

Tivoli. software

IBM



IBM White Paper: IBM Maximo 7.1 Integration Framework Configuration Basics

White Paper

Barbara Vander Weele (bcvander@us.ibm.com)

July 2008

Copyright Notice

Copyright © 2008 IBM Corporation, including this documentation and all software. All rights reserved. May only be used pursuant to a Tivoli Systems Software License Agreement, an IBM Software License Agreement, or Addendum for Tivoli Products to IBM Customer or License Agreement. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without prior written permission of IBM Corporation. IBM Corporation grants you limited permission to make hardcopy or other reproductions of any machine-readable documentation for your own use, provided that each such reproduction shall carry the IBM Corporation copyright notice. No other rights under copyright are granted without prior written permission of IBM Corporation. The document is not intended for production and is furnished "as is" without warranty of any kind. All warranties on this document are hereby disclaimed, including the warranties of merchantability and fitness for a particular purpose.

Note to U.S. Government Users—Documentation related to restricted rights—Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

Trademarks

The following are trademarks of IBM Corporation or Tivoli Systems Inc.: IBM, Tivoli, AIX, Cross-Site, NetView, OS/2, Planet Tivoli, RS/6000, Tivoli Certified, Tivoli Enterprise, Tivoli Ready, TME. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Lotus is a registered trademark of Lotus Development Corporation.

PC Direct is a trademark of Ziff Communications Company in the United States, other countries, or both and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC. For further information, see <http://www.setco.org/aboutmark.html>.

Other company, product, and service names may be trademarks or service marks of others.

Notices

References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to valid intellectual property or other legally protectable right of Tivoli Systems or IBM, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user. Tivoli Systems or IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, New York 10504-1785, U.S.A.

Printed in Ireland.

Table of Contents

Introduction

About this Paper III
Audience III

White Paper

Integration Framework Configuration 1
Basic Configuration Steps 3
 System Properties Application 4
 Setting System Properties 5
 Queues 7
 Integrate to an External System 8
 Identifying External Systems 8
 External System Application 9
 Example: Create an External System 10
 The Continuous Queue 20
 Produce an Event and a Message 21
 Transaction Flow 24
 Logging 24
 The Logging Application 25
 Summary 27

Conclusion

Summary 29
Acknowledgements 29



Introduction

About this Paper

The Integration Framework provides Web services and service-oriented architecture (SOA) technologies to support application services and coordination between enterprise systems and external applications. Integration can be quickly configured and customized to meet specific business requirements with predefined components and object structures.

The Integration Framework is composed of highly flexible business components to ensure compatibility with Web-based infrastructures. The technology is a proven solution that has been deployed in large enterprises and small organizations. Using the J2EE services and underlying components, the Integration Framework uses the latest Web technologies to integrate with external applications.

This paper explores configuration of the Integration Framework, which provides a solution for the synchronization and integration of data between applications.

Audience

This paper is intended for technical professionals who need an introduction to the Integration Framework.



White Paper

1 Integration Framework Configuration

The Integration Framework is part of Tivoli Process Automation Platform (TPAP). The environment in which the applications run is a J2EE-compliant application server. The two application servers currently used are IBM WebSphere and BEA WebLogic.

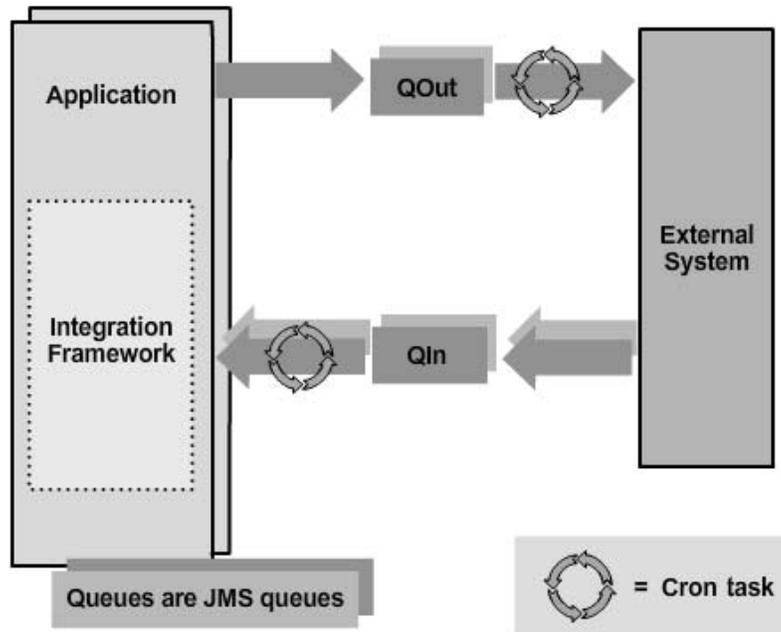
The Integration Framework is an integral part of TPAP. It allows the synchronization and integration of data between an external system and applications that use the TPAP common architecture and run under the application server. TPAP is synonymous with Base Services.

There are 57 integration objects (object structures) that are provided with the product and can be used by configuring the Integration Framework. The Integration Framework follows the service oriented architecture (SOA) directive and implements Web Services. It is completely customizable, and the behavior can be changed with rules instead of programming. There are also facilities to transform XML with XSL and Java customization.



In this paper, the word *application* is a generic term for any application that has been implemented using the Base Services (Maximo Asset Management, CCMDB, and so on). The Integration Framework is available to all applications that implement the Base Services.

The basic concept of integration starts with an application that has been implemented using the Base Services and an external system with which to integrate.



The queues are key Integration Framework components. They are JMS queues and defined using the application server's administration facilities. Each application server, whether BEA WebLogic or IBM WebSphere, requires that the queues be defined. The installation documentation details how to define the queues.

For IBM WebSphere, the installation of the application (Maximo or CCMDB) and Base Services has an option whereby the queues are created during installation.

To process the sequential outbound and inbound queues, a cron task is used. The Integration Framework comes with a predefined cron task **JMSQSEQCONSUMER**, which needs to be activated.

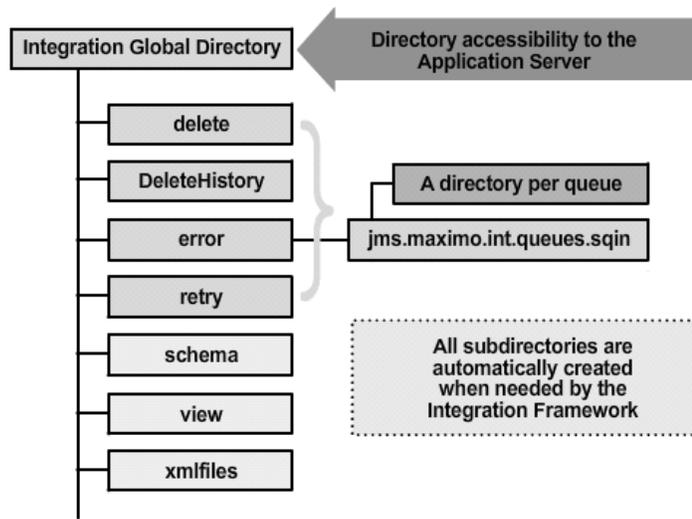
To process the continuous queue, a Message Driven Bean (MDB) is used. The MDB is disabled by default and must be enabled.

2 Basic Configuration Steps

There are some basics configuration steps to get started:

1. Define the global directory.
2. Create an external system to send and receive transactions
3. Enable publish channels for outbound transaction
4. Enable enterprise services for inbound transaction
5. Enable the event listener for the publish channel for a transaction to go out

The administrator needs to define a global directory that is used for the generated schema files, the XML files, error files, and so on.



The Integration Framework uses this global directory. The **delete**, **DeleteHistory**, **error**, and **retry** directories are per queue.

The Integration Framework creates the global directory and the subdirectories under the global directory when they are needed. For example, when a schema is created, it creates the **schema** subdirectory and puts the generated schema there. And the default output location for an XML end point is the **xmlfiles** subdirectory.

In a cluster environment, the global directory must be accessible to all the members of the cluster.

2.1 System Properties Application

The Base Services manages properties with the **System Properties** application.

The Base Services comes with many system properties predefined and the Integration Framework defines additional system properties for its own use. The Integration Framework properties begin with **mx.int.xxx**.

The following basic system properties are some of those associated with the Integration Framework:

- **mx.int.adminfromemail**: The e-mail address that will appear in the **from** field when the Integration Framework sends e-mail messages to an administrator.
- **mx.int.admintoemail**: One or more e-mail addresses separate by commas where the Integration Framework will send messages in case of errors.
- **mx.int.dfltuser**: Identifies the user that the Integration Framework uses to log in to the application for the inbound transactions.
- **mx.int.globaldir**: Defines the global directory to be used by the Integration Framework.

2.1.1 Setting System Properties

The two properties to initially set up are the global directory **mx.e.int.globaldir** and the Web application server URL **mx.e.int.webappurl**. The global directory is created as a subdirectory under the root directory where the application is installed. The URL for the servlet follows the format **http://<host>:port/meaweb**.

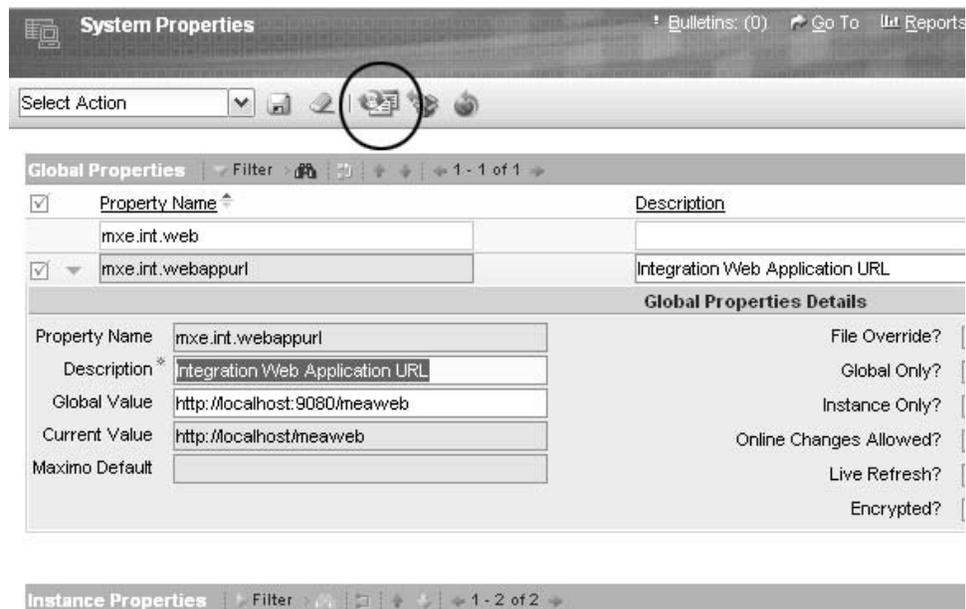
To set a system property, open the **System Properties** application and navigate to **System Configuration > Platform Configuration > System Properties**.

Filter with the prefix used by the Integration Framework properties, which is **mx.e.int**. There are 17 system properties that the Integration Framework uses.

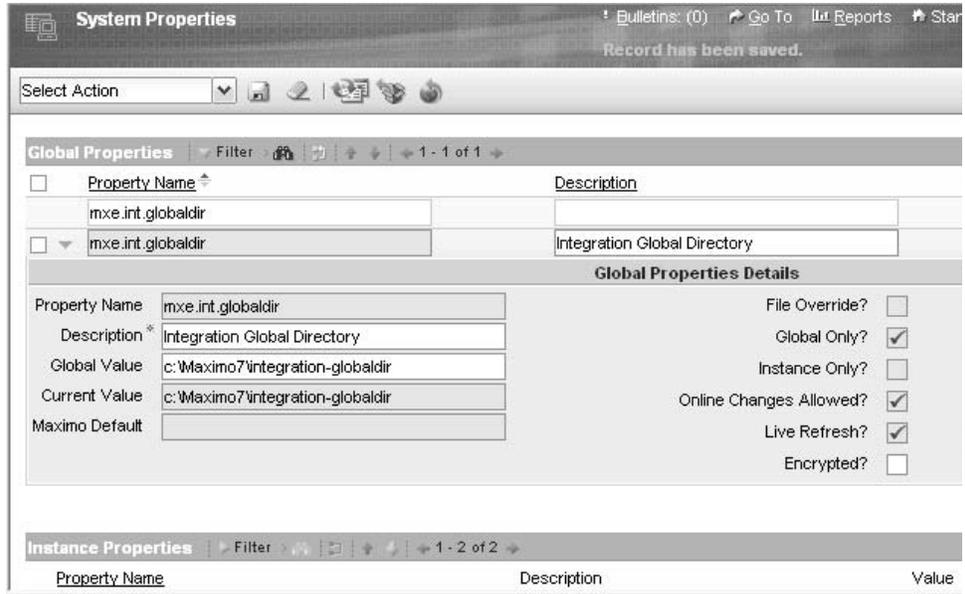
To change the system property value for the Web application server URL, filter or look for the property **mx.e.int.webappurl**. The default value does not have a port. The illustration below shows adding 9080 to specify a port for the WebSphere application server. The default port for IBM WebSphere is 9080.

The value **http://localhost:9080/meaweb** is the URL for an Integration Framework servlet that processes HTTP requests.

The system property value is not effective until the properties are refreshed. The **System Properties** application has the option of refreshing values dynamically. Select the property to refresh and click the **Live Refresh** icon. Confirm the change by clicking **OK**, and the new current value is displayed.

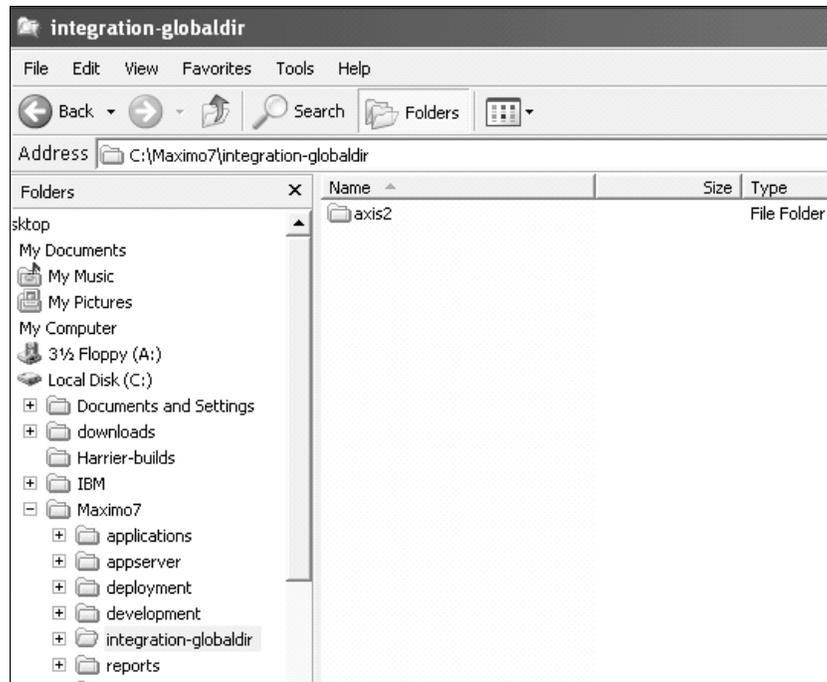


Other system properties, such as **mx.e.int.globaldir** can be set in a similar manner.



There is no default value for **mx.e.int.global.dir**. The Integration Framework will use the application server directories unless a value is defined.

For example, the global directory could be **c:\Maximo7\integration-globaldir**. After the new global directory is active, the first directory that the Integration Framework creates is **axis2**. Other directories are created as needed.



2.2 Queues

The Integration Framework has default JMS queues defined. They are referenced by their JNDI names.

The sequential queues are:

- **jms/maximo/int/queues/sqout** for the outbound sequential queue
- **jms/maximo/int/queues/sqin** for the inbound sequential queue

The inbound continuous queue is **jms/maximo/int/queues/cqin**.

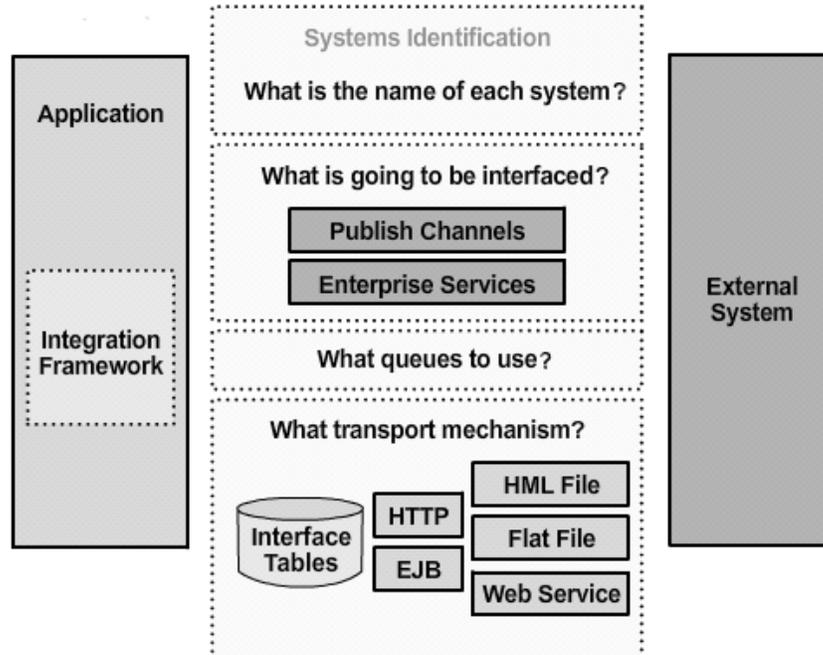
To improve WebSphere's performance, a second queue, **jms/maximo/int/queues/cqinerr**, is defined for transactions with errors. When a transaction in the continuous queue is processed and produces an error, the transaction is removed from the continuous queue and put in the error queue.

Continuous queues are processed using instances of a Message Driven Bean. Sequential queues are processed using a predefined cron task that needs to be activated.

2.3 Integrate to an External System

To integrate the application to an external system, it is necessary to:

1. Identify the external system with a name to differentiate it from other external systems.
2. Specify what is going to be interfaced. How are outbound and inbound transactions going to be processed? For example, what publish channels will be used for outbound, and will enterprise services support inbound?
3. Identify what inbound and outbound queues to use. In most cases the default queues that are predefined in the Integration Framework are used, but it is possible to define other queues.
4. Define the transport mechanism to use. HTTP, an XML file, EJB, and so on.

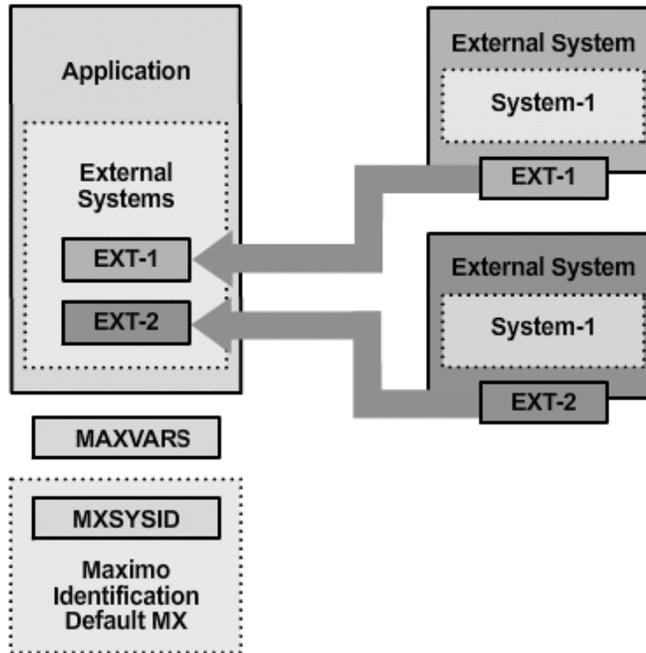


2.3.1 Identifying External Systems

To identify the application (Maximo Asset Management, CCMDB, and so on) and the external system, use the **MXSYSID** attribute in the **MAXVARS** table.

The application name, used for system identification, is stored in **MXSYSID**. The default value is **MX**.

The external system is identified by giving it a name in the **External Systems** application.



It is possible to have any number of external systems. This illustration has two external systems: **EXT-1** and **EXT-2**. The name of the application is **MX**, which is the default name.

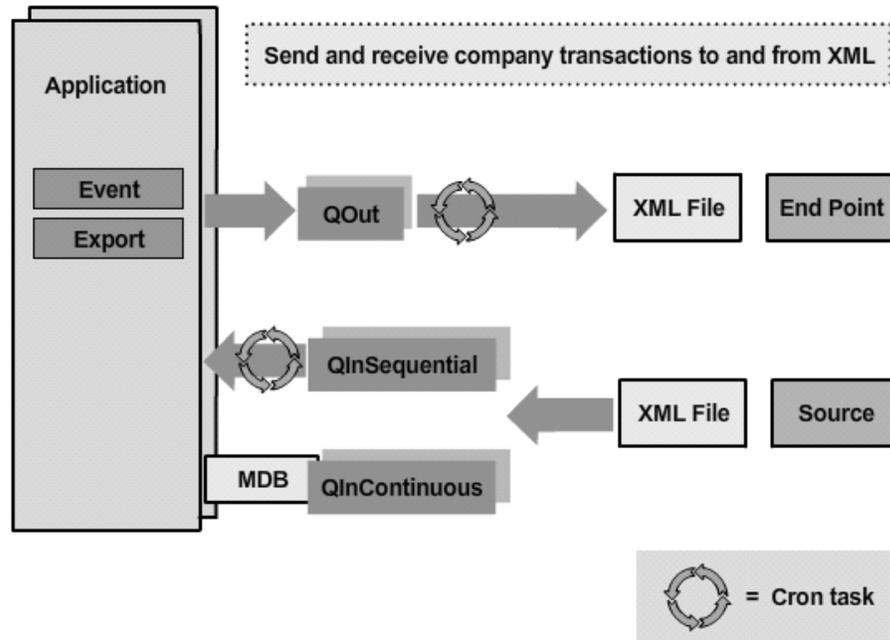
2.3.2 External System Application

To explain how to use the **External System** application, an example will define an external system that can send and receive **Companies** using XML files as the transport mechanism.

The **Companies** object has a predefined object structure named **MXVENDOR** in the Integration Framework.

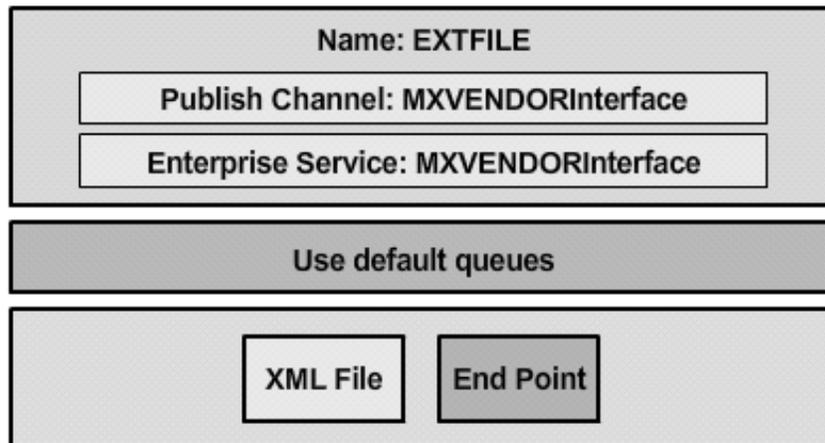
A transaction can be sent with an event or a manual export.

The XML is placed in the queue and sent to the end point.



The XML can come back through a sequential queue or a continuous queue, depending on which you choose to configure.

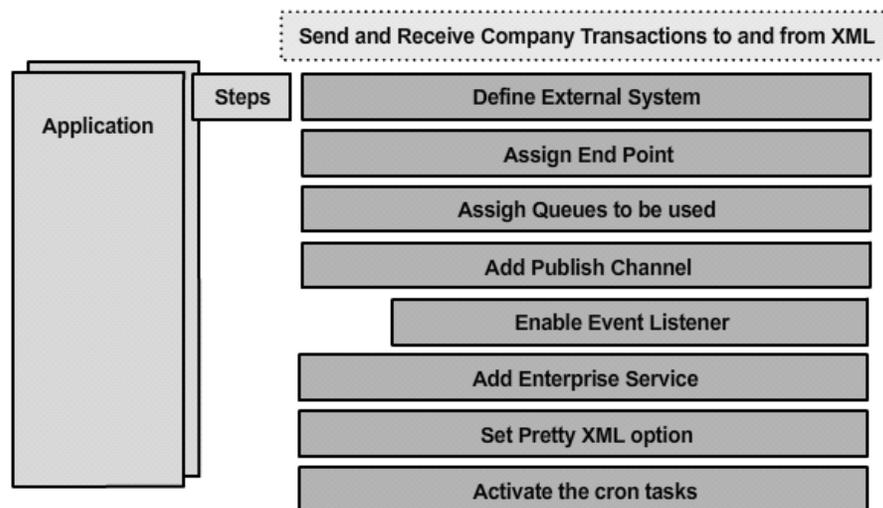
2.3.3 Example: Create an External System



Details and steps to create this external system to send and receive transactions for Companies are as follows:

1. The name for the external system is **EXTFILE**, and it must be defined. It will support sending and receiving transactions for the **Companies** object.
2. The object structure name is **MXVENDOR**.

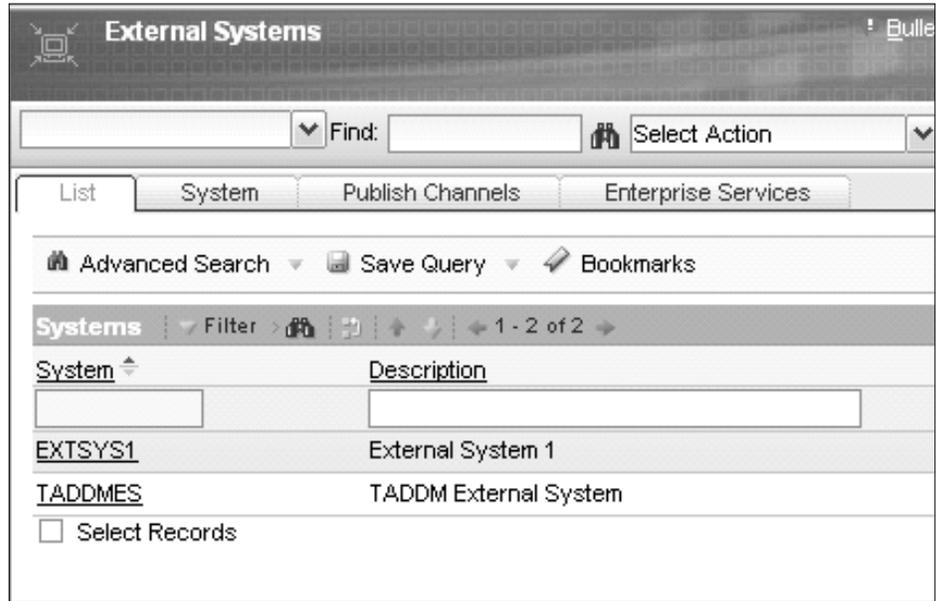
3. For outbound, the publish channel used is **MXVENDORInterface**. Add the publish channel.
4. For outbound transactions to be processed by events, it is necessary to enable the event listener for the publish channel.
5. For inbound, the enterprise service used is **MXVENDORInterface**. Add the enterprise service.
6. Assign the queues. Use the predefined default queues.
7. The final destination is an XML type endpoint. Assign the end point.
8. To see and edit the XML file, use the **Pretty** option for XML. If this option is not selected, the XML is one long string.
9. The cron tasks associated with the queues need to be activated.



To open the **External System** application, navigate to **Go To > Integration > External Systems**.

The **External System** application contains a **List**, **System**, **Publish Channels**, and **Enterprise Services** tab. The **List** tab is the standard List tab. The **System** tab shows the details for the selected external system. The **Publish Channels** tab shows the outbound interfaces that are defined for the current external system. The **Enterprise Services** tab shows the inbound interfaces that are defined for the current external system.

From the **List** tab, press the Return key to retrieve the list of the predefined external systems. The Integration Framework has two predefined external systems. **EXTSYS1** is for generic integration and has all the predefined publish channels and all the predefined enterprise services. **TADDMES** is a special external system used for integration with the Tivoli Application Dependency Discovery Manager. These two external systems should not be used or modified.



2.3.3.1 Create a New External System

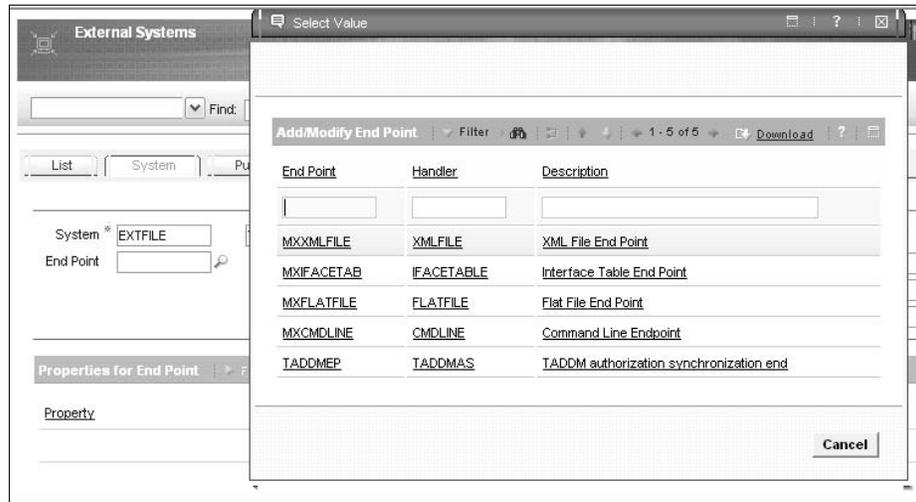
Always define your own external system with only the publish channels and enterprise services that will be used.

Within the **External System** application, click the **New** icon to create a new external system. For this example, name the external system *EXTFILE*.

2.3.3.2 Select an End Point

To select the final destination for the transactions, next to the **End Point** field, click the select list icon to view the list of available end points. From this list, select **MXXMLFILE**. The Integration Framework has the following predefined end points:

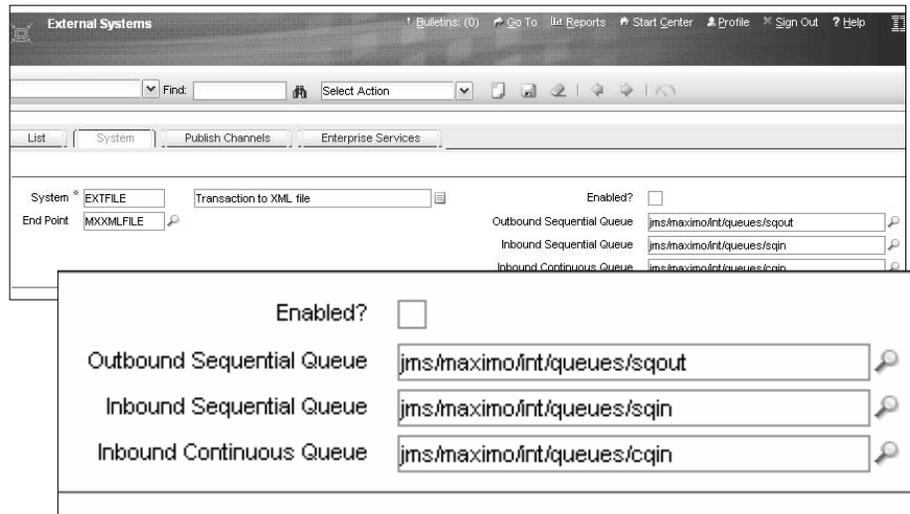
- **MXXMLFILE**: The transaction is written to an XML file.
- **MXIFACETAB**: The transaction is written to interface tables.
- **MXFLATFILE**: The transaction is written to a flat file.
- **MXCMDLINE**: This is a special end point to invoke a command in an external system. This end point uses SSH (Secure SHell) and provides a way to invoke a command on a remote system.
- **TADDMEP**: This end point should only be used with the Tivoli Application Dependency Discovery Manager.



2.3.3.3 Specify Queues

Next, specify what JMS queues are going to be used by this external system. Select the **Outbound Sequential Queue**. The JNDI names of the available queues are displayed. In this case, select the one queue named **jms/maximo/int/queues/sqout**.

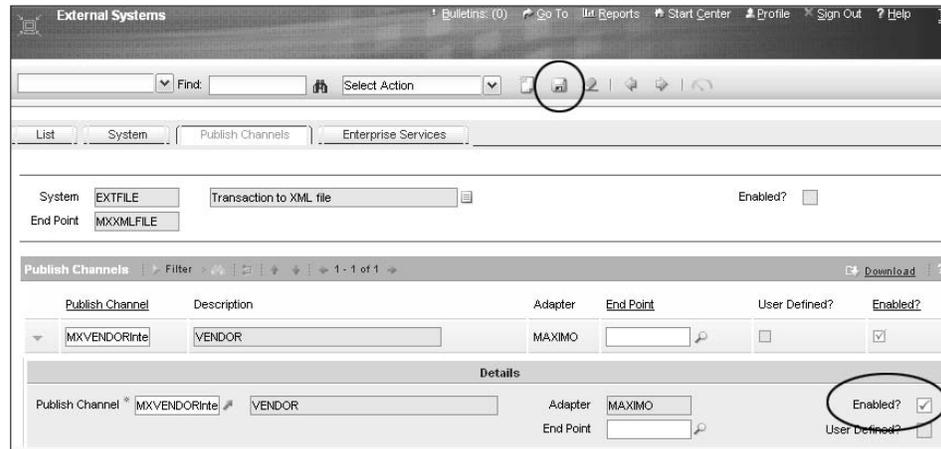
Follow the same process for the other two queues. Select **jms/maximo/int/queues/sqin** for the **Inbound Sequential Queue** and **jms/maximo/int/queues/cqin** for the **Inbound Continuous Queue**.



2.3.3.4 Select a Publish Channel

To select the publish channels and enterprise services for the Companies (Vendor) interface, click the **Publish Channels** tab. Click **New Row** to add a publish channel. The external system can have none, one, or more than one publish channels. Click the **Detail Menu** icon and **Select Value** to see the list of available publish channels. Scroll to look for the VENDOR interface, which is named **MXVENDORInterface**.

The publish channels are not used until enabled and the change is saved.



A new feature in the Integration Framework is that each channel can have a different end point, so the end point is not just associated with the external system. The external system can have a default end point and some of the publish channels assigned to the external system can have a different end point.

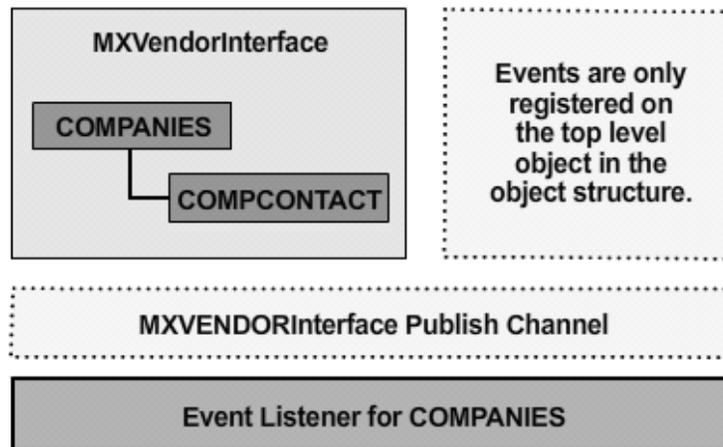
2.3.3.5 Events

Transactions go out when an action occurs, such as the approval of a work order or creation of a new company.

The Base Services capture events. The application registers the events for which to listen and the Java class to execute when the event occurs. When an event occurs, the Event Manager looks at the registered listener and all the listeners registered for that event are executed. The event must be activated.

The Integration Framework registers event listeners for the main object in the object structure. The main object for Vendor is Companies, and related to Companies are the Company Contacts.

The event listener is registered for Companies. Should an action (add, insert, delete, update, and so on) occur to the Companies object, the event listener is executed.

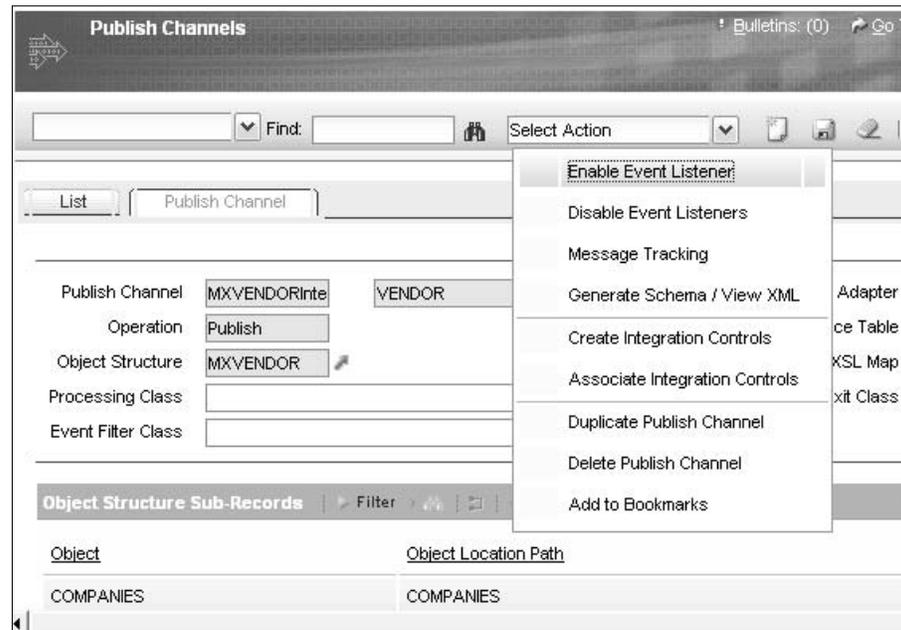


If a contact is added to a company without changing the Companies object, the event listener is not executed.

2.3.3.6 Enable Event Listeners

To enable the event listeners, use the **Publish Channels** application. Navigate to **Go To > Integration > Publish Channels**.

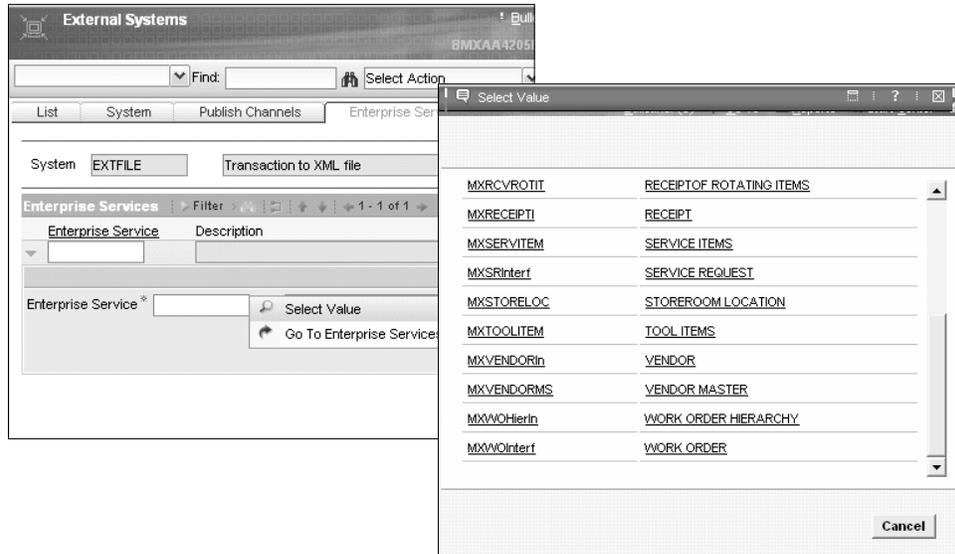
Find the **MXVENDORInterface** publish channel. From the **Select Action** menu, select **Enable Event Listener** to enable the event listener for the publish channel **MXVENDORInterface**.



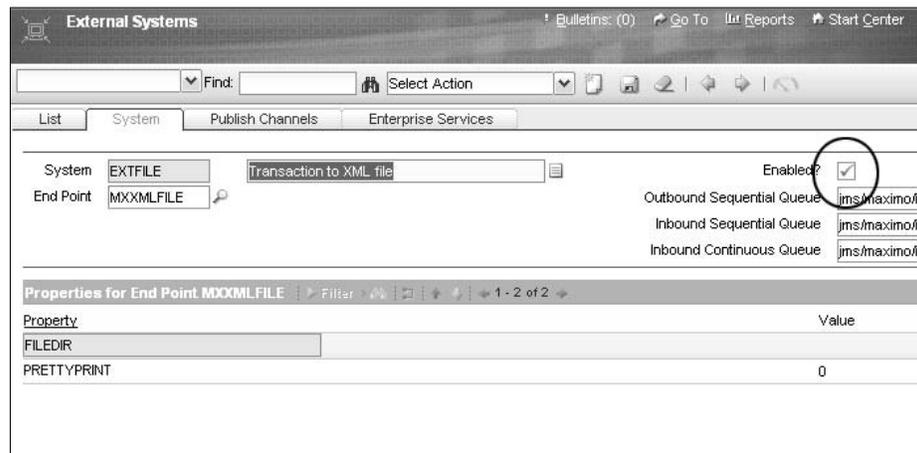
Every external system with this publish channel enabled will process the event and generate an outbound transaction.

To assign the inbound enterprise services to use with this external system, go back to the **External Systems** application, list the **EXTFILE** external system, and select the **Enterprise Services** tab. Click **New Row** to add an enterprise service.

The **Select Value** option displays available enterprise services for **Companies**. Scroll down to find the Vendor Service. Select **MXVENDORInterface**.



The enterprise service needs to be enabled to use it. The external system itself also needs to be enabled, which is done by selecting the **System** tab and checking the **Enabled** check box.



After the changes are saved, it is possible to send and receive Companies (Vendors) to and from XML files.

Thus far, these actions have been performed:

- An external system has been defined.
- An XML end point has been assigned.
- The default continuous queue has been assigned.
- A publish channel MXVENDORInterface was added.
- The event listener for outbound transactions was enabled.
- The MXVENDORInterface enterprise service was added and enabled.
- The external system was enabled.

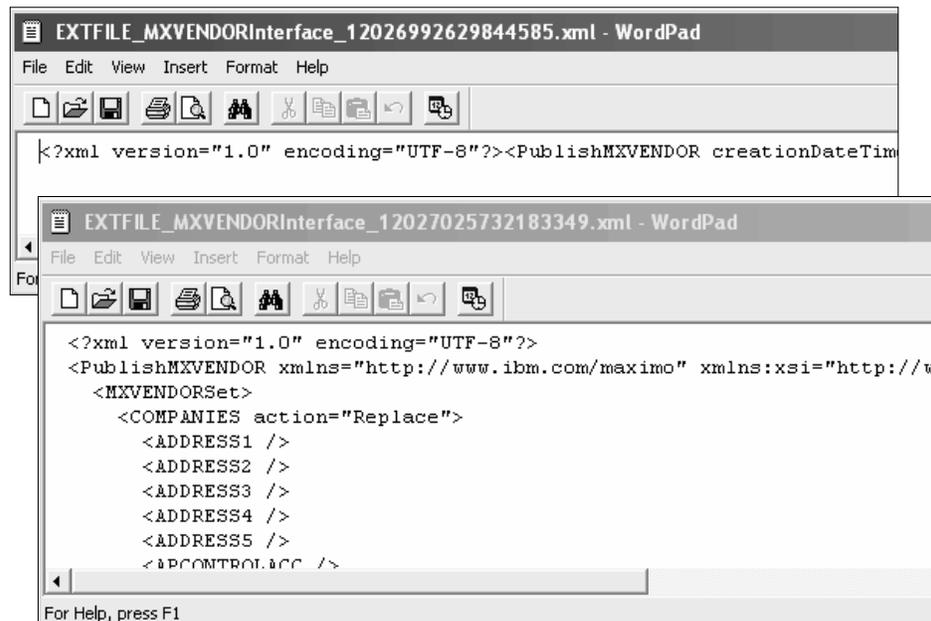
The next steps will be to set a special option for XML to easily edit the XML file generated from the application and to activate the cron tasks associated with the outbound and inbound queues.

2.3.3.7 Set XML Option

Use the **End Point** application to set the XML option for **PRETTYPRINT**.

Navigate to **Go To > Integration > End Point** and modify the XML end point **MXXMLFILE** used by the external system **EXTFILE**.

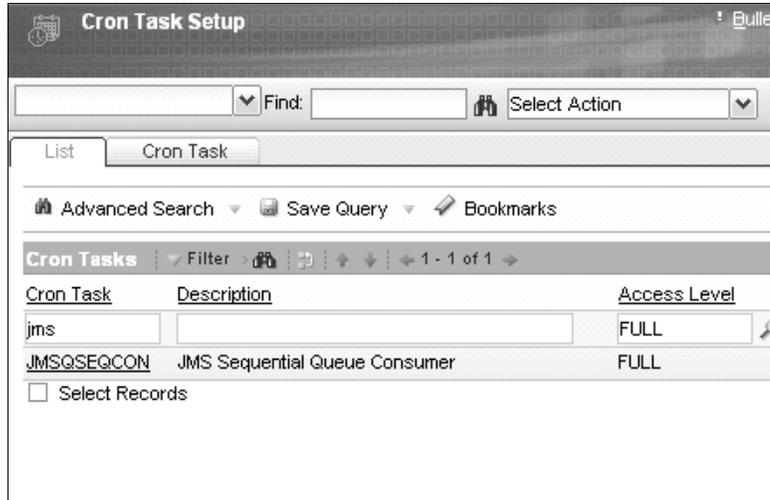
The **MXXMLFILE** end point has two properties: **FILEDIR** and **PRETTYPRINT**. **FILEDIR** is the directory where the XML output is written. By default, the file goes to the the global directory **xmlfiles** subdirectory. The **PRETTYPRINT** option default is 0 (false). By default, the XML is generated as one string, which is more efficient. To facilitate editing the XML file, set the option to true (1). The following illustration highlights the two options.



2.3.3.8 Activate the Cron Tasks

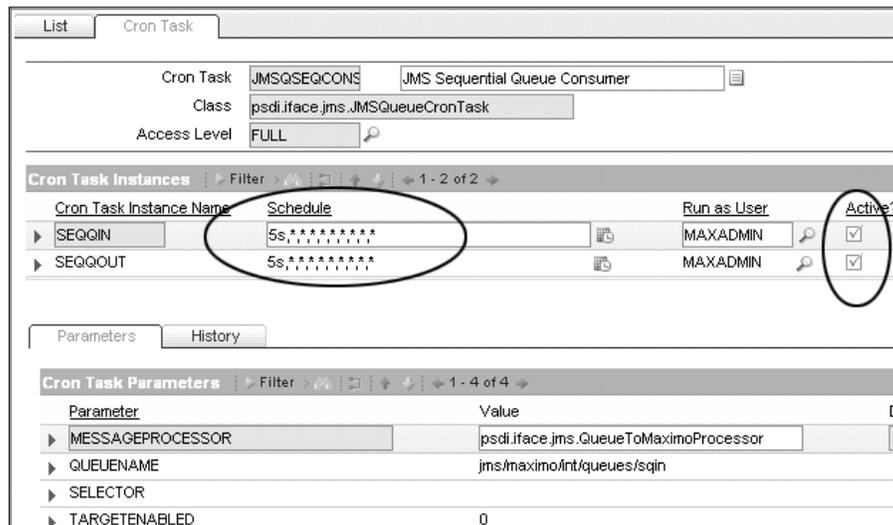
The cron tasks associated with the JMS queues used to store the transactions must be activated using the **Cron Tasks Setup** application. Navigate to **Go To > System Configuration > Cron Tasks Setup**.

The Integration Framework has a predefined cron task named **JMSQSEQCONSUMER**.



It has two predefined instances: **SEQQIN** and **SEQQOUT**. **SEQQIN** is a sequential queue for inbound transactions, and **SEQQOUT** is a sequential queue for outbound transactions.

The default schedule time for both cron tasks is 30 seconds. The time can be changed to 5 seconds for development for both instances of the cron task. Click the **Active** check box and then save so that the cron tasks start.



2.3.4 The Continuous Queue

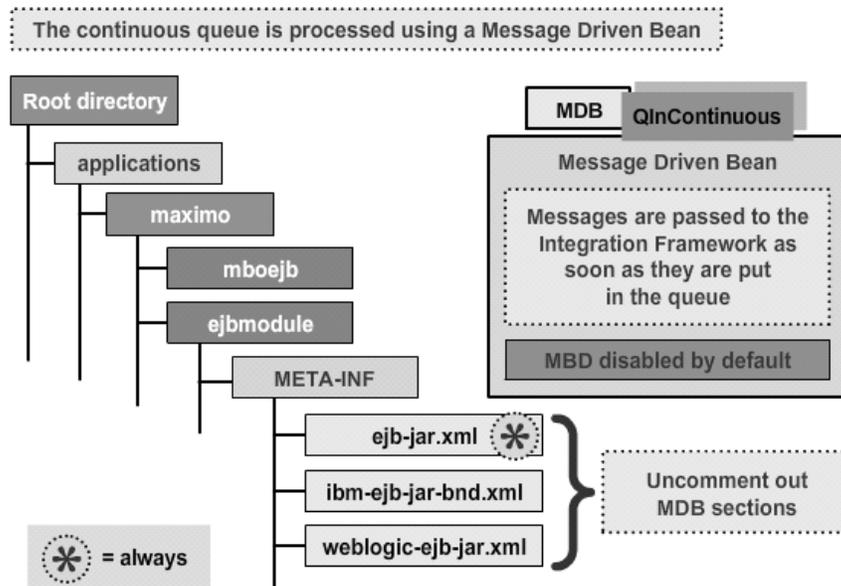
The continuous queue is processed using a Message Driven Bean associated with a JMS queue. When a message is put in the queue, the MDB is called and a message is passed.

2.3.4.1 Enable the MDB

The MDB is not enabled by default. To enable the MDBs, it is necessary to edit files located in the **root\applications\maximo\mboejb\ejbmodule\META-INF** directory.

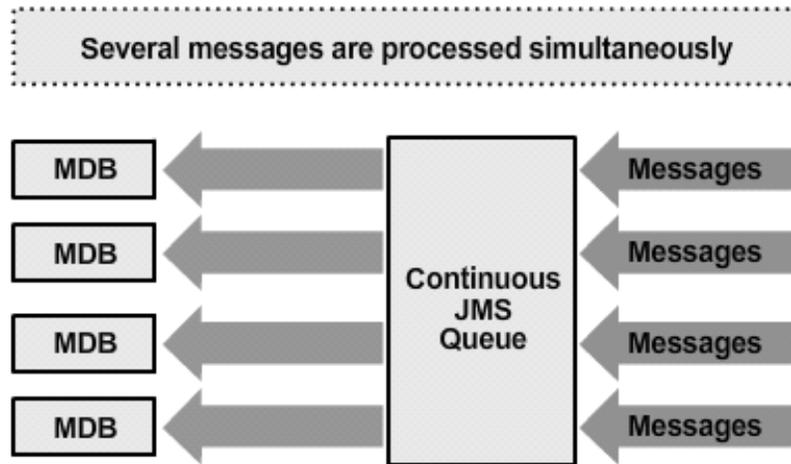
The **ejb-jar.xml** file must always be edited whether the application server is IBM WebSphere or BEA WebLogic. The sections referring to MDB need to be uncommented. The **ejb-jar.xml** has four sections for the MDB definition that are commented out by default.

For IBM WebSphere, the **ibm-ejb-jar-bnd.xml** file has two sections with the definitions to bind the MDB to the JMS queues. It is necessary to rebuild the ear file after changing these files for the changes to take effect.



2.3.4.2 Processing the Continuous Queue

The continuous queue's process is shown in the following illustration:



Transactions in the continuous queue are independent; so multiple messages can be processed simultaneously.

The number of MDBs that can be instantiated is configurable. The IBM WebSphere default is 5 and the BEA WebLogic default is 8.

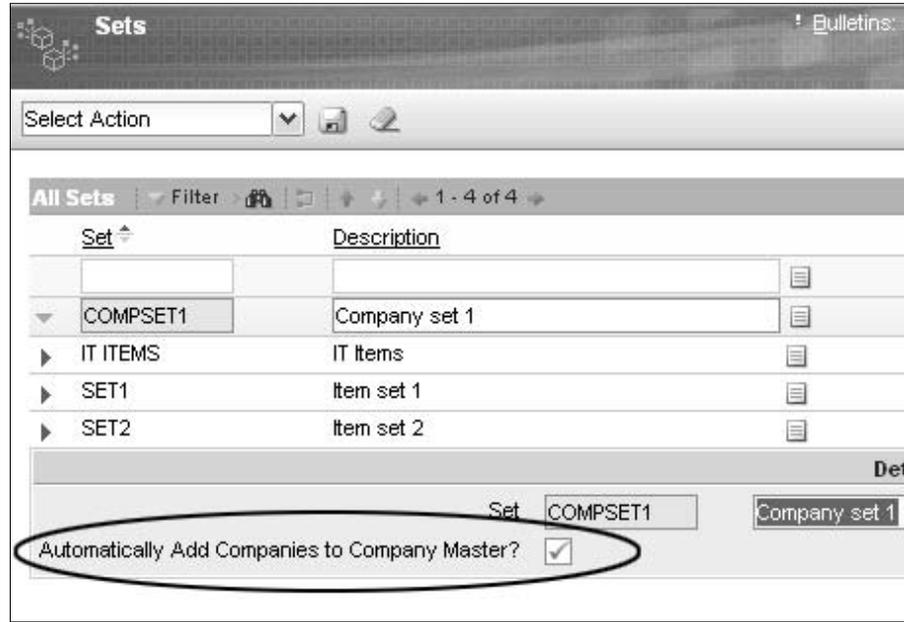
2.4 Produce an Event and a Message

To produce an event and message, this example will use Companies. For those readers not familiar with the **Companies** application or a Companies object, the Companies object must have a Company Master object.

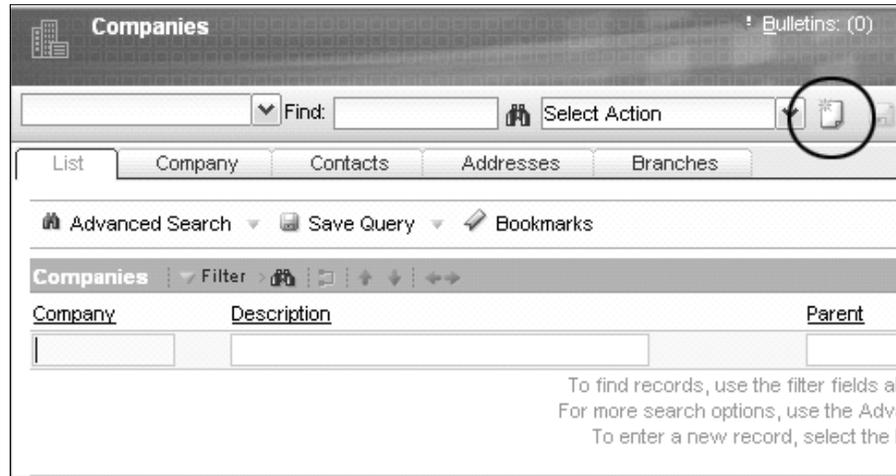
A special option is available for companies in which the company master is automatically created when a company is created. The option is available in the **Sets** application located at **Go To > Administration > Sets**.

The **COMPSET1** set is a default set. Expand the set by clicking the arrow on the left.

The **Automatically Add Companies to Company Master** option is off by default. When the automatic option is on, it creates a company master when a new company is created. This option is unique to a Companies set.



To test the configuration changes made thus far, open the **Companies** application by navigating to **Go To > Purchasing > Companies**. To create a new Company, click the **New** icon.



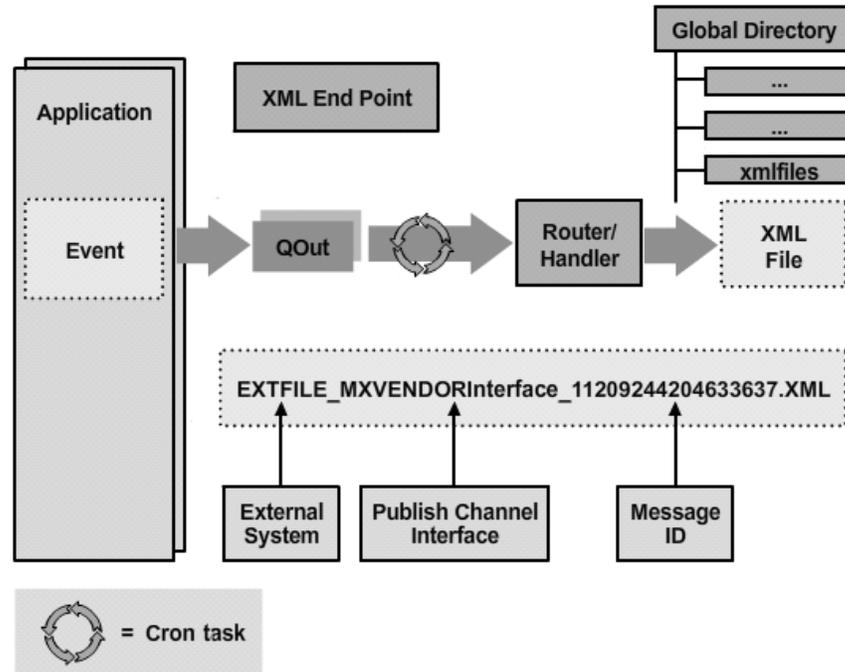
Because the automatic option is set, there is no problem within the **Companies** application if the company master does not exist. It will create a company master automatically.

The screenshot shows the 'Companies' form in IBM Maximo. The 'Company *' field is highlighted with a red circle and contains the value 'COFILE1'. The 'Company' field contains 'Company cofile'. Below these fields are 'Parent', 'Customer #', and 'Home Page' fields. The 'Purchasing Details' section contains several fields: 'Currency *' (USD), 'Tax Code', 'Tax Exempt Code', 'Tax Exempt Number', 'Freight Terms', 'FOB Point', 'Ship Via', and 'Registration #'. A 'Select Action' dropdown menu is visible in the top right corner, also circled in red.

The new company name is **COFILE1**. Saving the new company will generate an event. The event manager will call the Integration Framework's registered listener. The Integration Framework listens for the event and sends a transaction.

The publish channel listener for **MXVENDORInterface** was activated; so the outbound transaction will be sent out to the end point associated with the external system with the enabled publish channel.

2.5 Transaction Flow



When the event for the new Company occurs, the Integration Framework listener is called, and the transaction is put into the sequential outbound queue.

The cron task associated with the queue looks into the queue every 5 seconds. If there is a message, it reads the message and calls the Router/Handler to send the transaction to the end point. In this case, the end point is an XML file. The default directory where the XML file is written is the **xmlfiles** subdirectory under the global directory.

The name of the XML file uses the following convention: First is the name of the external system. In this example the name is **EXTFILE**. It is followed by the name of the publish channel, which in this case is **MXVENDORInterface**. The name ends with a unique message ID that is generated when the transaction is written to the JMS queue.

2.6 Logging

Informational messages (marked with **[INFO]** prefix) are displayed in the Console if the integration logger is active. When a message is written or retrieved from a queue, a message is written to the log.

```

0 13 Feb 2008 18:11:00:796 [INFO] SIGOPTIION: mbosets (1), mbos (3144)
0 13 Feb 2008 18:11:00:796 [INFO] SYNONYMDOMAIN: mbosets (1), mbos (2)
0 13 Feb 2008 18:11:00:796 [INFO] WEAPPTOOLBAR: mbosets (4), mbos (4)
0 13 Feb 2008 18:11:00:796 [INFO] Host Name:192.168.230.128 ,Server Name:MXServer ,Cron Task Nam
0 13 Feb 2008 18:11:00:796 [INFO] Host Name:192.168.230.128 ,Server Name:MXServer ,Cron Task Nam
0 13 Feb 2008 18:11:00:796 [INFO] Host Name:192.168.230.128 ,Server Name:MXServer ,Cron Task Nam
0 13 Feb 2008 18:11:00:796 [INFO] Host Name:192.168.230.128 ,Server Name:MXServer ,Cron Task Nam
0 13 Feb 2008 18:11:00:796 [INFO] Host Name:192.168.230.128 ,Server Name:MXServer ,Cron Task Nam
0 13 Feb 2008 18:11:00:796 [INFO] Host Name:192.168.230.128 ,Server Name:MXServer ,Cron Task Nam
0 13 Feb 2008 18:11:00:796 [INFO] Host Name:192.168.230.128 ,Server Name:MXServer ,Cron Task Nam
0 13 Feb 2008 18:11:00:796 [INFO] Host Name:192.168.230.128 ,Server Name:MXServer ,Cron Task Nam
0 13 Feb 2008 18:11:00:796 [INFO] Host Name:192.168.230.128 ,Server Name:MXServer ,Cron Task Nam
0 13 Feb 2008 18:11:00:874 [INFO] Server Host:192.168.230.128 ,Server Name:MXServer ,Number of
0 13 Feb 2008 18:11:00:874 [INFO] Total number of users connected to the system:2
0 13 Feb 2008 18:11:00:874 [INFO] Memory Total = 536870912 ,Free = 210274000
0 id=====1202943499618
0 13 Feb 2008 18:11:02:640 [INFO] wrote data to Queue jms/maximo/int/queues/sqout
0 13 Feb 2008 18:11:07:140 [INFO] got message from queue jms/maximo/int/queues/sqout
    
```

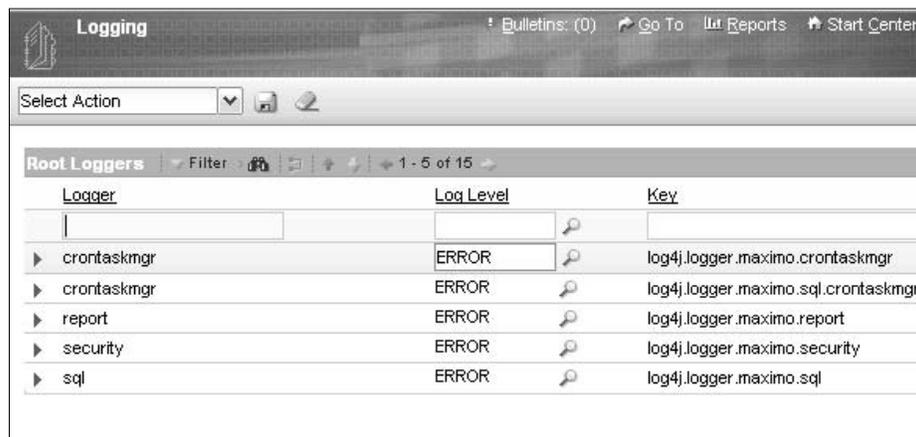
The Integration Framework uses **log4j** for logging.

2.6.1 The Logging Application

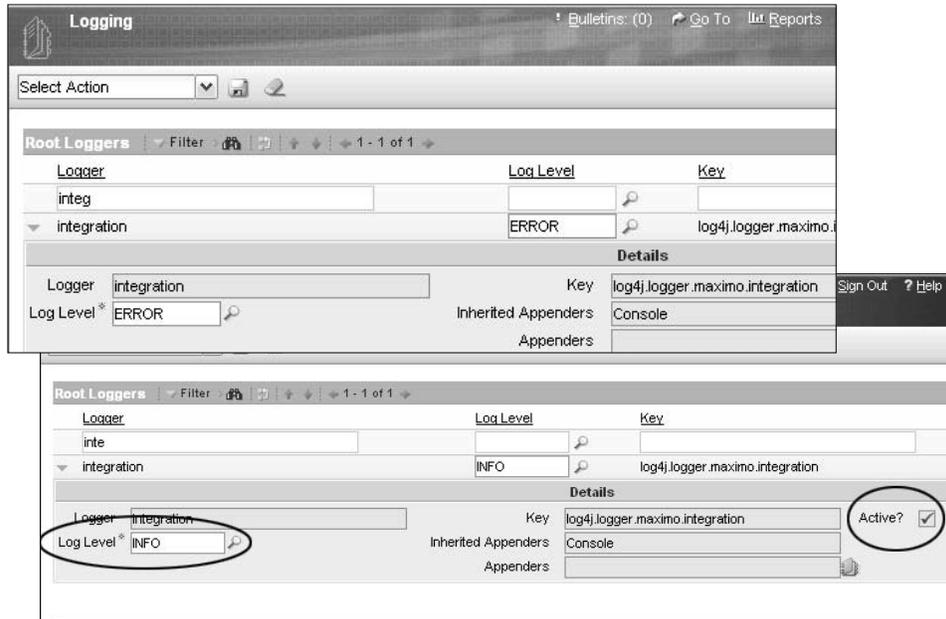
By default, the Integration Framework logger is set to **ERROR** and only writes in case of error. To show more details, the log level can be changed to **INFO**.

To change the logging level, open the **Logging Application** by navigating to **Go To > System Configuration > Platform Configuration > Logging**.

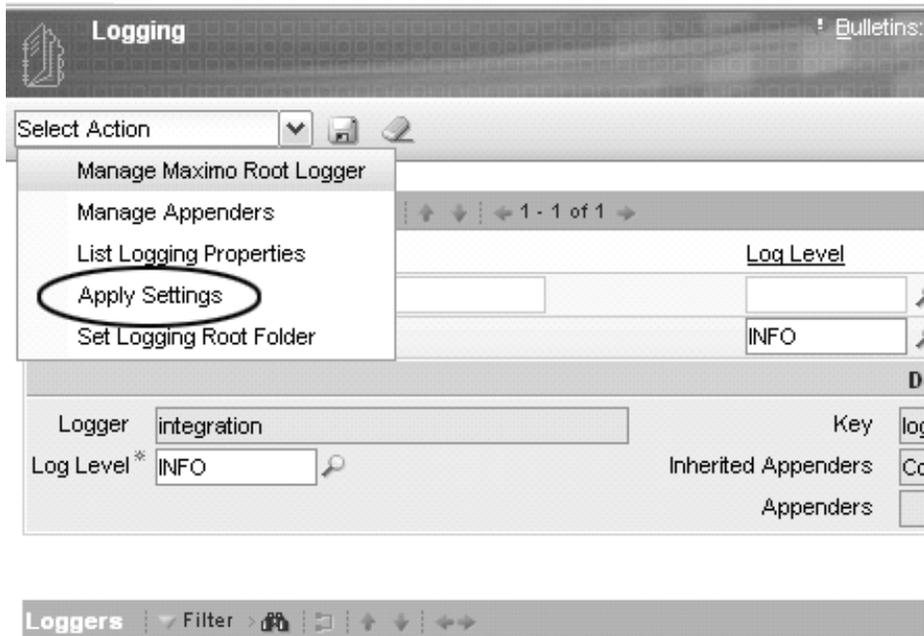
The **Logging** application displays loggers that have been defined.



The name of the Integration Framework logger is **integration**. Search for **integration** and expand the logger fields to see the details. Change the **ERROR** level to **INFO**.



For a logger to be used, it must be activated. The integration logger is not active by default. Change the logger to active, and save the changes. The changes become effective when the action **Apply Setting** is executed.



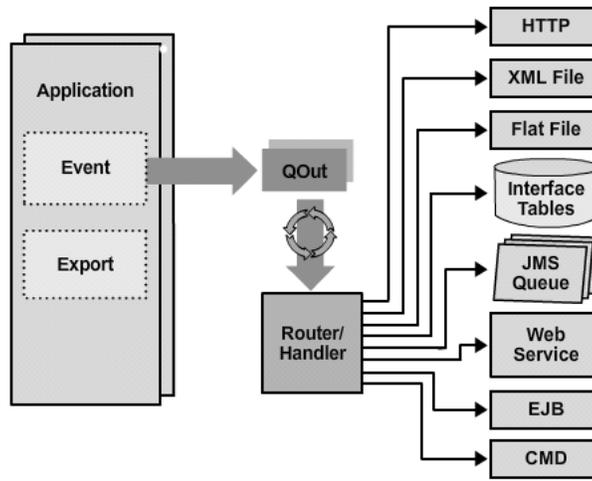
Where the logging output is display depends on the application server. For IBM WebSphere, the cumulative log file is **SystemOut.log**. It is located in the server directory.

2.7 Summary

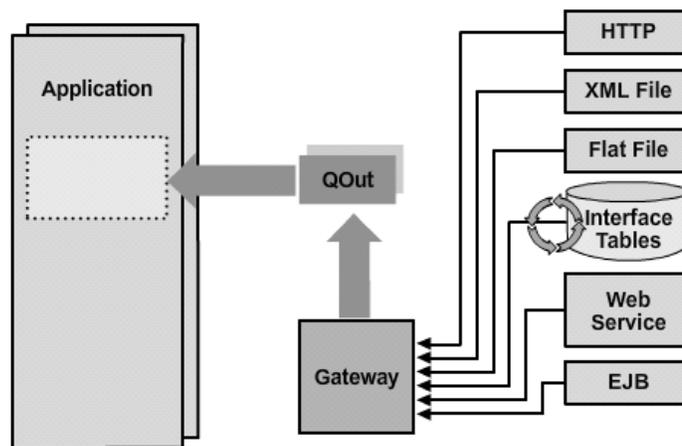
The Integration Framework allows data to flow in and out of applications (Maximo Asset Management, CCMDB, and so on) and to flow in and out of external systems.

Both inbound and outbound messages can be sent synchronously and asynchronously.

The external systems to which the data flows are supported by various protocols such as HTTP or writing to a file. There are many ways to send data to another system.



Likewise, data flowing into the system can be received using many different protocols.



And the data flow is not limited to one system. Data can flow to many systems at one time. These systems can be OMP systems, custom systems, or other enterprise systems.



Conclusion

Summary

You should now have an understanding of basic configuration regarding the Integration Framework.

Acknowledgements

Special thanks to Guido Viarisio, IBM Software Developer, for developing graphics to accompany this paper, in addition to sharing information regarding the Integration Framework technology and components.

