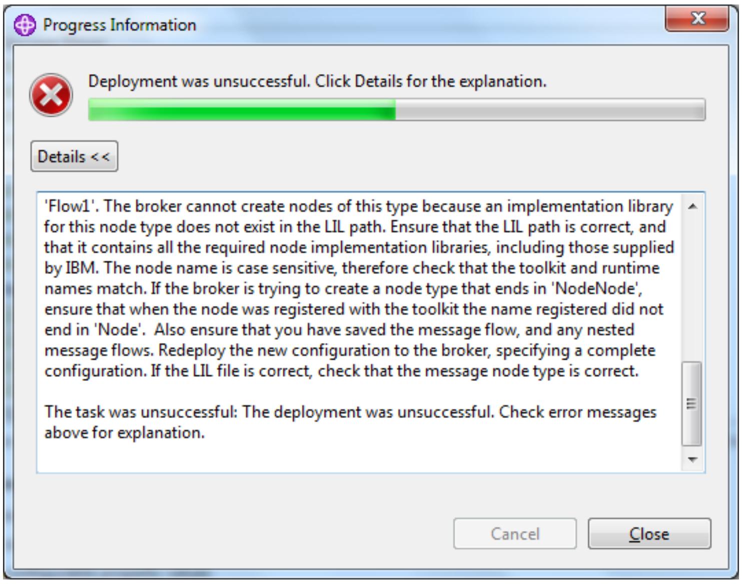
Integration Nodes

- Integration Nodes
  - T24_Integ_Node
    - default
      - FundsTransferFlowProject
        - FTMessageFlow
        - FTInputSchema
        - FTMap
  - T24_WMB_Broker

Once the Message broker archive is deployed, the user can send request to T24.

Troubleshooting

Problem: Error shown below while deploying:



Deployment was unsuccessful. Click Details for the explanation.

'Flow1'. The broker cannot create nodes of this type because an implementation library for this node type does not exist in the LIL path. Ensure that the LIL path is correct, and that it contains all the required node implementation libraries, including those supplied by IBM. The node name is case sensitive, therefore check that the toolkit and runtime names match. If the broker is trying to create a node type that ends in 'NodeNode', ensure that when the node was registered with the toolkit the name registered did not end in 'Node'. Also ensure that you have saved the message flow, and any nested message flows. Redeploy the new configuration to the broker, specifying a complete configuration. If the LIL file is correct, check that the message node type is correct.

The task was unsuccessful: The deployment was unsuccessful. Check error messages above for explanation.

Solution: Clean and rebuild the project

Glossary

Appendix

Starting TAFC Agent:

The TAFC Agent has to be started for establishing the connectivity to TEMENOS T24 using TAFC as runtime. SSL certificates can also be used while starting the TAFC Agent. Thus the connectivity between Integration Studio and jremote is secured using certificates. TAFC Agent may be started from the command line/shell via the following executable:

```
tafc_agent -p [PORT NUMBER]
```

Example:

```
tafc_agent -p 33333
```

(or)

```
tafc_agent -p [PORT NUMBER] -c [CERTIFICATE PATH] -k [KEY PATH]
```

Example:

```
tafc_agent -p 33333 -c c:\openssl\bin\keys\cert.cer -k c:\openssl\bin\keys\key.pem
```

References

Creating a queue manager in IBM MQ:
http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp?topic=%2Fcom.ibm.mq.amqzag.doc%2Ffa14220_.htm

Creating a queue in IBM MQ:

http://publib.boulder.ibm.com/infocenter/wmqv7/v7r0/index.jsp?topic=%2Fcom.ibm.mq.explorer.tutorials.doc%2Fbi00257_.htm

Creating a broker:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/index.jsp?topic=%2Fcom.ibm.etools.mft.doc%2Fbe10000_.htm