



TIBCO ActiveMatrix BusinessWorks™

Palette Reference

Version 5.16.1 | February 2025

Document Updated: September 2025

Contents

Contents	2
Process Palette	27
Process Definition	27
Configuration	27
Process Variables	28
Partners	29
ActiveEnterprise Adapter Palette	30
Common Features Across ActiveEnterprise Activities	30
Support for Message Filter Resources	30
Internationalization Support	31
Refresh Adapter Service Button	31
Transport Tab	32
Configure SSL Button	42
Adapter Request-Response Server	44
Configuration	45
Transport	46
Misc	46
Output	47
Adapter Subscriber	48
Configuration	49
Transport	50
Misc	50
Output	50
Invoke an Adapter Request-Response Service	52
Configuration	53
Transport	54

Input	54
Output	55
Error Output	56
Publish to Adapter	57
Configuration	58
Transport	59
Input	60
Output	61
Error Output	62
Respond to Adapter Request	63
Configuration	64
Input	64
Output	65
Error Output	65
Send Exception to Adapter Request	65
Configuration	66
Input	66
Output	67
Error Output	67
Wait for Adapter Message	68
Configuration	68
Transport	69
Event	69
Input	70
Output	71
Error Output	72
Wait for Adapter Request	72
Configuration	73
Transport	74
Event	74
Input	75
Output	76

File Palette	78
Copy File	78
Configuration	78
Input	79
Output	79
Error Output	79
Create File	80
Configuration	80
Input	81
Output	81
Error Output	82
File Poller	83
Configuration	83
Advanced	84
Misc	86
Output	87
List Files	88
Configuration	88
Input	89
Output	89
Error Output	90
Read File	91
Configuration	91
Input	91
Output	92
Error Output	93
Remove File	94
Configuration	94
Input	94
Output	94
Error Output	95

Rename File	96
Configuration	96
Input	97
Output	97
Error Output	98
Wait for File Change	98
Configuration	99
Event	100
Input	101
Output	102
Write File	103
Configuration	103
Input	104
Output	105
Error Output	106
FTP Palette	107
FTP Change Default Directory	107
Configuration	107
Input	108
Output	108
Error Output	109
FTP Connection	109
Configuration	109
Configure SSL Button	111
Test Connection Button	112
Available Commands Button	112
FTP Delete File	113
Configuration	113
Input	113
Output	115
Error Output	115

FTP DIR	115
Configuration	115
Input	116
Output	117
Error Output	118
FTP Get	118
Configuration	118
Input	120
Output	122
Error Output	123
FTP Get Default Directory	123
Configuration	124
Input	124
Output	125
Error Output	125
FTP Make Remote Directory	126
Configuration	126
Input	127
Output	128
Error Output	128
FTP Put	128
Configuration	129
Input	130
Output	132
Error Output	132
FTP Quote	133
Configuration	133
Input	134
Output	135
Error Output	136
FTP Remove Remote Directory	136

Configuration	136
Input	137
Output	138
Error Output	138
FTP Rename File	139
Configuration	139
Input	139
Output	141
Error Output	141
FTP Sys Type	141
Configuration	141
Input	142
Output	143
Error Output	143
General Activities Palette	145
Assign	145
Configuration	145
Input	146
Output	146
Call Process	146
Configuration	146
Dynamically Determining the Process to Call	147
Input	148
Output	149
Catch	149
Configuration	150
Input	150
Output	150
Checkpoint	151
Configuration	151
Input	152

Checkpoints and Transactions	152
Called Processes	152
Recovering After a Crash	153
Confirm	154
Configuration	155
Custom Activity	156
Configuration	156
Engine Command	157
Configuration	157
Input	158
Output	159
Commands	159
External Command	171
Long-Running Commands	171
Configuration	172
Input	174
Output	176
Error Output	177
Generate Error	177
Configuration	178
Input	179
Output	179
Get Shared Variable	179
Configuration	180
Output	180
Inspector	180
Configuration	181
Input	182
Output	182
JNDI Configuration	182
Configuration	183

Test Configuration Button	184
Advanced	184
IBM WebSphere and JNDI Context Caching	185
Job Shared Variable	186
Configuration	186
Schema	186
Initial Value	187
Label	187
Configuration	188
Lock Object	189
Configuration	189
Mapper	190
Input Editor	190
Input	191
Output	192
Notify	192
Configuration	192
Input	193
Output	194
Notify Configuration	194
Configuration	194
Data	195
Null	195
On Event Timeout	195
Configuration	196
Output	197
On Notification Timeout	197
Configuration	198
Output	199
On Shutdown	199
Configuration	199

On Startup	200
Configuration	200
Receive Notification	201
Configuration	201
Misc	202
Output	203
Rethrow	203
Configuration	203
Input	203
Output	204
Set Shared Variable	204
Configuration	204
Input	205
Output	205
Shared Variable	205
Configuration	206
Schema	207
Initial Value	207
Sleep	208
Configuration	208
Input	208
Timer	209
Configuration	209
Misc	210
Output	211
Wait	211
Configuration	212
Input	212
Output	213
Write To Log	213
Configuration	214

Input	214
Output	215
Custom Properties for the Write To Log Activity	215
OnError	216
Configuration	216
Misc	217
Output	217
HTTP Palette	218
HTTP Connection	218
Configuration	219
Configure SSL Button	221
Advanced	222
Custom Properties for the HTTP Palette	224
HTTP Receiver	228
Configuration	229
Advanced	231
Misc	232
Special Characters in HTTP Requests	233
Output Headers	233
Output	235
Proxy Configuration	240
Configuration	240
Send HTTP Request	241
Configuration	241
Sending Data in the HTTP Request	244
Configure SSL Button	246
Advanced	247
Input Headers/Output Headers	248
Input	251
Output	255
Persistent Connections	258

Error Output	262
Send HTTP Response	263
Configuration	263
Input Headers	264
Input	265
Error Output	268
Wait for HTTP Request	269
Configuration	269
Event	270
Output Headers	271
Input	273
Output	273
Error Output	277
Java Palette	278
Show Class Browser	278
Common Features of Java Code and Java Method	280
Accessing ActiveMatrix BusinessWorks Global Variables and Java System Properties	280
Sending Messages to The ActiveMatrix BusinessWorkss log4j File	281
Java Code	281
Configuration	282
Code	283
Compiling Java Code	284
Passing Java Objects Between Java Code Activities	286
Input	287
Output	287
Error Output	287
Java Custom Function	288
Configuration	290
Example Java Function	291
Java Event Source	293

Configuration	294
JavaProcessStarter Abstract Class	295
Code	296
Compiling Java Code	297
Advanced	298
Misc	299
Output	300
Error Output	300
Java Global Instance	301
Configuration	302
Advanced	303
Java Method	304
Configuration	304
Advanced	306
Caching the Java Class	308
Input	309
Output	310
Error Output	310
Java Schema	311
Configuration	312
Schema Text	313
Java To XML	313
Configuration	316
Advanced	317
Input	318
Output	318
Error Output	319
XML To Java	319
Configuration	320
Input	320
Output	321
Error Output	321

JDBC Palette	322
JDBC Call Procedure	322
Configuration	323
Advanced	324
Input	325
Output	326
Error Output	328
JDBC Connection	328
Configuration	328
Database Drivers	334
Connection Pooling	336
Test Connection Button	336
JDBC Get Connection	336
Configuration	337
Input	337
Output	338
Error Output	339
Using The JDBC Connection in Java	339
JDBC Query	340
Configuration	341
Fetch Button	342
Prepared Parameters	342
Advanced	343
Input	345
Fetching Subsets of the Result Set	346
Output	347
Error Output	349
JDBC Update	350
Configuration	350
Prepared Parameters	351
Advanced	352

Input	353
Output	354
Error Output	355
Query Designer	356
SQL Direct	361
Configuration	361
Advanced	362
Input	362
Output	363
Error Output	365
Time Zone Construction	366
Examples	366
Time Zone Names	366
JMS Palette	373
Enabling JMS in ActiveMatrix BusinessWorks	373
Get JMS Queue Message	374
Configuration	374
Advanced	377
Input	377
Output Editor	378
Output	378
Error Output	380
JMS Application Properties	381
Configuration	381
JMS Connection	382
Configuration	382
Test Connection Button	389
Advanced	389
SSL Configuration	390
JMS Properties	392
JMS Queue Receiver	396

Configuration	396
Misc	399
Load-Balancing of Incoming Messages	400
Advanced	401
Output Editor	402
Output	403
JMS Queue Requestor	404
Configuration	405
Advanced	406
Input Editor/Output Editor	407
Input	408
Output	410
Error Output	412
JMS Queue Sender	413
Configuration	413
Advanced	414
Input Editor	416
Input	416
Output	419
Error Output	419
JMS Topic Publisher	420
Configuration	420
Advanced	421
Input Editor	423
Input	423
Output	426
Error Output	426
JMS Topic Requestor	427
Configuration	427
Advanced	428
Input Editor/Output Editor	430
Input	430

Output	433
Error Output	435
JMS Topic Subscriber	436
Configuration	436
Advanced	440
Misc	441
Output Editor	441
Output	441
Reply To JMS Message	443
Configuration	443
Advanced	444
Input Editor	446
Input	446
Output	449
Error Output	449
Wait for JMS Queue Message	450
Configuration	450
Advanced	453
Message Event	453
Input	454
Output Editor	455
Output	455
Error Output	457
Wait for JMS Topic Message	457
Configuration	458
Advanced	460
Message Event	461
Input	463
Output Editor	463
Output	464
Error Output	465
Reconnecting to JMS Servers	466

Mail Palette	467
Receive Mail	467
Configuration	467
Configure SSL Button	470
Test Connection Button	470
Advanced	470
Output Headers	472
Misc	472
Output	473
Send Mail	476
Configuration	476
Configure SSL Button	478
Support for STARTTLS	479
Advanced	479
Input Headers	480
Input	480
Output	485
Error Output	485
Parse Palette	487
Data Format	487
Configuration	488
Data Format	490
Delimiter Separated Fields	491
Field Offsets	492
Parse Data	493
Configuration	493
Input	495
Output	496
Error Output	499
Parsing a Large Number of Records	499
Render Data	500

Policy Palette	503
Security Policy	503
Configuration	503
Authentication	505
Integrity	507
Confidentiality	510
Timeout	513
Security Policy Association	513
Configuration	514
Inbound	515
Outbound	516
Support for Unqualified Elements	517
Rendezvous Palette	518
RawRVMsg2 Schema	518
The RawRVMsg2 Schema	518
Adding the RawRVMsg2 Schema to Your Project	519
Datatype Conversion	520
TIBRVMSG_XML, TIBRVMSG_IPADDR32, and TIBRVMSG_IPPORT16	
Datatypes	522
Raw TIBCO Rendezvous Message Support	523
Using the RawRVMsg2 Schema	524
Field Names In Messages	525
Internationalization Support	526
Publish Rendezvous Message	527
Configuration	527
Input Editor	528
Input	528
Output	529
Error Output	529
Rendezvous Subscriber	530
Configuration	530

Misc	532
Output Editor	533
Output	533
Rendezvous Transport	533
Configuration	534
Configure SSL Button	535
Advanced	536
Reply to Rendezvous Request	538
Configuration	538
Input Editor	539
Input	539
Error Output	540
Send Rendezvous Request	540
Configuration	540
Input/Output Editor	542
Input	543
Output	543
Error Output	544
Wait for Rendezvous Message	544
Configuration	545
Event	546
Input	547
Output Editor	548
Output	548
Error Output	549
RMI Palette	550
RMI Overview	550
Using RMI to Call a Remote Object	550
Acting as a RMI Server	551
RMI Lookup	552
Configuration	552

Input	553
Output	553
RMI Server	554
Configuration	554
Misc	555
Output Editor	555
Output	555
Service Palette	557
Context Resource	557
Configuration	557
Schema	558
Get Context	558
Configuration	558
Output	559
Invoke Partner	559
Configuration	559
Input	560
Output	560
Error Output	561
Partner Link Configuration	561
Configuration	561
Advanced SOAP Settings	571
Receive Partner Notification	579
Configuration	579
Misc	580
Output	580
Error Output	580
Service	581
Configuration	583
Advanced SOAP Settings	591
Partner Binding	600

WSDL Source	601
Overview	602
Set Context	602
SOAP Palette	605
SOAP Specification Compliance	605
Retrieve Resources	607
Using Retrieve Resources	607
Configuration	610
Input	610
Output	611
Error Output	612
Built-in Resource Provider	612
SOAP Event Source	614
WSDL File and SOAP Event Source	614
Configuration	615
Transport Details	616
Messages With Attachments	620
Advanced	621
WSDL Source	626
Misc	627
Output	627
Error Output	637
SOAP Request Reply	638
Configuration	638
Reload Button	639
Test Transport Connection	639
Transport Details	639
Configure SSL	646
Messages With Attachments	647
Advanced	649
Input	650

Output	656
Error Output	659
SOAP Send Fault	660
Configuration	661
Transport	661
Input	662
Output	663
Error Output	663
SOAP Send Reply	664
Configuration	664
Transport	665
Input	666
Output	667
Error Output	667
SOAP Messages Over the JMS Transport	667
WSDL Extensions	670
MimeParser	675
Configuration	675
Advanced	675
Input	676
Output	676
Error Output	676
TCP/IP Palette	678
Overview of the TCP/IP Palette	678
Using the TCP/IP Palette Activities	678
Read TCP Data	679
Configuration	679
Input	680
Output	681
TCP Close Connection	682
Configuration	682

Input	683
Output	683
TCP Connection	683
Configuration	683
Advanced	685
Connection Pooling	686
TCP Open Connection	686
Configuration	687
Input	687
Output	688
TCP Receiver	688
Configuration	688
Setting TCP Keep-Alive Parameters	689
Misc	690
Output	690
Wait for TCP Request	691
Configuration	691
Event	692
Input	692
Output	693
Write TCP Data	693
Configuration	693
Input	694
Output	695
Transaction Palette	696
JTA UserTransaction Configuration	696
Configuration	696
Test Connection Button	698
Transaction State	699
Configuration	699
Input	700

Output	700
Error Output	700
XA TransactionManager Configuration	701
Configuration	701
WSDL Palette	703
WSDL	703
Configuration	703
Toolbar	705
Editing WSDL Resources	705
Copying WSDL Definitions	707
External References	710
Message	710
Configuration	711
Toolbar	712
Editing Message Components	712
PortType	712
Configuration	712
Toolbar	713
Editing PortType Components	713
Operation	713
Configuration	713
Toolbar	715
Editing Operation Components	715
XML Activities Palette	716
Parse XML	716
Parsing Date and Datetime Strings	716
Configuration	716
Input	717
Output Editor	718
Output	718

Error Output	719
Render XML	719
Rendering Date and Datetime Strings	720
Configuration	720
Advanced	721
Input Editor	721
Input	721
Output	722
Error Output	722
Transform XML	723
Configuration	723
Input	724
Output	725
Error Output	726
Example of Transforming XML	726
XSLT File	728
Specifying Data Schema	730
Editor	730
TIBCO Product Documentation and Support Services	737
Legal and Third-Party Notices	740

Process Palette

The **Process** palette allows you to create process definitions.

Process Definition

Resource



The Process Definition resource allows you to define business processes. Drag and drop the Process Definition resource into the design panel to create a new process definition.

Select a process definition in the project tree panel to edit a process definition. When editing process definitions, the palette panel changes to a list of activity palettes that you can use in a process definition. Subsequent sections of this manual describe the activity palettes.

For more information about creating and editing process definitions, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the process.
Description	Short description of the process.
Custom Icon File	You can choose your own custom icon for the process, if desired. Use the Browse button to locate an image file (GIF, JPEG, and so on) to use as the icon for this process.

Field	Description
Namespace Registry	Clicking the Edit button in this field allows you to view, add, change or delete the namespaces used in the input partners defined for this process definition. You can also view and edit schema and WSDL imports.
Target Namespace	Target namespace of the process definition. This field is mandatory when specifying partners in the process definition. This field is generated automatically whenever you create a process definition. While creating new processes, it is recommended that the targetNamespace value generated automatically be retained. Typically, this value is <code>http://xmlns.example.com/<rand_num></code>, where <code><rand_num></code> is replaced by a random number. If you want to specify partners in processes that were created in older versions, then you must provide the targetNamespace manually. Ensure that the value specified for targetNamespace is unique across the project.

Process Variables

The Process Variables tab allows you to specify user-defined process variables for the process definition. For a complete description of user-defined process variables, see the Processes section of *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

You can assign a value to the process variables with the [Assign](#) activity. For more information about assigning a value to a user-defined process variable, see [Assign](#).

This tab contains two panels:

- The left panel allows you to add, delete, and rename process variables.
- The right panel allows you to define data schemas for the process variable selected in the left panel. For more information about specifying data schema, see [Specifying Data Schema](#).

Partners

The Partners tab allows you to define partners for the process definition. The partner can be either located inside the same project or it can be an external service that can be invoked over the Internet by way of SOAP protocol.

The Partners tab has the following fields.

Field	Description
Partners	The Partners tab allows you to specify binding information for partner definitions. The port type and port type namespace fields in the table are non-editable. To modify the value of the Port Type for a Partner, select the Browse button at the right corner of the field Port Type below the Partners table. Partner Name - specifies the name used by the process to reference a partner. Click the Browse button on the right side to add a new Partner. Port Type - This is a non-editable field displaying the Port Type selected. PortType Namespace - This is a non-editable field displaying the PortType Namespace corresponding to the Port Type selected. Partner Link - specifies the partner link configuration for the partner. This field can optionally be specified by double-clicking on the field and browsing the list of available partner link resources in the project.
Port Type	Allows you to select an interface (port type) from the list of available partner interfaces in the project. To add or modify the Port Type, click the Browse Resources button at the right end of the field and select the Port Type you want to add to the Partners tab.

ActiveEnterprise Adapter Palette

The **ActiveEnterprise Adapter** palette contains activities for communicating with configured TIBCO ActiveEnterprise adapters. This section assumes you are familiar with the concepts and features of TIBCO ActiveEnterprise adapter products. For more information about configuring adapters, see the documentation for your adapter product.

 **Note:** You can use these activities to communicate with adapters that are supported by TIBCO ActiveMatrix BusinessWorks™. For information about whether ActiveMatrix BusinessWorks™ is supported, see your adapter documentation .

Common Features Across ActiveEnterprise Activities

The activities in the **ActiveEnterprise Adapter** palette reference Adapter Configuration resources in your project. This section describes common features across all activities in this palette.

Support for Message Filter Resources

An Adapter Configuration resource can use a Message Filter resource. Message filters are used to provide custom, user-written code to transform the messages sent or received from an adapter, call out to TIBCO Adapter SDK functions, or perform other tasks. For more information about Message Filter resources, see *TIBCO Designer™ Palette Reference*.

The activities in the **ActiveEnterprise Adapter** palette support the use of Message Filter resources. When using TIBCO Designer™, the message filter is specified for the Adapter Configuration, and the ActiveEnterprise Adapter activities automatically make use of the class specified in the Message Filter resource.

When you deploy your project, however, you must ensure that the ActiveMatrix BusinessWorks process engine has access to any classes specified in a Message Filter

resource. For example, if you have a **Publish to Adapter** activity in your process definition, the process engine must have access to the class used when executing the `transformOutbound()` message filter call. If you deploy your TIBCO Adapter and ActiveMatrix BusinessWorks process engine on the same machine, no additional configuration steps are required. If the components are installed on different machines, you must place your message filter class on the machine where your process engine is deployed and make certain the location of the message filter class is in the CLASSPATH for the process engine (this is set in the `bwengine.tra` file).

For more information about deploying process engines, see *TIBCO ActiveMatrix BusinessWorks™ Administration*.

Internationalization Support

ActiveMatrix BusinessWorks and TIBCO Adapter products can send text messages that are encoded using either the Latin-1 or UTF-8 character encoding. Both the sender and receiver of the message must use the same encoding. The character encoding sent by resources configured in TIBCO Designer is set at the project level during design time. For deployed projects, TIBCO Administrator™ allows you to set the message encoding.

At design time, the message encoding property can be set when creating a new project. In the Save Project dialog, choose either ISO8859-1 or UTF-8 in the TIBCO Messaging Encoding option list. For more information about saving projects, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*. After you choose the messaging encoding; it takes effect at design time, such as in debug mode, or if the project is running in a standalone mode before deployment.

After the project is deployed, the messaging encoding is determined by the encoding property of the TIBCO Administrator domain. This property is set when the domain is created. ISO8859-1 and UTF-8 are the two supported character encoding options. The messaging encoding set for the domain overrides any encoding set for a project.

Refresh Adapter Service Button

All activities in the **ActiveEnterprise Adapter** palette can use preconfigured adapter services. These adapter service configurations can change after an adapter service is specified in the configuration of an **ActiveEnterprise Adapter palette** activity. Therefore, the configured adapter service may become inconsistent with your process definition.

The **Refresh Adapter Service** button is available on most activities in the **ActiveEnterprise Adapter** palette. This button allows you to synchronize your activities with the adapter service specified in the activity. Any changes that have occurred to the adapter service are reflected in the activity after the refresh completes.

For example, an adapter service may change the schema it uses for incoming or outgoing messages. When this schema change occurs, any activity using that adapter service may no longer have the correct input and output schemas. Clicking the **Refresh Adapter Service** button changes the activity to use the current configuration of the adapter service. This invalidates input mappings, and you may need to use the **Mapper Check and Repair** button on the Input tab to correct the activity input schemas.

i Note: The **Refresh Adapter Service** button is only enabled when you use all settings from the Adapter Service. That is, if you change the Transport Type on the Transport tab to anything other than Adapter Service Default, the **Refresh Adapter Service** button becomes disabled. If you want to keep your ActiveEnterprise Adapter activities synchronized with an adapter service, do not change the Transport Type. If you want to use a different transport configuration for the activity, you should check the Custom AESchema field on the Configuration tab and specify the adapter service schema in the Custom AESchema Reference field, and then modify the settings on the Transport tab.

Transport Tab

Transports specify the underlying communication characteristics for messages sent and received by adapter services. Normally, the activities of the **ActiveEnterprise Adapter** palette use the same transport configuration as the adapter service with which they are communicating. However, in some situations, you may want to override the transport configuration of the adapter service and use different transport parameters.

The Transport tab is available in activities in the **ActiveEnterprise Adapter** palette. This tab allows you to override the transport configuration of the adapter service. The Transport tab has different fields depending upon which transport is selected in the Transport Type field. [The Transport tab](#) describes the fields of the transport tab.

The Transport tab

Field	Global Var?	Description
Transport Type	No	The transport receives the incoming request. This is normally the transport used by the adapter service, but you can override this by specifying different transport parameters. The possible values for this field are: • Adapter Service Default • Tibrv Reliable • Tibrv Certified • Tibrv Certified Message Queue • JMS Topic • JMS Queue The sub-sections of this table describe the fields that appear when each of these options is selected as the transport type.

Tibrv Reliable Transport

For more information about these fields, see TIBCO Rendezvous documentation.

Subject	Yes	The subject of the incoming request.
Service	Yes	The service parameter of the transport of the incoming request.
Network	Yes	The network parameter of the transport of the incoming request.
Daemon	Yes	The daemon parameter of the transport of the incoming request.
SSL	No	Specifies that SSL should be used when connecting to the TIBCO Rendezvous daemon. For more information about configuring SSL parameters, see Configure SSL Button .

Field	Global Var?	Description
Message Format	Yes	Specifies the message format for the message. This can be either AE message format or AE XML format. This field is only available on the Transport tab of the following activities: • Adapter Subscriber • Publish to Adapter • Wait for Adapter Message

Tibrv Certified Transport

For more information about these fields, see TIBCO Rendezvous documentation.

Subject	Yes	The subject of the incoming request.
Service	Yes	The service parameter of the transport of the incoming request.
Network	Yes	The network parameter of the transport of the incoming request.
Daemon	Yes	The daemon parameter of the transport of the incoming request.
SSL	No	Specifies that SSL should be used when connecting to the TIBCO Rendezvous daemon. For more information about configuring SSL parameters, see Configure SSL Button .
CM Name	Yes	The CM name of the incoming request.
PreRegister Listeners	Yes	This field is only available on activities that publish messages (for example, Invoke an Adapter Request Response Service or Publish to Adapter). This field specifies the cmname(s) of the certified subscriber (s). This allows the publisher to pre-register with any expected listeners. If multiple cmnames are specified, separate each cmname with a comma.

Field	Global Var?	Description
Relay Agent	Yes	Name of the relay agent, if one is used.
Ledger File	Yes	Name and location of the ledger file. Warning: ActiveMatrix BusinessWorks constructs the default value of this field from global variables and the Adapter Service configuration. The value of this field must result in a valid path on the filesystem where the process is run. You may need to modify the default value in this field or modify the global variables used in this field to make sure the field results in a valid filesystem path.
Sync Ledger	Yes	When checked, this specifies the ledger file should be kept synchronous (asynchronous ledger updates are performed, if this field is unchecked).
Default Time Limit	Yes	The default message time limit for certified messages specified in milliseconds.
Require Old Messages	Yes	Specifies that unconfirmed messages should be kept in the ledger file.
Message Format	Yes	Specifies the message format for the message. This can be either AERV or XML. This field is only available on the Transport tab of the following activities: • Adapter Subscriber • Publish to Adapter • Wait for Adapter Message

Tibrv Certified Message Queue Transport

For more information about these fields, see TIBCO Rendezvous documentation.

Subject	Yes	The subject of the incoming request.

Field	Global Var?	Description
Service	Yes	The service parameter of the transport of the incoming request.
Network	Yes	The network parameter of the transport of the incoming request.
Daemon	Yes	The daemon parameter of the transport of the incoming request.
SSL	No	Specifies that SSL should be used when connecting to the TIBCO Rendezvous daemon. For more information about configuring SSL parameters, see Configure SSL Button .
CMQ Name	Yes	The CMQ name of the incoming request.
Complete Time	Yes	The amount of time (in seconds) the scheduler waits for a worker process to complete. If the worker process does not complete in the specified period, the scheduler reassigns the message to another worker.
Scheduler Weight	Yes	The scheduler weight for the process engine.
Scheduler Heartbeat	Yes	The scheduler heartbeat interval (in milliseconds).
Scheduler Activation	Yes	The scheduler activation interval (in milliseconds).
Worker Weight	Yes	The worker weight for the process engine.
Worker Tasks	Yes	The maximum number of incoming adapter requests this process engine can accept.
Message Format	Yes	Specifies the message format for the message. This can be either AERV or XML. This field is only available on the

Field	Global Var?	Description
		Transport tab of the following activities: • Adapter Subscriber • Wait for Adapter Message

JMS Topic Transport

For more information about these fields, see TIBCO Enterprise Message Service documentation.

Topic	Yes	The topic name of the incoming request.
Auto Confirm	No	Specifies whether the incoming request should be automatically confirmed. This field is only available for activities that receive messages (for example, Adapter Subscriber). If check box is not selected then the message gets acknowledged at a later point by using the Confirm activity. If the message is not confirmed before the process instance ends, the message is redelivered and a new process instance is created to handle the new incoming message. Ensure that your process definition confirms the message.
Transactional	No	This field is only available for activities that receive messages (for example, Adapter Subscriber). Check this field when a transaction that can process JMS messages is included in the process definition and you want the incoming message to be a part of the transaction. Any reply or exception message associated with the incoming message is also part of the transaction. The message is acknowledged when the transaction commits. Checking this field disables the Auto Confirm field. For more information about creating transactions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.

Field	Global Var?	Description
Override Transaction	No	This field is only available for activities that publish messages (such as Publish to Adapter or Invoke an Adapter Request-Response Service). By default, ActiveEnterprise Adapter activities that are included in the appropriate transaction group are included in the transaction. Check this field if you want to exclude the activity from the transaction.
Durable	No	Specifies whether the client making the request is a durable subscriber. The durable subscription name is a concatenation of the process definition name and the name of the activity. This field is only available on the Transport tab of the following activities: • Adapter Subscriber • Adapter Request-Response Server • Wait for Adapter Message • Wait for Adapter Request
Durable Name	No	When the Durable field is checked, you can specify the name of the durable subscription that is registered with the JMS server in this field. This field is only available when the Durable field is checked.
Delivery Mode	No	This field is only available on the Transport tab of the Publish to Adapter or Invoke an Adapter Request-Response Service activities. This field specifies the delivery mode of the published message. This can be either persistent or non-persistent.
Time To Live	Yes	This field is only available on the Transport tab of the Publish to Adapter or Invoke an Adapter Request-Response Service activities.

Field	Global Var?	Description
		This field specifies the expiration time (in milliseconds) for the message. Specifying zero signifies the message should never expire.
JNDI Lookups	No	Specifies that references to the JMS objects (ConnectionFactory, Topic, and so on) should be obtained from a JNDI server.
SSL	No	Specifies that SSL should be used when connecting to the JMS server. For more information about configuring SSL parameters, see Configure SSL Button .
Topic Connection Factory Name	Yes	Specifies the Topic ConnectionFactory object.
Connection User Name	Yes	The user name to use when connecting to the JMS server.
Password	Yes	The password to use when connecting to the JMS server.
Provider URL	Yes	The URL for the JMS server.
JMS Application Properties	No	Any application-specific message properties that is part of the message. This is specified by the JMS Application Properties shared configuration resource. Once specified, these properties appear on the Input or Output tab as "OtherProperties".
Transacted Session	Yes	Specifies if the AE Event Source is a part of a transaction. When selected, this field disables Transactional and Auto Confirm. The fields Precreate Transactions and XA Transaction Manager is not visible.
Precreate	Yes	When selected this field specifies that the transaction is

Field	Global Var?	Description
Transactions		created before a message arrives at the JMS receiver. This field is available when Transactional field is selected. Note: The Precreate Transactions feature should be used only with IBM MQ Websphere.
XA Transaction Manager	Yes	Specifies the XA Transaction Manager Configuration used to create the XA transaction. This field is available when Precreate Transactions field is selected.

JMS Queue Transport

For more information about these fields, see TIBCO Enterprise Message Service documentation.

Queue	Yes	The queue name of the incoming request.
Delivery Mode	No	This field is only available on the Transport tab of the Publish to Adapter or Invoke an Adapter Request-Response Service activities. This field specifies the delivery mode of the published message. This can be either persistent or non-persistent.
Time To Live	Yes	This field is only available on the Transport tab of the Publish to Adapter or Invoke an Adapter Request-Response Service activities. This field specifies the expiration time (in milliseconds) for the message. Specifying zero signifies the message should never expire.
Transactional	No	This field is only available for activities that receive messages (for example, Adapter Subscriber). Check this field when a transaction that can process JMS messages is included in the process definition and you want

Field	Global Var?	Description
		the incoming message to be a part of the transaction. Any reply or exception message associated with the incoming message is also part of the transaction. The message is acknowledged when the transaction commits. Checking this field disables the Auto Confirm field. For more information about creating transactions. see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Override Transaction	No	This field is only available for activities that publish messages (such as Publish to Adapter or Invoke an Adapter Request-Response Service). By default, ActiveEnterprise Adapter activities that are included in the appropriate transaction group are included in the transaction. Check this field if you want to exclude the activity from the transaction.
JNDI Lookups	No	Specifies that references to the JMS objects (ConnectionFactory, Queue, and so on) should be obtained from a JNDI server.
SSL	No	Specifies that SSL should be used when connecting to the JMS server. For more information about configuring SSL parameters, see Configure SSL Button .
Queue Connection Factory Name	Yes	Specifies the Queue ConnectionFactory object.
Connection User Name	Yes	The user name to use when connecting to the JMS server.
Password	Yes	The password to use when connecting to the JMS server.
Provider URL	Yes	The URL for the JMS server.

Field	Global Var?	Description
JMS Application Properties	No	Any application-specific message properties that will be part of the message. This is specified by the JMS Application Properties shared configuration resource. Once specified, these properties appear on the Input or Output tab as "OtherProperties".
Transacted Session	No	Specifies if the AE Event Source is a part of a transaction. When checked, this field disables the Precreate Transactions at the receiver and the fields Precreate Transactions and XA Transaction Manager Reference are not visible.
Precreate Transactions	Yes	When checked, this field specifies that the transaction is created before a message arrives at the JMS receiver. This field is available when Transactional field is selected. Note: The Precreate Transactions feature should be used only with IBM MQ Websphere.
XA Transaction Manager Reference	Yes	Specifies the XA Transaction Manager Configuration used to create the XA transaction. This field is available when Precreate Transactions field is selected.

Configure SSL Button

The **Configure SSL** button allows you to configure the SSL parameters when the transport supports SSL and the SSL field is checked. The following are the fields of the Configure SSL dialog.

Basic

The Basic tab of the SSL Configuration dialog allows you to specify the simplest required SSL configuration parameters for the connection. The Basic tab has the following fields:

Field	Description
Trusted Certificates Folder (JMS Transports)	Location of the trusted certificates on this machine. The trusted certificates are a collection of certificates from servers to whom you establish connections. If the server you want to establish a connection to presents a certificate that does not match one of your trusted certificates, the connection is refused. This prevents connections to unauthorized servers.
Daemon Certificate (TIBCO Rendezvous transports)	File containing one or more certificates from trusted certificate authorities. This file is checked when connecting to a daemon to ensure that the connection is to a daemon that is trusted. This prevents connections to rogue TIBCO Rendezvous daemons that attempt to impersonate trusted daemons. You can retrieve a daemon's certificate using the administration interface in TIBCO Rendezvous. For more information about obtaining certificates through the administration interface, see the TIBCO Rendezvous documentation. Once retrieved, you can select a folder in your project and import this certificate into the folder using the Tools>Trusted Certificates>Import Into PEM Format menu item.
Identity	This is an Identity resource used to authenticate to the JMS server or TIBCO Rendezvous daemon. The Browse button allows you to select from a list of appropriately configured Identity resources. For TIBCO Rendezvous transports, only Identity resources with the Type field set to Identity File or Username/Password are listed. For more information, see <i>TIBCO Designer™ Palette Reference</i>.

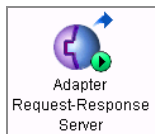
Advanced

The Advanced tab of the SSL Configuration dialog allows you to specify more advanced SSL configuration parameters for the connection. The Advanced tab is available only for JMS transports.

Field	Description
Trace	Specifies whether SSL tracing should be enabled during the connection. If checked, the SSL connection messages are logged and sent to the console.
Debug Trace	Specifies whether SSL debug tracing should be enabled during the connection. Debug tracing provides more detailed messages than standard tracing.
Verify Host Name	Specifies whether to verify that the host you are connecting to is the expected host. The host name in the host's digital certificate is compared against the value you specify in the Expected Host Name field. If the host name does not match the expected host name, the connection is refused.
Expected Host Name	Specifies the name of the host you are expecting to connect to. This field is only relevant if the Verify Host Name field is also checked. If the name of the host in the host's digital certificate does not match the value specified in this field, the connection is refused. This prevents other hosts from attempting to impersonate the specified host.
Strong Cipher Suites Only	When checked, this field specifies that the minimum strength of the cipher suites used can be specified with the <code>bw.plugin.security.strongcipher.minstrength</code> custom engine property. For more information about this property, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i>. The default value of the property disables cipher suites with an effective key length below 128 bits. When this field is unchecked, only cipher suites with an effective key length of up to 128 bits can be used.

Adapter Request-Response Server

Process Starter

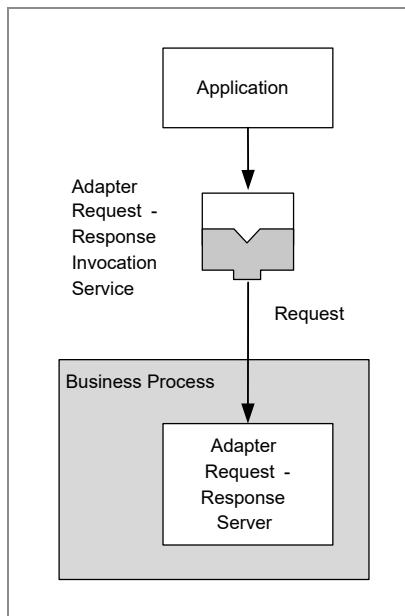


Starts a process based on the receipt of a request from an adapter. The adapter sends a request using a preconfigured request-response invocation service, and the process that is started acts as the implementation of the request. The process sends any required responses back to the adapter's service using the [Respond to Adapter Request](#) activity.

Adapter request-response invocation services are configured during adapter configuration, and the activity uses the information in the adapter configuration to fill in most of the fields of this activity. For more information about creating adapter configurations and creating adapter services, see your adapter documentation.

An [Adapter Request-Response Server activity](#) illustrates an **Adapter Request-Response Server** activity starting a business process to handle an incoming request from an adapter request-response invocation service. At a later point in the business process, the [Respond to Adapter Request](#) activity is used to perform the response.

Figure 1: An Adapter Request-Response Server activity



Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.

Field	Global Var?	Description
Description	No	Short description of the activity.
Adapter Service	No	The preconfigured adapter request-response invocation service. You must specify the adapter service when configuring the adapter. Use the Browse button to locate the appropriate service for this activity.
Operation	No	This field contains a drop down list of any operations defined by the schema used in the selected Adapter Service. Select the operation that this process definition implements.

Transport

For more information about the Transport tab, see [Transport Tab](#).

Misc

The Misc tab has the following fields.

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom ID	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of

Field	Description
	TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output

The output for the activity is the following.

Output Item	Datatype	Description
subject	string	The subject of the message.
replySubject	string	The reply subject of the message.
certifiedSequenceNumber	string	This element is only available when Tibrv Certified is the transport. This element contains the sequence number in which this message was received.
certifiedSender	string	This element is only available when Tibrv Certified is the transport. This element contains the cmname of the message sender.
trackingInfo	complex	The AE tracking info of the sender. This element contains the tracking ID and applInfo sent by the application.
MessageID	string	This element is only available when JMS Topic or JMS Queue is the transport. This is the message ID of the message.
JMSHeaders	complex	This element is only available when JMS Topic or JMS Queue is the transport.

Output Item	Datatype	Description
		This element contains the message header fields for the message. For more information about message header fields, see JMS Properties . Only properties applicable to this type of message are displayed.
OtherProperties	complex	This element is only available when JMS Topic or JMS Queue is the transport and a JMS Application Properties resource is specified on the Transport tab. This element contains the message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
body	complex	The class specified for the client service of the specified adapter configuration.

Adapter Subscriber

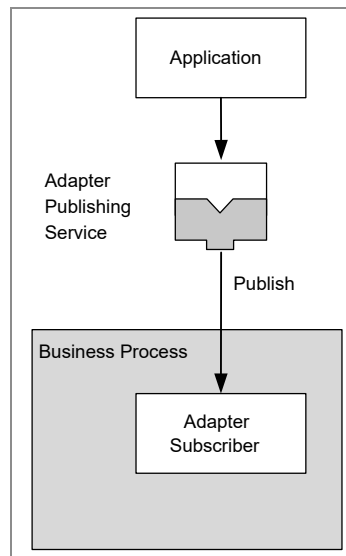
Process Starter



Starts a process based on the receipt of a message from the publication service of the specified adapter.

Publication services are configured during adapter configuration, and the activity uses the information in the adapter configuration to fill in most of the fields of this activity. For more information about creating adapter configurations and creating adapter publication services, see your adapter documentation.

[An Adapter Subscriber activity receiving a message and starting a process](#) illustrates an adapter publishing service publishing a message and the Adapter Subscriber process starter starting a business process.

Figure 2: An Adapter Subscriber activity receiving a message and starting a process

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Adapter Service	No	The preconfigured adapter publishing service. You must specify the adapter service when configuring the adapter. Use the Browse button to locate the appropriate service for this activity.
Custom AESchema	No	Specifies whether you want to supply an output class for this activity.
Custom	No	Only appears when the Custom AESchema field is checked.

Field	Global Var?	Description
AESchema Reference		This field specifies the TIBCO ActiveEnterprise class to use for the body of the output message of this activity.

Transport

For more information about the Transport tab, see [Transport Tab](#).

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value will be executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output

The output for the activity is the following.

Output Item	Datatype	Description
subject	string	The subject of the message.
replySubject	string	The reply subject of the message.
certifiedSequenceNumber	string	This element is only available when Tibrv Certified is the transport. This element contains the sequence number in which this message was received.
certifiedSender	string	This element is only available when Tibrv Certified is the transport. This element contains the cmname of the message sender.
trackingInfo	complex	The AE tracking info of the sender. This element contains the tracking ID and appInfo sent by the application.
MessageID	string	This element is only available when JMS Topic or JMS Queue is the transport. This is the message ID of the message.
JMSHeaders	complex	This element is only available when JMS Topic or JMS Queue is the transport. This element contains the message header fields for the message. For more information about message header fields, see JMS Properties. Only properties applicable to this type of message are displayed.
OtherProperties	complex	This element is only available when JMS Topic or JMS Queue is the transport and a JMS Application Properties resource is specified on the Transport tab. This element contains the message

Output Item	Datatype	Description
		properties For the message. For more information about message properties, see JMS Properties . Only properties applicable to this type of message are displayed.
body	complex	The class specified for the publication service of the specified adapter configuration.

Invoke an Adapter Request-Response Service

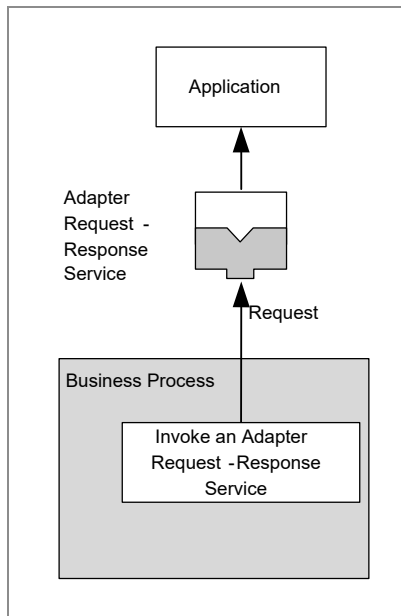
Activity



The **Invoke an Adapter Request-Response Service** activity is used to communicate with an adapter's request-response service. This service invokes an operation with input and output by way of the adapter. Request-response services are configured during adapter configuration, and the activity uses the information in the adapter configuration to fill in most of the fields of this activity. For more information about creating adapter configurations and creating adapter request-response services, see your adapter documentation.

[An Invoke an Adapter Request-Response Service activity sending a request](#) illustrates an **Invoke an Adapter Request-Response Service** activity sending a request to an adapter request-response service.

Figure 3: An Invoke an Adapter Request-Response Service activity sending a request



Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Adapter Service	No	The preconfigured adapter request-response invocation service. You must specify the adapter service when configuring the adapter. Use the Browse button to locate the appropriate service for this activity. You can choose services configured with the RVCMQ transport, but the transport is converted to RVCM when used

Field	Global Var?	Description
		by this activity. This is because the RVCMQ transport does not apply to publishing activities.
Operation	No	This field contains a drop down list of any operations defined by the schema used in the selected Adapter Service. Select the operation that you want to invoke.
One Way	No	This is a read-only field that displays whether the selected operation is one-way.

Transport

For more information about the Transport tab, see [Transport Tab](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
subject	string	The subject to send the request on. The default subject is displayed on the Advanced tab, but you can override the default by specifying a subject here.
requestTimeout	integer	The time to wait (in milliseconds) for the synchronous request-reply operation to complete. An error is returned if the operation does not complete in this time limit.
replySubject	string	The Subject/Destination that is used to send the response when using the RV/JMS transport.

Input Item	Datatype	Description
businessKey	string	This string is appended to the AE tracking info appInfo field. The AE tracking info for messages published is a concatenation of bw + <i><processName></i> + <i><jobID></i> + <i><businessKey></i>. Note: It is a good idea to have all publishers in a process use the same business key. Only the most recent publisher's business key is used in the tracking info for the message.
OtherProperties	complex	This element is only available when JMS Topic or JMS Queue is the transport and a JMS Application Properties resource is specified on the Transport tab. This element contains the message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
body	complex	The input parameters for the specified operation. The input parameters are defined by the operation class for the specified Adapter Service for this activity.

Output

The output for the activity is the following.

Output Item	Datatype	Description
replyDestination	string	A reply message can optionally have a reply destination, presumably so that the two applications can communicate with further messages. This element contains the reply subject (RV) or destination (JMS) set on the reply message, if one is set. If no reply subject or destination is set on the reply message, this element is not populated.

Output Item	Datatype	Description
trackingInfo	string	The AE tracking info. The id element is set to the same value for all message publishers in a process definition. The applInfo element is obtained from the businessKey element on the Input tab.
JMSHeaders	complex	This element is only available when JMS Topic or JMS Queue is the transport. This element contains the message header fields for the message. For more information about message header fields, see JMS Properties. Only properties applicable to this type of message are displayed.
OtherProperties	complex	This element is only available when JMS Topic or JMS Queue is the transport and a JMS Application Properties resource is specified on the Transport tab. This element contains the message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
body	complex	The reply schema for the specified operation.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
TransportException	The transport cannot be created.

Exception	Thrown When...
SerializationException	The outgoing message cannot be serialized.
DeserializationException	The incoming response message cannot be serialized.
ConfirmationException	The message could not be confirmed.
TimeoutException	A timeout occurred.
<i>operationException</i>	The adapter service returned an exception encountered while performing the operation. Any exception class defined on the operation schema can be returned. The potential exceptions are displayed as a choice element for activities after an error transition.

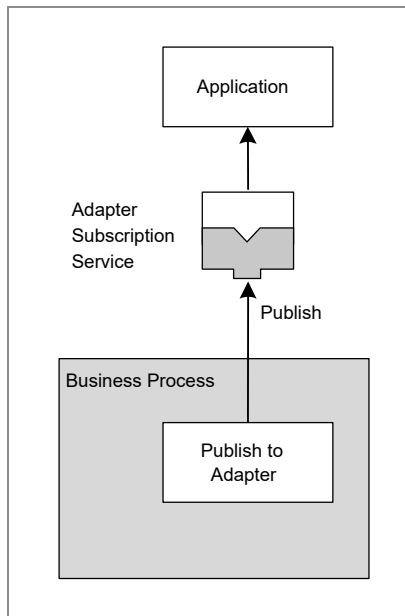
Publish to Adapter

Activity



The **Publish to Adapter** activity publishes a message that can be received by an adapter's subscription service. Subscription services are configured during adapter configuration, and the activity uses the information in the adapter configuration to fill in most of the fields of this activity. For more information about creating adapter configurations and creating adapter subscription services, see your adapter documentation.

A [Publish to Adapter activity sending a message](#) illustrates a **Publish to Adapter** activity sending a message to a preconfigured adapter subscription service.

Figure 4: A Publish to Adapter activity sending a message

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Adapter Service	No	The preconfigured adapter subscription service. You must specify the adapter service when configuring the adapter. Use the Browse button to locate the appropriate service for this activity. You can choose services configured with the RVCMQ transport, but the transport is converted to RVCM when used by this activity. This is because the RVCMQ transport

Field	Global Var?	Description
		does not apply to publishing activities.
RequestReply	No	Specifies that this publish activity is sending a request message and expecting a response message. This is similar to using the Invoke an Adapter Request-Response Service activity, except this activity uses an adapter Subscription service instead of a Request-Response service. When this field is checked, the Request AE Schema and Reply AE Schema fields appear. Also, the replyDestination element appears in the activity output.
Request AE Schema	No	Only appears when the RequestReply field is checked. This field specifies the TIBCO ActiveEnterprise schema to use for the body of the message in the input of this activity.
Reply AE Schema	No	Only appears when the RequestReply field is checked. This field specifies the TIBCO ActiveEnterprise schema to use as the output reply message.
Custom AESchema	No	Specifies whether you want to supply a class for this activity's input.
Custom AESchema Reference	No	Only appears when the Custom AESchema field is checked. This field specifies the TIBCO ActiveEnterprise schema to use for the body of the message in the input of this activity.

Transport

For more information about the Transport tab, see [Transport Tab](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
subject	string	The subject to send the request on. The default subject is displayed on the Advanced tab, but you can override the default by specifying a subject here.
requestTimeout	integer	When publishing in request/reply mode, this element specifies the timeout (in milliseconds) to wait for the reply.
preRegisterListener	string	When using TIBCO Rendezvous Certified Messaging (RVCM), this field specifies the cmname(s) of the RVCM subscriber(s). This allows the publisher to pre-register with any expected listeners. If multiple cmnames are specified, separate each cmname with a comma. Any listeners preregistered on the Transport tab are also preregistered with the listeners specified in this element. For more information about RVCM, see the TIBCO Rendezvous documentation.
replySubject	string	The subject that replies to this message should be sent to.
businessKey	string	This string is appended to the AE tracking info applInfo field. The AE tracking info for messages published is a concatenation of <code>bw + <processName> + <jobID> + <businessKey></code>. The AE tracking info id is set to the same value for all message publishers in a process definition.

Input Item	Datatype	Description
Note: It is a good idea to have all publishers in a process use the same business key. Only the most recent publisher's business key is used in the tracking info for the message.		
OtherProperties	complex	This element is only available when JMS Topic or JMS Queue is the transport and a JMS Application Properties resource is specified on the Transport tab. This element contains the message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
body	complex	The schema definition for the specified adapter service. This can be overridden by specifying a different schema in the Custom AESchema Reference field on the Configuration tab, or by specifying a Request AE Schema when RequestReply is checked on the Configuration tab.

Output

This activity only produces output when the RequestReply field on the Configuration tab is checked. The output for the activity is the following.

Output Item	Datatype	Description
replyDestination	string	A reply message can optionally have a reply destination, presumably so that the two applications can communicate with further messages. This element contains the reply subject (RV) or destination (JMS) set on the reply message, if one is set. If no reply subject or destination is set on the reply message, this

Output Item	Datatype	Description
		element is not populated.
trackingInfo	string	The AE tracking info. The id element is set to the same value for all message publishers in a process definition. The applInfo element is obtained from the businessKey element on the Input tab.
MessageID	string	This element is only available when JMS Topic or JMS Queue is the transport. This is the message ID of the message.
JMSHeaders	complex	This element is only available when JMS Topic or JMS Queue is the transport. This element contains the message header fields for the message. For more information about message header fields, see JMS Properties. Only properties applicable to this type of message are displayed.
OtherProperties	complex	This element is only available when JMS Topic or JMS Queue is the transport and a JMS Application Properties resource is specified on the Transport tab. This element contains the message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
body	complex	The specified reply schema.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
TransportException	The transport cannot be created.
SerializationException	The outgoing message cannot be serialized.
TimeoutException	The specified timeout has been reached for the request/reply operation.

Respond to Adapter Request

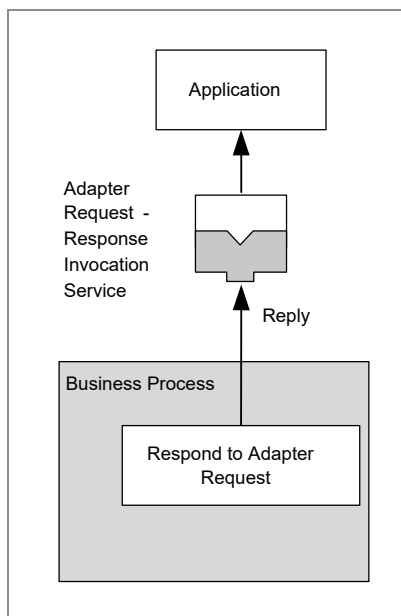
Activity



The **Respond to Adapter Request** activity is used to send a response to an adapter for a previously received request. For example, you may have a [Wait for Adapter Request](#) activity in a process definition. The incoming adapter request may require a response from your process. The **Respond to Adapter Request** is used to send that response.

[A Respond to Adapter Request sending a response](#) illustrates a **Respond to Adapter Request** activity sending a response to a previously received request from an adapter.

Figure 5: A Respond to Adapter Request sending a response



Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Reply For	No	drop down list of activities that this activity can send a response for.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
businessKey	string	This string is appended to the AE tracking info applInfo field. The AE tracking info for messages published is a concatenation of <code>bw + <processName> + <jobID> + <businessKey></code>. Note: It is a good idea to have all publishers in a process use the same business key. Only the most recent publisher's business key is used in the tracking info for the message.
body	complex	The schema definition for the response. The structure of this element is retrieved from the adapter service of the activity that this activity is publishing a response for.

Output

This activity produces no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, *see TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
TransportException	The transport cannot be created.
SerializationException	The outgoing reply cannot be serialized.
ReplyException	The reply could not be sent.

Send Exception to Adapter Request

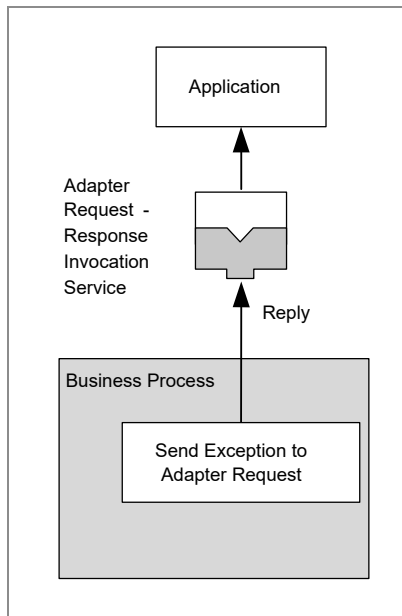
Activity



The **Send Exception to Adapter Request** activity is used to send an exception to an adapter for a previously received request. For example, you may have a [Adapter Request-Response Server](#) process starter in a process definition. An error may occur while attempting to perform the requested operation. The **Send Exception to Adapter Request** activity allows you to return an exception to the adapter service with information about the exception.

A [Send Exception to Adapter Request sending an exception](#) illustrates a **Send Exception to Adapter Request** activity sending an exception to a previously received request from an adapter.

Figure 6: A Send Exception to Adapter Request sending an exception



Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Send Exception To	No	drop down list of activities that this activity can send an exception to.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
businessKey	string	This string is appended to the AE tracking info applInfo field. The AE tracking info for messages published is a concatenation of <code>bw + <processName> + <jobID> + <businessKey></code>. Note: It is a good idea to have all publishers in a process use the same business key. Only the most recent publisher's business key is used in the tracking info for the message.
body	complex	The schema definition for the exception. The schema used by the adapter service must define exception schemas that can be sent.

Output

This activity produces no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
TransportException	The transport cannot be created.
SerializationException	The outgoing reply cannot be serialized.
ReplyException	The exception could not be sent.

Wait for Adapter Message

Activity



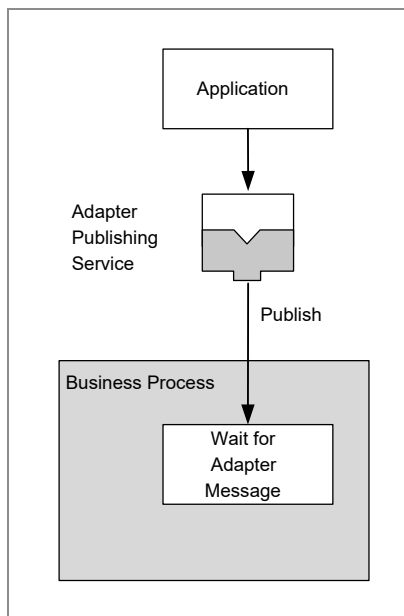
Waits for the receipt of a message from the publication service of the specified adapter.

Publication services are configured during adapter configuration, and the activity uses the information in the adapter configuration to fill in most of the fields of this activity. For more information about creating adapter configurations and creating adapter publication services, see your adapter documentation .

For more information about running the "Wait for" activities in a multi-engine mode, see [Scalability With Incoming Events](#) in *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

A [Publish to Adapter activity sending a message](#) illustrates an adapter publishing service publishing a message and the **Wait for Adapter Message** activity receiving the message in a business process.

Figure 7: A Publish to Adapter activity sending a message



Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Adapter Service	No	The preconfigured adapter publishing service. You must specify the adapter service when configuring the adapter. Use the Browse button to locate the appropriate service for this activity.
Custom AESchema	No	Specifies whether you want to supply a class for this activity's output.
Custom AESchema Reference	No	Only appears when the Custom AESchema field is checked. This field specifies the TIBCO ActiveEnterprise schema to use for the body of the message in the output of this activity.

Transport

For more information about the Transport tab, see [Transport Tab](#).

Event

The Event tab has the following fields.

Field	Description
Candidate Event Key	Expression built from the data of the incoming message. This expression should evaluate to a string and it is compared to the "key" field of the activity's input. If the Candidate Event Key and the activity's key match, then the process accepts the incoming message. For example, you may have a Publish to Adapter activity that sends a

Field	Description
	message with a particular ID. You are expecting a reply message that contains that same ID so that you can determine the message is a response to your sent message. You would specify the field of the incoming message that contains your ID in the Candidate Event Key. You would then use the message ID of the message you sent earlier in the process as the "key" field in the input. This expression is specified in XPath, and only data from the incoming event is available for use in this XPath expression. For more information about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Event Timeout (msec)	A message may arrive before this activity is executed. This field specifies the amount of time (in milliseconds) a message waits if it is received before this activity is executed. If the event timeout expires, an error is logged and the event is discarded. If no value is specified in this field, the message waits indefinitely. If zero is specified, the event is discarded immediately, unless this activity has already been reached.

i Note: When building an expression in the Candidate Event Key field, only data from the incoming event is available. This is because you want to place an expression containing incoming event data in the Candidate Event Key field. When the results of this expression match the results of the expression in the "key" item on the Input tab, the **Wait For Adapter Message** activity proceeds.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
key	string	The value that is compared to the Candidate Event Key field of the Event tab to determine if this message applies to this process. For an example of using the Candidate Event Key field with the key input item, see Event .
processTimeout	integer	The time (in milliseconds) to wait for the incoming message. An error is returned if the message is not received in this time limit.

Output

The output for the activity is the following.

Output Item	Datatype	Description
subject	string	The subject of the message.
replySubject	string	The reply subject of the message.
certifiedSequenceNumber	string	This element is only available when Tibrv Certified is the transport. This element contains the sequence number in which this message was received.
certifiedSender	string	This element is only available when Tibrv Certified is the transport. This element contains the cmname of the message sender.
trackingInfo	complex	The AE tracking info of the sender. This element contains the tracking ID and applInfo sent by the application.
body	complex	The class specified for the publication

Output Item	Datatype	Description
		service of the specified adapter configuration or in the Custom AESchema Reference field on the Configuration tab.

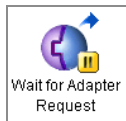
Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ActivityTimeoutException	The specified processTimeout has been reached.
TransportException	The transport cannot be created.
DeserializationException	The outgoing reply cannot be serialized.
ConfirmationException	The message could not be confirmed.

Wait for Adapter Request

Activity



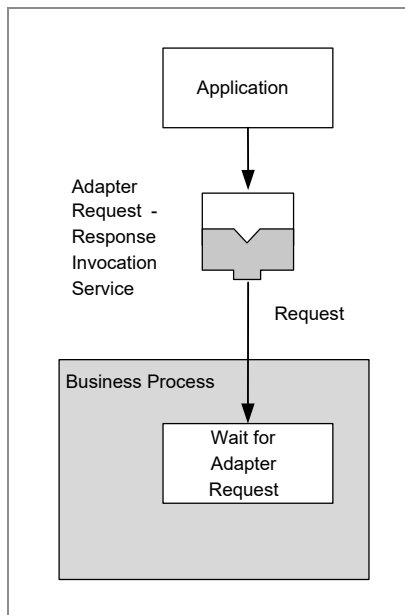
Waits for the receipt of a request from an adapter request-response invocation service. The adapter sends a request using a preconfigured service, and the process definition acts as the implementation of the request. The process sends any required responses back to the adapter's service by placing the [Respond to Adapter Request](#) activity at a later point in the process definition.

Adapter request-response invocation services are configured during adapter configuration, and the activity uses the information in the adapter configuration to fill in most of the fields of this activity. For more information about creating adapter configurations and creating adapter services, see your adapter documentation .

For more information about running the "Wait for" activities in a multi-engine mode, see [Scalability With Incoming Events](#) in *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

A [Wait for Adapter Request activity](#) illustrates a **Wait for Adapter Request** activity waiting for a request from an adapter request-response invocation service. At a later point in the business process, a [Respond to Adapter Request](#) activity sends the response.

Figure 8: A Wait for Adapter Request activity



Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.

Field	Global Var?	Description
Adapter Service	No	The preconfigured adapter request-response invocation service. You must specify the adapter service when configuring the adapter. Use the Browse button to locate the appropriate service for this activity.
Operation	No	This field contains a drop down list of any operations defined by the schema used in the selected Adapter Service. Select the operation that this process definition is waiting for.

Transport

For more information about the Transport tab, see [Transport Tab](#).

Event

The Event tab has the following fields.

Field	Description
Candidate Event Key	Expression built from the data of the incoming message. This expression should evaluate to a string and it is compared to the "key" field of the activity's input. If the Candidate Event Key and the activity's key match, then the process accepts the incoming message. For example, you may have a Invoke an Adapter Request-Response activity that sends a request with a particular ID. You are expecting a response message that contains that same ID so that you can determine the message is a response to your sent message. You would specify the field of the incoming reply that contains your ID in the Candidate Event Key. You would then use the message ID of the request you sent earlier in the process as the "key" field in the input.

Field	Description
	This expression is specified in XPath, and only data from the incoming event is available for use in this XPath expression. For more information about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i> .
Event Timeout (msec)	A message may arrive before this activity is executed. This field specifies the amount of time (in milliseconds) a message waits if it is received before this activity is executed. If the event timeout expires, an error is logged and the event is discarded. If no value is specified in this field, the message waits indefinitely. If zero is specified, the event is discarded immediately, unless this activity has already been reached.

**Note:**

When building an expression in the Candidate Event Key field, only data from the incoming event is available. This is because you want to place an expression containing incoming event data in the Candidate Event Key field. When the results of this expression match the results of the expression in the "key" item on the Input tab, the **Wait For Adapter Request** activity proceeds.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
key	string	The value that is compared to the Candidate Event Key field of the Event tab to determine if this message applies to this process. For an example of using the Candidate Event Key field with the key input item, see Event .

Input Item	Datatype	Description
processTimeout	integer	The time (in milliseconds) to wait for the incoming message. An error is returned if the message is not received in this time limit.

Output

The output for the activity is the following.

Output Item	Datatype	Description
subject	string	The subject of the message.
certifiedSequenceNumber	string	This element is only available when Tibrv Certified is the transport. This element contains the sequence number in which this message was received.
certifiedSender	string	This element is only available when Tibrv Certified is the transport. This element contains the cmname of the message sender.
trackingInfo	complex	The AE tracking info of the sender. This element contains the tracking ID and applInfo sent by the application.
MessageID	string	This element is only available when JMS Topic or JMS Queue is the transport. This is the message ID of the message.
JMSHeaders	complex	This element is only available when JMS Topic or JMS Queue is the transport. This element contains the message header fields for the message. For more information

Output Item	Datatype	Description
		about message header fields, see JMS Properties . Only properties applicable to this type of message are displayed.
OtherProperties	complex	This element is only available when JMS Topic or JMS Queue is the transport and a JMS Application Properties resource is specified on the Transport tab. This element contains the message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
body	complex	The class specified for the client service of the specified adapter configuration or the schema specified in the Custom AESchema Reference field on the Configuration tab.

File Palette

The **File** palette is used to read, write, delete, or create files. This palette also has a process starter that allows you to poll for files and start a process based on the presence of a file.

Copy File

Activity



The **Copy File** activity allows you to copy files and directories to a new location.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Overwrite	No	Overwrite existing file or directory with the same name, if one exists. The activity raises an exception if the specified file or directory exists and this field is unchecked.
Create Destination Directory	No	Create any directories in the destination path that do not already exist. This activity raises an exception if any of the directories on the specified destination path do not exist and this field is

Field	Global Var?	Description
		unchecked.
Include Sub-Directories	No	When the source to copy is a directory, include all sub-directories in the source directory.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
fromFileName	string	The path and name of the file or directory to copy. For directories, you must specify an absolute path. You may use wildcards when specifying files within a directory. For example, to copy a directory, specify c:\myDirectory. To copy all text files in a directory, specify c:\myDirectory*.txt.
toFileName	string	The destination for the copy operation. This must be an absolute path; wildcards cannot be used in this element.

Output

This activity produces no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO*

ActiveMatrix BusinessWorks™ Error Codes.

Exception	Thrown When...
FileNotFoundException	The source file is not found or the destination is not found and the Create Destination Directory field is not checked on the Configuration tab.
IllegalCopyException	The destination exists and its type is different than the type of the specified source. For example, in the activity input, the fromFileName is a directory and the toFileName is a file.
FileAlreadyExistsException	The destination you are attempting to copy to already exists and the Overwrite field is not checked on the Configuration tab.
FileIOException	There is an I/O error. For example, the file or directory already exists and is write protected.

Create File

Activity



The **Create File** activity is a synchronous activity that creates a new file or directory with the specified name.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.

Field	Global Var?	Description
Description	No	Short description of the activity.
Overwrite	No	Overwrite existing file with the same name, if one exists. The activity raises an exception if the specified file exists and this field is unchecked.
Is a Directory	No	When checked, the activity creates a directory instead of a file.
Create Non-Existing Directories	No	When checked, the activity creates all directories in the specified path, if they do not already exist. If this field is unchecked and there are one or more directories in the specified path that do not exist, an exception is raised.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
fileName	string	The path and name of the file to create. If the Is a Directory field is checked on the Configuration tab, this specifies the name of the directory to create.

Output

The output for the activity is the following.

Output Item	Datatype	Description
fileInfo	complex	Element containing the fileName, location, type, readProtected, writeProtected, and size data.
fullName	string	The name of the file or directory, including the path information.
fileName	string	The name of the file or directory without the path information.
location	string	The path to the file or directory.
configuredFileName	string	This element is optional and it is not populated by this activity.
type	string	The file type.
readProtected	boolean	Whether the file or directory is protected from reading.
writeProtected	boolean	Whether the file or directory is protected from writing.
size	integer	Size of the file (in bytes).
lastModified	string	Time stamp indicating when the file was last modified.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
FileAlreadyExistsException	The file or directory you are attempting to create already exists and the Overwrite field is not checked on the Configuration tab.
FileIOException	There is an I/O error. For example, the file or directory already exists and is write protected.

File Poller

Process Starter



The File Poller process starter polls for files or directories with the given name and starts a process when the specified change (creation, modification, deletion) is detected.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
File Name	Yes	The path and name of the file or directory to monitor. Use the Browse button to locate an existing file. You can also use wildcards to monitor a directory for files that match the given specification. For example, C:\files*.log would match any change to a file in the files directory with an extension of .log.
Polling	Yes	Polling interval (in seconds) to check for the specified file.

Field	Global Var?	Description
Interval (sec)		
Include Existing Files	No	Check for the file among the files that existed before the process engine started running. When this field is checked, if an existing file matches the specification in the File Name field when a process engine starts, then a new process instance is created. If this field is not checked, any existing files that match the specification in the File Name field are ignored until there is a change in the file.
Exclude File Content	No	Specifies not to load the data from the file into this activity's output. If this check box is selected, the contents of the file are not available to subsequent activities in the process definition.
Content as	Yes	The type of content in the file. Can be either text or binary.
Encoding	Yes	The character encoding for the file (when text is specified in the Content as field).

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Poll for Create Events	No	Checking this field starts a new process when a file or directory matching the specification in the File Name field is created. If this field is unchecked, newly created files/directories are ignored.
Poll for Modify	No	Checking this field starts a new process when a file or directory

Field	Global Var?	Description
Events		matching the specification in the File Name field is modified. If this field is unchecked, modifications to files/directories are ignored.
Poll for Delete Events	No	Checking this field starts a new process when a file or directory matching the specification in the File Name field is deleted. If this field is unchecked, deletions of files/directories are ignored.
Include Sub-Directories	No	When this field is checked, events that occur in subdirectories of the specified directory are also monitored. If this field is unchecked, only the directory is monitored, any subdirectories are ignored.
Mode	No	Specifies the type of polling to perform. Specify one of the following: • only-directories – only tracks the specified changes for directories. • only-files – only tracks the specified changes for files. • files-and-directories – tracks the specified changes for both files and directories.
Sort By	No	Specifies the order in which files should be processed, if more than one file is detected. You can sort by file name, modification date, or specify no sorting is to be performed. Files are processed as the polling interval occurs. Therefore, the Sort By field applies to all files that have changed since the last polling interval. The Sort By field can be very useful if you want to process files in a specific order when the process engine starts and the Include Existing Files field is checked (then all existing files are processed in the specified order).

Field	Global Var?	Description
Note: The File Poller starts process instances based on the specified Sort By order, but the process instances may not complete in the same order. If you want to guarantee that processes start and complete in a given order, you must specify Max Jobs = 1 and enable the Activation Limit field when you deploy your project. For more information about deploying projects, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i>.		
Sort Order	No	Specifies whether the order specified in the Sort By field is ascending or descending. This field is only applicable if File Name or Date Modified is selected in the Sort By field.

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output

The output for the activity is the following.

Output Item	Datatype	Description
action	string	The event that occurred to trigger the File Poller activity. The possible values are: create, remove, or modify.
timeOccured	integer	The time that the File Poller activity detected the change in the file. The time is expressed as the amount of time, in milliseconds since midnight, January 1, 1970, UTC.
fileInfo	complex	Element containing the fileName, location, type, readProtected, writeProtected, and size data.
fullName	string	The name of the file, including the path information.
fileName	string	The name of the file without the path information.
location	string	The path to the file.
configuredFileName	string	The name of the file or directory this activity is configured to operate on.
type	string	The file type.
readProtected	boolean	Whether the file is read-protected.
writeProtected	boolean	Whether the file is write-protected.
size	integer	Size of the file in bytes.
lastModified	string	Time stamp indicating when the file was last modified.

Output Item	Datatype	Description
fileContent	complex	Element containing the textContent and binaryContent data.
textContent	string	Contents of the file (text files), if the Exclude File Contents field is not checked on the Configuration tab.
binaryContent	binary	Contents of the file (binary files), if the Exclude File Contents field is not checked on the Configuration tab.
encoding	string	The character encoding for the file.

List Files

Activity



The **List Files** activity returns information about files or directories, or a listing of all the files in the specified directory.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Mode	No	The type of listing you would like to retrieve. The following are the choices for this field:

Field	Global Var?	Description
		• directories-only • files-only • files-and-directories

Input

The input for the activity is the following.

Input Item	Datatype	Description
fileName	string	The path and name of the directory with files or subdirectories to list. If you supply a file or directory name, information about that file or directory is returned. You can use wildcards to list files or directories that match the given specification. For example, C:\files*.log lists any file in the files directory that have a .log extension. C:\files\mylogs*. * lists all files in the directory mylogs.

Output

The output for the activity is the following.

Output Item	Datatype	Description
fileInfo	complex	Element containing the fileName, location, type, readProtected, writeProtected, and size data. This is a repeating element. Each item in the element is a file or directory contained in the specified directory that matches the given criteria.

Output Item	Datatype	Description
fullName	string	The name of the file, including the path information.
fileName	string	The name of the file without the path information.
configuredFileName	string	The name of the file or directory this activity is configured to operate on.
location	string	The path to the file.
type	string	The file type.
readProtected	boolean	Whether the file is read-protected.
writeProtected	boolean	Whether the file is write-protected.
size	integer	Size of the file in bytes.
lastModified	string	Time stamp indicating when the file was last modified. You can compare values in this element using the <code>compare-dateTime()</code> XPath function. For example, <pre>compare-dateTime(\$List-Files/ ListFilesActivityOutput/files/ fileInfo[fileName="myFile.txt"]/ lastModified, current-dateTime())</pre>

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
FileNotFoundException	The specified path is not found.

Read File

Activity



The **Read File** activity is used to read a file and place its contents into the activity's output.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Exclude File Content	No	Specifies not to load the data from the file into this activity's output. If this field is checked, the contents of the file are not available to subsequent activities in the process definition.
Read as	Yes	The type of content in the file. Can be either text or binary.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
fileName	string	The path and name of the file to read.
encoding	string	The character encoding for text files. This element is only available when the Read as field on the Configuration tab is set to text. Encoding names supported by Java can be specified in this element. If this element is not specified, the default encoding of the Java Virtual Machine used by the process engine is used.

Output

The output for the activity is the following.

Output Item	Datatype	Description
fileInfo	complex	Element containing the fileName, location, type, readProtected, writeProtected, and size data.
fullName	string	The name of the file, including the path information.
fileName	string	The name of the file without the path information.
location	string	The path to the file.
configuredFileName	string	This element is optional and it is not populated by this activity.
type	string	The file type.
readProtected	boolean	Whether the file is read-protected.
writeProtected	boolean	Whether the file is write-protected.

Output Item	Datatype	Description
size	integer	Size of the file in bytes.
lastModified	string	Time stamp indicating when the file was last modified.
fileContent	complex	Element containing the textContent and binaryContent data.
textContent	string	Contents of the file (text files), if the Exclude File Content field is not checked on the Configuration tab.
binaryContent	binary	Contents of the file (binary files), if the Exclude File Content field is not checked on the Configuration tab.
encoding	string	The character encoding for the file.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
FileNotFoundException	The file does not exist.
UnsupportedEncodingException	The text file's encoding is not valid and the content of the file is read into process data.
FileIOException	An I/O exception occurred when trying to read the file.

Remove File

Activity



The **Remove File** activity removes the specified file. This activity can also remove empty directories. If a directory that is not empty is specified, an exception is thrown.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
fileName	string	The path and name of the file or empty directory to remove.

Output

The output for the activity is the following.

Output Item	Datatype	Description
fileInfo	complex	Element containing the fileName, location, type, readProtected, writeProtected, and size data.
fullName	string	The name of the file (or empty directory), including the path information.
fileName	string	The name of the file or empty directory without the path information.
location	string	The path to the file or empty directory.
configuredFileName	string	This element is not populated by this activity.
type	string	The file type.
readProtected	boolean	Whether the file (or empty directory) is read-protected.
writeProtected	boolean	Whether the file (or empty directory) is write-protected.
size	integer	Size of the file in bytes.
lastModified	string	Time stamp indicating when the file was last modified.

Error Output

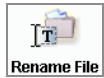
The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
FileNotFoundException	The file does not exist.

Exception	Thrown When...
FileIOException	An I/O exception occurred when trying to remove the file or empty directory.

Rename File

Activity



The **Rename File** activity is used to rename or move files. This activity can also rename directories, but you cannot use this activity to move a directory to a new location.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Overwrite	Yes	Overwrite existing file with the same name, if one exists when renaming or moving. The activity raises an exception if the specified file or directory exists and this field is unchecked.
Create Destination Directory	Yes	Creates all directories in the specified destination path where the file is moved to, if the directories do not already exist. If this field is unchecked and one or more directories in the specified path do not exist, an exception is raised.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
fromFileName	string	The path and name of the file to rename or move, or the path and name of the directory to rename. The value in this element must be an absolute path.
toFileName	string	The new name and location of the file or directory. Files can be moved to a new location, but directories must remain in the same location. The value in this element must be an absolute path.

Output

The output for the activity is the following.

Output Item	Datatype	Description
fileInfo	complex	Element containing the fileName, location, type, readProtected, writeProtected, and size data.
fullName	string	The name of the file (or directory), including the path information.
fileName	string	The name of the file or directory without the path information.
location	string	The path to the file or directory.
configuredFileName	string	This element is not populated by this activity.

Output Item	Datatype	Description
type	string	The file type.
readProtected	boolean	Whether the file (or directory) is read-protected.
writeProtected	boolean	Whether the file (or directory) is write-protected.
size	integer	Size of the file in bytes.
lastModified	string	Time stamp indicating when the file was last modified.

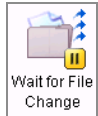
Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
FileNotFoundException	The file does not exist.
IllegalRenameException	The destination file exists and its type is different than the type of the source file.
FileAlreadyExistsException	The file already exists.
FileIOException	An I/O exception occurred when trying to rename the file or directory. This exception is also thrown if an attempt is made to move a directory.

Wait for File Change

Activity



The **Wait for File Change** activity waits for a file creation, modification, or deletion event to occur during process execution. When this activity is executed, the process instance suspends and waits for the specified change to occur before resuming.

For more information about running the "Wait for" activities in a multi-engine mode, see [Scalability With Incoming Events](#) in *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
File Name	Yes	The path and name of the file to poll for. You can use the Browse button to locate an existing file. You can also use wildcards to monitor a directory for files that match the given specification. For example, C:\files*.log would match any change to a file in the files directory with an extension of .log.
Polling Interval (sec)	Yes	Polling interval (in seconds) to check for changes in the specified file.
Include Existing Files	Yes	Check for the file among the files that existed before the process engine started running. When this field is checked, if an existing file matches the specification in the File Name field when a process engine starts, then the Wait for File Change activity accepts that file as changed and proceeds to the next activity.

Field	Global Var?	Description
		If this field is not checked, any existing files that match the specification in the File Name field are ignored until there is some change in the file.
Exclude File Content	Yes	Specifies not to load the data from the file into this activity's output. If this field is checked, the contents of the file are not available to subsequent activities in the process definition.
Content as	Yes	The type of content in the file. Can be either text or binary.
Encoding	Yes	The character encoding for the file.

Event

The Event tab has the following fields.

Field	Description
Candidate Event Key	Expression built from the data of the incoming file change event. This expression should evaluate to a string and it is compared to the "key" field of the activity's input. If the Candidate Event Key and the activity's key match, then the process accepts the incoming file change event. For example, you may have created a file earlier in the process and you may want to wait for an external application to modify the file. When the external application modifies the file, it appends the word "Changed" to the filename to signify the file is ready for processing. You should specify the \$fileInfo/fileName for the Candidate Event Key field, and you should specify concat("\$CreateFile/fileName", "Changed") for the "key" field on the Input tab. \$CreateFile/fileName is the name of the file created earlier in the process. This expression is specified in XPath, and only data from the incoming event is available for use in this XPath expression. For more information about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.

Field	Description
Event Timeout (msec)	A file change may occur before this activity is executed. This field specifies the amount of time (in milliseconds) to wait if the file change occurs before this activity is executed in the process instance. If the event timeout expires, an error is logged and the event is discarded. If no value is specified in this field, the message waits indefinitely. If zero is specified, the event is discarded immediately, unless this activity has already been executed.

i Note: When building an expression in the Candidate Event Key field, only data from the incoming event is available. This is because you want to place an expression containing incoming event data in the Candidate Event Key field. When the results of this expression match the results of the expression in the "key" item on the Input tab, the **Wait For File Change** activity proceeds.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
key	string	The value that is compared to the Candidate Event Key field of the Event tab to determine if this message applies to this process. For an example of using the Candidate Event Key field with the key input item, see Event .
processTimeout	integer	The time (in milliseconds) to wait for the incoming file change event. An error is returned if the event is not received in this time limit.

Output

The output for the activity is the following.

Output Item	Datatype	Description
action	string	The change in the file that occurred. The action is either the create, remove, or modify.
timeOccured	integer	The time that the Wait for File Change activity detected the change in the file. The time is expressed as the amount of time, in milliseconds since midnight, January 1, 1970, UTC.
fileInfo	complex	Element containing the fileName, location, type, readProtected, writeProtected, and size data.
fullName	string	The name of the file, including the path information.
fileName	string	The name of the file without the path information.
location	string	The path to the file.
configuredFileName	string	This element is not populated by this activity.
type	string	The file type.
readProtected	boolean	Whether the file is read-protected.
writeProtected	boolean	Whether the file is write-protected.
size	integer	Size of the file in bytes.
lastModified	string	Time stamp indicating when the file was last modified.
fileContent	complex	Element containing the textContent and

Output Item	Datatype	Description
		binaryContent data.
textContent	string	Contents of the file (text files), if the Exclude File Contents field was not checked on the Configuration tab.
binaryContent	binary	Contents of the file (binary files), if the Exclude File Contents field was not checked on the Configuration tab.
encoding	string	The character encoding for the file.

Write File

Activity



The **Write File** activity writes content to the specified file.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Append	Yes	Appends contents to an existing file. If unchecked, the file is overwritten.
Write as	Yes	Specifies the type of file content to write: text or binary.

Field	Global Var?	Description
Create Non-Existing Directories	Yes	When checked, the activity creates all directories in the specified path, if they do not already exist. If this field is unchecked and there are one or more directories in the specified path that do not exist, an exception is raised.
Compress	No	Specifies whether to compress the output file using GZip format. Specify None for no compression or GZip for a compressed output file. When you specify GZip for this field, you must rename the file to use the .gz suffix and use gunzip to uncompress the file.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
fileName	string	The path and name of the file. Wildcards are not permitted in this field.
textContent	string	Contents of the file (text files).
binaryContent	binary	Contents of the file (binary files).
addLineSeparator	boolean	Specifies whether to add a carriage return after each input line.
encoding	string	The character encoding for text files. This element is only available when the Write as field on the Configuration tab is set to text. Encoding names supported by Java can be specified in this element. If

Input Item	Datatype	Description
		this element is not specified, the default encoding of the Java Virtual Machine used by the process engine is used.

Output

The output for the activity is the following.

Output Item	Datatype	Description
fileInfo	complex	Element containing the fileName, location, type, readProtected, writeProtected, and size data.
fullName	string	The name of the file, including the path information.
fileName	string	The name of the file without the path information.
location	string	The path to the file.
configuredFileName	string	This element is not populated by this activity.
type	string	The file type.
readProtected	boolean	Whether the file is read-protected.
writeProtected	boolean	Whether the file is write-protected.
size	integer	Size of the file in bytes.
lastModified	string	Time stamp indicating when the file was last modified.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
FileNotFoundException	The file does not exist.
UnsupportedEncodingException	The encoding is not valid.
FileIOException	An I/O exception occurred when trying to write to the file.

FTP Palette

The FTP palette is used to issue FTP commands.

FTP Change Default Directory

Activity



The FTP Change Default Directory activity changes the current default directory on the remote machine to the specified directory path.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
FTP Connection	No	Path to the shared configuration resource containing the user information such as host name, user name, and password. For more information, see FTP Connection .
Quit (post-command)	No	When checked, this field specifies that the FTP connection should be closed after completing this command. If this field is not checked, the FTP connection remains open so that all FTP activities that specify the same FTP session (that is, the same host, port, username, and password) share the FTP session until the connection is closed.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Host	string	The name of the FTP server. This overrides the value specified in the FTP Connection resource.
Port	integer	Port number for incoming FTP requests on the FTP server. This overrides the value specified in the FTP Connection resource.
UserName	string	Username for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Password	string	Password for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
NewDefaultDir	string	Path to the new default directory on the FTP server.
Timeout	integer	The amount of time to wait for the FTP server to respond. This timeout is used when the FTP server places the FTP connection used by this activity into a waiting state. This overrides the value specified in the FTP Connection resource. For more information, see FTP Connection .

Output

This activity produces no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ConnectionErrorException	An error occurred when connecting to the FTP server.
CommandExecutionException	An error occurred when executing the FTP command.

FTP Connection

Shared Configuration



The FTP Connection shared configuration resource describes a connection to an FTP server. FTP connections are used when configuring activities in the FTP palette.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the resource.
Description	Short description of the shared resource.
SSL	Specifies to use the HTTPS (secure socket layer or SSL) protocol for the FTP connection. This protocol authenticates the FTP client to the server. Enabling this field allows you to specify SSL parameters with the Configure SSL button (described in Configure SSL Button).

Field	Description
Host	Host name or IP address of the FTP server.
Port	Port number of the FTP server. For connections inside of a firewall, 21 is the default when no value is specified in this field.
User Name	User name to use when logging into the FTP server.
Password	Password to use when logging into the FTP server.
Mode	Specifies whether to use Active or Passive mode to connect to the FTP server.
Connection Timeout (msec)	The timeout specified in this field (in milliseconds) applies to the following situations: • Making the control socket connection to the FTP Server. • Making the data socket connection to the FTP server. (passive mode) • Waiting for the data socket connection from the FTP server. (active mode) • Reading a reply from control socket after sending a FTP control command. • Reading data from data socket. If the specified time is reached when attempting any of these operations, the activity using this FTP connection terminates without error and takes any matching success transitions to the next activity in the process definition.
Remote Filename Encoding	A drop down list of values for specifying the character encoding of file names supplied to FTP commands using this connection. The encoding specified in this field is used when sending path and file names supplied to FTP commands that use this FTP Connection resource. Set this field when the FTP server has a different system encoding than the machine where the process engine is running.

Field	Description
Firewall	Specifies whether the FTP host resides outside of a firewall. If this field is checked, you must supply the Proxy Host and Proxy Port fields with the name and the port of the proxy server used to access sites outside of a firewall. You can optionally specify the User Name and Password for the proxy server, if necessary.
Proxy Host	Used when the remote FTP server is outside of a firewall. Specifies the host name or IP address of the firewall server.
Proxy Port	Used when the remote FTP server is outside of a firewall. Specifies the port number of the firewall server.
Proxy User Name	Used when the remote FTP server is outside of a firewall. Specifies the user name to use to connect to the proxy server, if necessary.
Proxy Password	Used when the remote FTP server is outside of a firewall. Specifies the password to use for the specified user name when connecting to the proxy server, if necessary.

Configure SSL Button

The **Configure SSL** button allows you to specify the SSL parameters for the HTTP connection. The following are the fields in the SSL Configuration for HTTPS Connections dialog:

Field	Description
Trusted Certificates Folder	Specifies a folder in the project containing one or more certificates from trusted certificate authorities. This folder is checked when a client connects to ensure that the server's certificate is from a trusted certificate authority. This prevents connections from rogue servers.
Identity	An Identity resource that contains the HTTP server's digital certificate and private key. For more information, see <i>TIBCO Designer™ Palette Reference</i> .

Field	Description
Verify Host Name	Specifies to check that the host name of the FTP server against the host name listed in the server's digital certificate. This provides additional verification that the host name you believe you are connecting to is in fact the desired host. If the host name specified in the Host field on the Configuration tab is not an exact match to the host name specified in the server's digital certificate, the connection is refused. Note: If you specify an equivalent hostname (for example, an IP address) in the Host field, but the name is not an exact match of the hostname in the host's digital certificate, the connection is refused.
Strong Cipher Suites Only	When checked, this field specifies that the minimum strength of the cipher suites used can be specified with the <code>bw.plugin.security.strongcipher.minstrength</code> custom engine property. For more information about this property, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i>. The default value of the property disables cipher suites with an effective key length below 128 bits. When this field is unchecked, only cipher suites with an effective key length of up to 128 bits can be used.

Test Connection Button

When the **Test Connection** button is clicked, ActiveMatrix BusinessWorks attempts to connect to the specified remote FTP server on the specified port using the specified username and password. You can use this button to determine if the specified connection configuration is correct.

Available Commands Button

The **Available Commands** button displays a dialog with details about which FTP commands are available on the specified remote server. FTP servers on different operating systems and from different vendors support slightly different commands. You can use this button to determine which commands are available on the remote FTP server before attempting to execute them with the [FTP Quote](#) activity.

FTP Delete File

Activity



The **FTP Delete File** activity issues an FTP delete or mdelete command to remove one or more files from the remote server.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
FTP Connection	No	Path to the shared configuration resource containing the user information such as host name, user name, and password. For more information, see FTP Connection .
Quit (post-command)	No	When checked, this field specifies that the FTP connection should be closed after completing this command. If this field is not checked, the FTP connection remains open so that all FTP activities that specify the same FTP session (that is, the same host, port, username, and password) share the FTP session until the connection is closed.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Host	string	The name of the FTP server. This overrides the value specified in the FTP Connection resource.
Port	integer	Port number for incoming FTP requests on the FTP server. This overrides the value specified in the FTP Connection resource.
UserName	string	Username for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Password	string	Password for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
RemoteDirectory	string	Path to the directory containing the file(s) to be deleted. This element is concatenated with the RemoteFilename element to provide the complete path to the file(s) to be deleted. This element is optional.
RemoteFilename	string	The path and name of the file to delete on the remote server. Wildcards (? for any one character, * for one or more characters) are permitted in this field. You can specify an absolute or relative path for the remote file name. Relative paths are relative to the path specified in the RemoteDirectory element or the user's home directory, if RemoteDirectory is null.
Timeout	integer	The amount of time to wait for the FTP server to respond. This timeout is used when the FTP server places the FTP connection used by this activity into a waiting state. This overrides the value specified in the FTP Connection resource. For more information, see FTP Connection.

Output

This activity produces no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, *see TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ConnectionErrorException	An error occurred when connecting to the FTP server.
CommandExecutionException	An error occurred when executing the FTP command.

FTP DIR

Activity



The FTP DIR activity provides a listing of files in the specified directory of the FTP server.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.

Field	Global Var?	Description
FTP Connection	No	Path to the shared configuration resource containing the user information such as host name, user name, and password. For more information, see FTP Connection .
Quit (post-command)	No	When checked, this field specifies that the FTP connection should be closed after completing this command. If this field is not checked, the FTP connection remains open so that all FTP activities that specify the same FTP session (that is, the same host, port, username, and password) share the FTP session until the connection is closed.
NLST	Yes	Specifies to use the NLST command instead of the DIR command. The DIR command produces verbose output including file size, modification time, and permissions. NLST produces only a list of filenames.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Host	string	The name of the FTP server. This overrides the value specified in the FTP Connection resource.
Port	integer	Port number for incoming FTP requests on the FTP server. This overrides the value specified in the FTP Connection resource.
UserName	string	Username for logging into the FTP server. This

Input Item	Datatype	Description
		overrides the value specified in the FTP Connection resource.
Password	string	Password for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
DirParameters	string	Command parameters to pass to the DIR or NLST command.
Directory	string	Path to the directory for which you want to list the files. This parameter is optional, and if null, the current remote directory is listed.
Timeout	integer	The amount of time to wait for the FTP server to respond. This timeout is used when the FTP server places the FTP connection used by this activity into a waiting state. This overrides the value specified in the FTP Connection resource. For more information, see FTP Connection .

Output

The output for the activity is the following.

Output Item	Datatype	Description
ItemCount	integer	The number of entries returned from the remote directory listing.
DirectoryItems	string	A repeating element containing the list of items returned.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ConnectionErrorException	An error occurred when connecting to the FTP server.
CommandExecutionException	An error occurred when executing the FTP command.

FTP Get

Activity



The **FTP Get** activity issues an FTP get or mget command to the specified server. The content of the remote files can be placed in the activity's output or written directly to local storage. If you choose to place the contents of a remote file into the activity's output, you can only retrieve one remote file (FTP get command) and the file's content gets stored in memory as part of the activity's output. If you choose to write the contents of the retrieved files to local storage, you can retrieve one or more files (FTP get or mget commands).

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.

Field	Global Var?	Description
FTP Connection	No	Path to the shared configuration resource containing the user information such as host name, user name, and password. For more information, see FTP Connection .
Quit (post-command)	No	When checked, this field specifies that the FTP connection should be closed after completing this command. If this field is not checked, the FTP connection remains open so that all FTP activities that specify the same FTP session (that is, the same host, port, username, and password) share the FTP session until the connection is closed.
Use Process Data	Yes	Specifies to write contents of the file retrieved from the FTP server to the process data. The activity's output includes an element that contains the remote file's contents, and the data can be used by subsequent activities in the process definition. Warning: Do not use this option if you are retrieving multiple files. You can only write the contents of one file to this activity's output. An error is returned if you specify wildcards in the RemoteFilename input element when this field is checked.
Binary	Yes	Specifies whether to retrieve the file content in binary or ASCII mode (checked signifies binary). When ASCII mode is used, differences in line endings between the FTP server and the machine executing the FTP command are automatically handled. Also, ASCII mode implies text data, and that data can be encoded in a character encoding other than ASCII (such as UTF-8 or Latin-1).
Overwrite Existing File	Yes	When this field is checked, and the Use Process Data field is unchecked, any existing local files are overwritten if the FTP command returns files with the same names.

Field	Global Var?	Description
		If this field is unchecked, any attempt to write a file to disk with the same name as an existing file results in an exception.
Maintain 2.0 Compatibility	No	This field is present only for activities migrated from ActiveMatrix BusinessWorks 2.0. If checked, this field ensures the output schema retains the elements available in ActiveMatrix BusinessWorks 2.0.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
RemoteFilename	string	The path and name of the file to retrieve from the remote server. Wildcards (? for any one character, * for one or more characters) are permitted in this field. You can specify an absolute or relative path for the remote file name. Relative paths are relative to the path specified in the RemoteDirectory element or the current directory, if RemoteDirectory is null. Note: Do not specify wildcards for this element if the Use Process Data Configuration field is checked.
LocalFileName	string	This input item is available only if the Use Process Data field on the Configuration tab is unchecked and only one remote file is retrieved. This input item specifies the name and path of the

Input Item	Datatype	Description
		local file where the contents of the retrieved file should be written. If more than one remote file is retrieved, the local files are named to match the remote file names.
Host	string	The name of the FTP server. This overrides the value specified in the FTP Connection resource.
Port	integer	Port number for incoming FTP requests on the FTP server. This overrides the value specified in the FTP Connection resource.
UserName	string	Username for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Password	string	Password for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Encoding	string	When the retrieved file is not binary and the Use Process Data field on the Configuration tab is checked, this specifies the character encoding of the retrieved file. If this is not specified, it is assumed that the encoding of the file is the same encoding of the machine where the FTP command is executed. If the encoding of the local machine and the FTP server are different, specify the encoding used by the FTP server in this element and the activity converts the retrieved file into the local machine's encoding.
Timeout	integer	The amount of time to wait for the FTP server to respond. This timeout is used when the FTP server places the FTP connection used by this activity into a waiting state.

Input Item	Datatype	Description
		This overrides the value specified in the FTP Connection resource. For more information, see FTP Connection .

Output

The output for the activity is the following.

Output Item	Datatype	Description
Data	string	When Use Process Data is checked and only one file is retrieved, this element contains the data retrieved from the remote file.
FilesTransferred	complex	When one or more remote files are retrieved and written to local storage, this element contains a list of the remote filenames that were retrieved. This element contains two elements, Name and NumOfBytes. If an error occurred while retrieving files, the activity output is not available, but the Error Output tab lists the successfully transferred files and the files that failed to be transferred.
Name	integer	The name of the remote file retrieved.
NumOfBytes	integer	The number of bytes contained in the associated file name.
Success	boolean	Only available when Maintain 2.0 compatibility is checked on the Configuration tab. True or false depending upon whether the FTP <code>get</code> command was successfully executed by the FTP server.

Output Item	Datatype	Description
Errormsg	string	Only available when Maintain 2.0 compatibility is checked on the Configuration tab. Any error messages returned from the FTP server.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
GetFilesException	An error occurred during the FTP get operation.
ConnectionErrorException	An error occurred when connecting to the FTP server.
CommandExecutionException	An error occurred when executing the FTP command.
ConnectionTimeoutException	The connection to the FTP server has timed out before the command has completed its execution.

FTP Get Default Directory

Activity



The **FTP Get Default Directory** activity retrieves the name of the current remote directory. The default remote directory is operating system-dependent and determined by the remote FTP server. On UNIX systems, the default remote directory is usually the home directory of the user account that is used to establish an FTP connection.

The current remote directory may be different from the default directory, because you can use the [FTP Change Default Directory](#) or [FTP Quote](#) activities to issue an FTP command to change to a different directory. This activity returns the currently set remote directory.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
FTP Connection	No	Path to the shared configuration resource containing the user information such as host name, user name, and password. For more information, see FTP Connection .
Quit (post-command)	No	When checked, this field specifies that the FTP connection should be closed after completing this command. If this field is not checked, the FTP connection remains open so that all FTP activities that specify the same FTP session (that is, the same host, port, username, and password) share the FTP session until the connection is closed.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Host	string	The name of the FTP server. This overrides the value specified in the FTP Connection resource.
Port	integer	Port number for incoming FTP requests on the FTP server. This overrides the value specified in the FTP Connection

Input Item	Datatype	Description
		resource.
UserName	string	Username for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Password	string	Password for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Timeout	integer	The amount of time to wait for the FTP server to respond. This timeout is used when the FTP server places the FTP connection used by this activity into a waiting state. This overrides the value specified in the FTP Connection resource. For more information, see FTP Connection .

Output

The output for the activity is the following.

Output Item	Datatype	Description
CurrentDirectory	string	The currently set directory on the remote FTP server. By default, this directory is set by the FTP server, but you can use the FTP Quote activity to issue an FTP command to change to another directory.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ConnectionErrorException	An error occurred when connecting to the FTP server.
CommandExecutionException	An error occurred when executing the FTP command.

FTP Make Remote Directory

Activity



The **FTP Make Remote Directory** activity creates the specified directory on the remote FTP server.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
FTP Connection	No	Path to the shared configuration resource containing the user information such as host name, user name, and password. For more information, see FTP Connection
Quit (post-command)	No	When checked, this field specifies that the FTP connection should be closed after completing this command. If this field is not checked, the FTP connection remains open so that all FTP activities that specify the same FTP session (that is, the same host, port, username, and password) share the FTP session until the connection is closed.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Host	string	The name of the FTP server. This overrides the value specified in the FTP Connection resource.
Port	integer	Port number for incoming FTP requests on the FTP server. This overrides the value specified in the FTP Connection resource.
UserName	string	Username for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Password	string	Password for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
RemoteSITECommand	string	Optional, system-specific command to execute before the directory is created.
ParentRemoteDirectory	string	Path to the remote directory to create. This element is concatenated with the RemoteDirName element to provide the complete path to the directory to create. This element is optional.
RemoteDirName	string	Name of the remote directory to create. You can optionally include the complete path of the directory to create.
Timeout	integer	The amount of time to wait for the FTP server

Input Item	Datatype	Description
		to respond. This timeout is used when the FTP server places the FTP connection used by this activity into a waiting state.
		This overrides the value specified in the FTP Connection resource. For more information, see FTP Connection .

Output

This activity produces no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ConnectionErrorException	An error occurred when connecting to the FTP server.
CommandExecutionException	An error occurred when executing the FTP command.

FTP Put

Activity



The **FTP Put** activity issues an FTP put or mput command to the specified server. You can use process data as the content of the file to send to the remote server or you can send files in local disk storage. If you choose to use process data, you can place only one file on the remote server (FTP put command). If you use locally stored files, you can place one or more files on the remote server (FTP mput command).

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
FTP Connection	No	Path to the shared configuration resource containing the user information such as host name, user name, and password. For more information, see FTP Connection .
Quit (post-command)	No	When checked, this field specifies that the FTP connection should be closed after completing this command. If this field is not checked, the FTP connection remains open so that all FTP activities that specify the same FTP session (that is, the same host, port, username, and password) share the FTP session until the connection is closed.
Binary	Yes	Specifies whether to send the file content in binary or ASCII mode (checked signifies binary). When ASCII mode is used, differences in line endings between the FTP server and the machine executing the FTP command are automatically handled. Also, ASCII mode implies text data, and that data can be encoded in a character encoding other than ASCII (such as UTF-8 or Latin-1).
Timeout (msec)	Yes	The amount of time to wait (in milliseconds) for the FTP server to respond. This timeout is used when the FTP server places the FTP connection used by this activity into a waiting state. This overrides the value specified in the FTP Connection resource. For more information, see FTP Connection .

Field	Global Var?	Description
Overwrite Existing File	Yes	When this field is checked, any existing files on the FTP server are overwritten if the FTP command sends files with the same names. If this field is unchecked, any attempt to send a file to the remote server with the same name as an existing file results in an exception.
Append	Yes	This field is only available when the Overwrite Existing File field is unchecked. This field specifies whether to append data to an existing file. If the specified file exists on the FTP server, append the data to the file when this field is checked. If this field is not checked, the file is overwritten if it exists on the FTP server.
Use Process Data	Yes	Specifies to use process data as the content of the file to place on the remote server. If this field is checked, you can only place one file on the remote server.
Maintain 2.0 Compatibility	No	This field is present only for activities migrated from ActiveMatrix BusinessWorks 2.0. If checked, this field ensures the output schema retains the elements available in ActiveMatrix BusinessWorks 2.0.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
RemoteFilename	string	The path and name of the file to place onto the remote server. Wildcards are not permitted in this

Input Item	Datatype	Description
		field.
LocalFilename	string	The path and name of the local file to use in the FTP Put command. This item is only available if the Use Process Data field is unchecked.
Data	binary or string (depends upon if the Binary Configuration field is checked)	The contents of the file to be transferred to the FTP server, if the Use Process Data field is checked on the Configuration tab. You must map one or more process variables to this input item. The most common way to get the data for the remote file is to read a local file using the Read File activity, but you can map other data to this input item as well.
Host	string	The name of the FTP server. This overrides the value specified in the FTP Connection resource.
Port	integer	Port number for incoming FTP requests on the FTP server. This overrides the value specified in the FTP Connection resource.
UserName	string	Username for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Password	string	Password for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Encoding	string	When the file to send is not binary and the Use Process Data field on the Configuration tab is checked, this specifies the character encoding of the file to send to the FTP server. If this is not specified, it is assumed that the encoding of the file is the same encoding of the

Input Item	Datatype	Description
		machine where the FTP command is executed. If the encoding of the local machine and the FTP server are different, specify the encoding used by the FTP server in this element and the activity converts the file into the specified encoding.

Output

The output for the activity is the following.

Output Item	Datatype	Description
FilesTransferred	string	A repeating string element containing a list of names of files that were transferred. If an error occurred while transferring files, the activity output is not available. However, the Error Output tab lists the successfully transferred files and the files that failed to be transferred.
Success	boolean	Only available when Maintain 2.0 compatibility is checked on the Configuration tab. True or false depending upon whether the FTP <code>put</code> command was successfully executed by the FTP server.
Errormsg	string	Only available when Maintain 2.0 compatibility is checked on the Configuration tab. Any error messages returned from the FTP server.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO*

ActiveMatrix BusinessWorks™ Error Codes.

Exception	Thrown When
PutFilesException	An error occurred during the FTP put operation.
ConnectionErrorException	An error occurred when connecting to the FTP server.
CommandExecutionException	An error occurred when executing the FTP command.
ConnectionTimeoutException	The connection to the FTP server has timed out before the command has completed its execution.

FTP Quote

Activity



The **FTP Quote** activity sends an arbitrary FTP command to the FTP server. You can determine which FTP commands are supported by using the **Available Commands** button on the [FTP Connection](#) shared configuration resource. FTP commands vary by operating system and by FTP version and configuration, so you should determine which commands are available on the remote server before using this activity.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.

Field	Global Var?	Description
FTP Connection	No	Path to the shared configuration resource containing the user information such as host name, user name, and password. For more information, see FTP Connection .
Quit (post-command)	No	When checked, this field specifies that the FTP connection should be closed after completing this command. If this field is not checked, the FTP connection remains open so that all FTP activities that specify the same FTP session (that is, the same host, port, username, and password) share the FTP session until the connection is closed.
Has Socket Data	No	If checked, indicates that the FTP server send back the data as a result of executing the command.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Host	string	The name of the FTP server. This overrides the value specified in the FTP Connection resource.
Port	integer	Port number for incoming FTP requests on the FTP server. This overrides the value specified in the FTP Connection resource.
UserName	string	Username for logging into the FTP server. This overrides the value specified in the FTP Connection resource.

Input Item	Datatype	Description
Password	string	Password for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
RemoteSiteCommand	string	Optional, system-specific command to be executed before the specified command is executed.
ValidReturnCode	string	Optional list of valid return codes for validating the reply. Codes are three-digit numbers. To specify multiple codes, separate each code with a comma or a space. If the remote server returns a code not specified in this list, this activity raises an exception.
RemoteCommand	string	Command to be executed on the remote FTP server.
Timeout	integer	The amount of time to wait for the FTP server to respond. This timeout is used when the FTP server places the FTP connection used by this activity into a waiting state. This overrides the value specified in the FTP Connection resource. For more information, see FTP Connection.

Output

The output for the activity is the following.

Output Item	Datatype	Description
CommandItems	string	Each element in this repeating element is a returned value from the command executed on the remote server.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When
ConnectionErrorException	An error occurred when connecting to the FTP server.
CommandExecutionException	An error occurred when executing the FTP command.

FTP Remove Remote Directory

Activity



The FTP Remove Remote Directory activity deletes the specified directory from the remote FTP server.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
FTP Connection	No	Path to the shared configuration resource containing the user information such as host name, user name, and password. For more information, see FTP Connection .
Quit (post-	No	When checked, this field specifies that the FTP connection

Field	Global Var?	Description
command)		should be closed after completing this command. If this field is not checked, the FTP connection remains open so that all FTP activities that specify the same FTP session (that is, the same host, port, username, and password) share the FTP session until the connection is closed.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Host	string	The name of the FTP server. This overrides the value specified in the FTP Connection resource.
Port	integer	Port number for incoming FTP requests on the FTP server. This overrides the value specified in the FTP Connection resource.
UserName	string	Username for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Password	string	Password for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
ParentRemoteDirectory	string	Path to the remote directory you want to remove. This element is concatenated with the RemoteDirName element to provide the

Input Item	Datatype	Description
		complete path to the directory to remove. This element is optional.
RemoteDirName	string	Name of the remote directory to remove. You can optionally include the complete path of the directory to remove.
Timeout	integer	The amount of time to wait for the FTP server to respond. This timeout is used when the FTP server places the FTP connection used by this activity into a waiting state. This overrides the value specified in the FTP Connection resource. For more information, see FTP Connection .

Output

This activity produces no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ConnectionErrorException	An error occurred when connecting to the FTP server.
CommandExecutionException	An error occurred when executing the FTP command.

FTP Rename File

Activity



The FTP Rename File activity renames the specified file on the remote FTP server.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
FTP Connection	No	Path to the shared configuration resource containing the user information such as host name, user name, and password. For more information, see FTP Connection .
Quit (post-command)	No	When checked, this field specifies that the FTP connection should be closed after completing this command. If this field is not checked, the FTP connection remains open so that all FTP activities that specify the same FTP session (that is, the same host, port, username, and password) share the FTP session until the connection is closed.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Host	string	The name of the FTP server. This overrides the value specified in the FTP Connection resource.
Port	integer	Port number for incoming FTP requests on the FTP server. This overrides the value specified in the FTP Connection resource.
UserName	string	Username for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Password	string	Password for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
OldRemoteDirectory	string	Path to the remote file you want to rename. This element is concatenated with the OldRemoteFileName element to provide the complete path to the file to rename. This element is optional.
OldRemoteFilename	string	Name of the remote file you want to rename. You can optionally include the complete path of the file to rename.
NewRemoteDirectory	string	New path for the file to rename. This element is concatenated with the NewRemoteFileName element to provide the complete path to the file to rename. This element is optional.
NewRemoteFilename	string	New name for the remote file. You can optionally include the complete path of the file.
Timeout	integer	The amount of time to wait for the FTP server to respond. This timeout is used when the FTP

Input Item	Datatype	Description
		server places the FTP connection used by this activity into a waiting state.
		This overrides the value specified in the FTP Connection resource. For more information, see FTP Connection .

Output

This activity produces no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ConnectionErrorException	An error occurred when connecting to the FTP server.
CommandExecutionException	An error occurred when executing the FTP command.

FTP Sys Type

Activity



The FTP Sys Type activity retrieves the FTP server's operating system type.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
FTP Connection	No	Path to the shared configuration resource containing the user information such as host name, user name, and password. For more information, see FTP Connection .
Quit (post-command)	No	When checked, this field specifies that the FTP connection should be closed after completing this command. If this field is not checked, the FTP connection remains open so that all FTP activities that specify the same FTP session (that is, the same host, port, username, and password) share the FTP session until the connection is closed.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Host	string	The name of the FTP server. This overrides the value specified in the FTP Connection resource.
Port	integer	Port number for incoming FTP requests on the FTP server. This overrides the value specified in the FTP Connection resource.
UserName	string	Username for logging into the FTP server. This overrides the value specified in the FTP Connection resource.

Input Item	Datatype	Description
Password	string	Password for logging into the FTP server. This overrides the value specified in the FTP Connection resource.
Timeout	integer	The amount of time to wait for the FTP server to respond. This timeout is used when the FTP server places the FTP connection used by this activity into a waiting state. This overrides the value specified in the FTP Connection resource. For more information, see FTP Connection .

Output

The output for the activity is the following.

Output Item	Datatype	Description
SystemType	string	The FTP server type. This element can be one of the following: • UNIX • UUNIX • Windows_NT • MVS_OS390 • UNKNOWN

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ConnectionErrorException	An error occurred when connecting to the FTP server.
CommandExecutionException	An error occurred when executing the FTP command.

General Activities Palette

The **General Activities** palette contains several general-purpose activities and process starters. For example, the Timer process starter is used to start a process at a specific time. This section describes the activities contained in the **General Activities** palette.

Assign

Activity



The **Assign** activity allows you to assign a value to a user-defined process variable. For more information about creating user-defined process variables, see [Process Definition](#).

When the **Assign** activity is executed, the entire schema for the selected process variable is replaced with the specified values. Elements that do not have a value specified in the Input tab are set to null. Therefore, be certain to set all necessary values when using the **Assign** activity to set a process variable.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Process Variable to Set	A drop down list of the available user-defined process variables for the process definition. You can select any user-defined process variable to modify.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

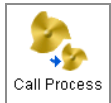
Input Item	Datatype	Description
<schema>		The specified data schema for the user-defined process variable selected on the Configuration tab.

Output

This activity produces no output.

Call Process

Activity



The **Call Process** activity calls and executes an existing process definition. The input to the called process is defined in the Start activity of the called process. The output of the called process is defined in the End activity of the called process. For more information about specifying the input and output of a process, see the description of the Start and End activities in *TIBCO ActiveMatrix BusinessWorks™ Process Design*.



Warning: You should only call processes that have Start activities. Do not call processes that begin with a process starter such as HTTP Receiver or Receive Mail. Providing the name of a process with a process starter results in an exception at run time.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Process Name	The process definition you would like to call. You can browse the available process definitions.
Process Name Dynamic Override	An XPath formula specifying the name of the process to call. Use this field to dynamically determine which process to call when the process instance is running. For more information about this field, see Dynamically Determining the Process to Call.
Spawn	Specifies whether to spawn a new machine process for executing the called process. If this option is checked, the parent process cannot access the called process' output. The called process is executed in a separate process instance.
Custom Id	This field is only available when the Spawn field is checked. This field can contain an XPath expression that specifies a custom ID for the spawned process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable in the spawned process instance.
Custom Icon File	File to use for the Call Process activity in the process definition. This allows you to customize the look of your process definition with your own images.

Dynamically Determining the Process to Call

You can use the Process Name Dynamic Override field on the Configuration tab to specify an XPath expression that determines which process to call. This is useful for calling a different process depending upon the value of the XPath expression. For example, you have an incoming HTTP request that contains an attribute named `orderAmount`. For all incoming orders over \$10,000, you want to call a process that includes activities in the Manual Work palette to obtain the proper approvals. For orders

under \$10,000, you want to call a process that handles the order automatically. You can create the following expression that resolves to the proper process name for each type of order:

```
if($IncomingOrder/orderAmount > 10000) then
    '/MyProject/manualApproval.process'
else '/MyProject/processOrder.process'
```

Use the full path and name of the process file as stored in the project directory.

When you use the Process Name Dynamic Override field, you must also specify a process to call in the Process field. The input, output, and error definitions of the specified process must be the same as any process that the expression in the Process Name Dynamic Override field can evaluate to.

- ✓ **Tip:** It may be helpful to create process definitions that act as programmatic interfaces when using the Process Name Dynamic Override field.
- ✓ **Tip:** In the example above, you may create a process named `orderProcessOrApprove`. The only purpose of this process is for specifying the input, output, and error schemas. Place this process in the Process field of the Call Process tab. Then, create the `manualApproval` and `processOrder` process definitions as copies of the `orderProcessOrApprove` process. The **Call Process** activity then has the correct input, output, and error schemas for all processes that can be called.
- i **Note:** If you use the Process Name Dynamic Override field, make sure you include all potentially callable subprocesses when you create your Process Archive for deployment. TIBCO Designer cannot determine which subprocesses are potentially callable at design time, and therefore they cannot be automatically included in a process archive. For more information about creating process archives, see *TIBCO Designer™ User's Guide*.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
input	varies	The input to the called process is defined by the Output Editor tab in the called process' Start activity. For more information, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i> .

Output

The output for the activity is the following.

Output Item	Datatype	Description
output	varies	The output of the called process is defined by the Input Editor tab in the called process' End activity. For more information, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i> .

Catch

Activity



The **Catch** activity receives control of execution when an unhandled exception occurs. You can select a specific exception type to catch or you can specify that this activity should catch all unhandled exceptions. You can have more than one **Catch** activity in each exception scope, but each **Catch** activity must have a unique exception type.

The **Catch** activity allows you to transition to activities you want to perform to handle the exception. Transitions are permitted between Catch tracks within an exception scope, but you cannot transition back to the main execution track from the Catch track.

If you want to propagate the caught exception to the next highest scope, use the [Rethrow](#) activity.

For more information about using the **Catch** activity, see the description error handling in *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Catch All	Checking this box indicates you want processing to resume with this activity when any exception is encountered that is not already handled by an error transition or another Catch activity. You can have only one Catch activity within an exception scope that has this field checked.
Exception to Catch	Specifies the exception type to catch. The list of available exceptions that can be raised in the current scope is automatically placed in the drop-down list in this field. This field is not available when the Catch All field is checked. Each Catch activity within an exception scope must specify a different value for this field.

Input

This activity has no Input.

Output

The output for the activity is the following.

Output Item	Datatype	Description
<exceptionName>	varies	Contains the schema for the thrown exception. The contents of this element vary depending upon the exception that is encountered.

Checkpoint

Activity



The **Checkpoint** activity performs a checkpoint in a running process instance. A checkpoint saves the current process data and state so that it can be recovered at a later time in the event of a failure. If a process engine fails, all process instances can be recovered and resume execution at the location of their last checkpoint in the process definition. If a process instance fails due to an unhandled exception or manual termination, it can optionally be recovered at a later time, if the process engine is configured to save checkpoint data for failed processes. For more information about recovering failed process instances, see *TIBCO ActiveMatrix BusinessWorks™ Administration*.

Only the most recent state is saved by a checkpoint. If you have multiple checkpoints in a process, only the state from the last checkpoint is available for recovering the process.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
duplicateKey	String	A key value that is used to compare to other process instances to determine whether another process instance with the same duplicateKey value already exists. For more information on detecting duplicate process instances, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>

Checkpoints and Transactions

A **Checkpoint** activity cannot be placed in or in parallel to a transaction. You can, however, specify that an implicit checkpoint should be taken as part of a transaction by checking the Include Checkpoint field on a transaction group.

For explicit Checkpoints, place the **checkpoint** activity outside of any transaction group. Also, make sure that if you have multiple paths in your process definition, the **Checkpoint** activity does not occur in parallel with a path that has a transaction group. Instead, any **Checkpoint** activities should be placed at points that are guaranteed to be reached before or after the transaction group is reached.

For more information about creating groups for transactions and including an implicit checkpoint in a transaction, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*

Called Processes

Checkpoints save the state of the entire process instance. By default when a process calls another process, the subprocess is executed in the same process instance as the calling process. If the called process spawns a new machine process, however, the called process is a new process instance.

When a checkpoint occurs in a called process, the checkpoint saves the state of the current process instance. If no called processes spawn new process instances, then a checkpoint in any called process saves the state of the process instance, including state from the parent process(es) of the current process. In the case of a called process that spawns a new process instance, only the spawned process instance is saved.

Recovering After a Crash

If a process engine crashes, all process instances can be recovered up to the point of their last checkpoint. You must be careful with certain types of process starters or incoming events when placing your checkpoint in a process definition.

For example, if the process starter is waiting for an incoming HTTP request, and a checkpoint is taken after the process starts but before the response to the request is sent, the process cannot respond to the request when the process instance is restarted. The socket for the HTTP request is closed when the process engine crashes, therefore the **Send HTTP Response** activity in the restarted process returns an error. In this case, place the response activity before the checkpoint so that any response is sent before a checkpoint is taken.

There are other examples of situations where an incoming event must be handled before the checkpoint is taken. The following lists some of these circumstances:

- An email message is received, then deleted from the email server.
- An HTTP request is received.

You should exercise care in placing checkpoints in your process definitions. Make certain that the process has all of the data required to continue at the time of the checkpoint so that in the event of a failure, a restarted process does not attempt to access resources that no longer exist.

i Note: By default, checkpointed process instances are restarted when the engine restarts. If the engine encounters errors during startup, the restarted process instances continue to be processed and may eventually be lost depending upon the type of error at startup. You can specify to shut down the process engine if any errors are encountered during startup so that checkpointed jobs are not lost in the event of an error. The custom engine property named `Engine.ShutdownOnStartupError` controls this behavior. By default, the value of the property is `false`. Setting the property to `true` shuts the engine down if errors are encountered when the engine starts. For more information about setting custom engine properties, see *TIBCO ActiveMatrix BusinessWorks™ Administration*.

Checkpoints and the Confirm Activity

In the case of confirmable messages (for example, a confirmable TIBCO Rendezvous or Adapter message is received), you must consider the consequences of performing a checkpoint before or after a [Confirm](#) activity.

If the checkpoint is taken before the **Confirm** activity, then a crash occurs after a checkpoint but before a confirm, the original message is resent. In this case, the restarted process can no longer send the confirmation. However, a new process is started to handle the resent message, and you can implement your process to handle the restarted and new processes appropriately.

If the checkpoint is taken after a **Confirm** activity, there is potential for a crash to occur after the Confirm but before the checkpoint. In this case, the message is confirmed and therefore not redelivered. The process instance is not restarted, because the crash occurred before the checkpoint.

You must consider the type of processing your process definition performs to determine when a checkpoint is appropriate if your process definition receives confirmable messages.

Confirm

Activity



The **Confirm** activity confirms any confirmable messages received by the process instance. For example, if a process is started because of the receipt of an RVCM message, the **Confirm** activity can send a confirmation message to the publisher

of the RVCM message.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Confirm Event	The message you want to confirm. This is a drop down list of any process starter or activity that expects an incoming event in the current process definition. You can only confirm messages from confirmable message sources, for example, RVCM, RVDQ, or JMS messages. E-mail is also confirmable when received by the Receive Mail process starter (the e-mail message is deleted from the server when it is confirmed). Select only the event that sent the confirmable message. If you select an event from the list that does not correspond to a confirmable message (for example, a Wait for File Change event) or if you try to confirm the same message more than once, no message is confirmed.

For more information about using the Checkpoint and Confirm activities in a process definition, see [Checkpoints and the Confirm Activity](#).

It is not recommended to use the **Confirm** Activity with in the Service Resource process as this can unblock the ActiveMatrix BusinessWorks provider thread. The new message could still flow on that thread eventhough the ActiveMatrix BusinessWorks process execution continues for the previous message.

In case of the Service Resource operation process, perform Auto-confirm in case of clientAck after the end Activity which would still work even if you are not using **Confirm** Activity in the Operation Process.

Custom Activity

Resource



The **Custom Activity** resource allows you to create an activity that references a process definition. This is useful if you want to develop processes for others to use, but you also want to hide the implementation details of the process from users. The **Custom Activity** behaves like any other ActiveMatrix BusinessWorks activity.

You can also package and distribute custom activities within custom palettes. For more information about creating and packaging custom activities, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the custom activity. This name also becomes the default name of the activity when it is dragged and dropped into a process definition.
Description	Short description of the activity.
Select Process	The process definition that contains the implementation of the custom activity. You can only specify process definitions that start with the Start activity. You cannot specify process definitions that use any other process starter. Any resources referenced by the selected process are automatically included in the custom activity. Dynamically called subprocesses are not known at design time, and therefore are not referenced by the process definition. Therefore, dynamically called subprocesses cannot be used by a process definition within a custom activity. Global variables referenced by the resources in the custom activity are added to the <code>\$_globalVariables</code> process variable of the process that uses the custom activity. You should document any required global variables so that users of the custom activity can specify appropriate values for the

Field	Description
	variables.
Custom Icon File	The icon you want to display for the custom activity.

Engine Command

Activity



The **Engine Command** activity allows you to retrieve statistics and information about process definitions, process instances, and activities in the currently running process engine. This activity also lets you perform engine maintenance, such as suspending and resuming process instances and shutting down the engine.

This activity uses TIBCO Hawk to perform the desired command. A direct call is made to the TIBCO Hawk AMI method (no message is sent).



Note: Before using this activity, the Hawk.Enabled property must be set to either true or local. Also, some commands require memory and processor overhead for gathering statistics or for getting information on the current state of the process. Because of the performance implications, certain instrumentation is disabled by default. You can enable instrumentation for a specific process or for all processes with the Instrumentation property. For more information about setting custom engine properties, see *TIBCO ActiveMatrix BusinessWorks™ Administration*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Command to Execute	No	The command to execute. For more information about the commands and their input and output, see Commands .
Process Definition Name	No	Certain commands accept a process definition name. This field allows you to specify the process definition name that applies to the selected command. This field only appears when the one of the following commands is selected in the Command to Execute field: • GetActivityStats • GetProcessInstanceInfo • ResumeProcessStarter • SuspendProcessStarter

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
<commandInput>	varies	The input for this activity varies depending upon the command selected in the Command to Execute field on the Configuration tab. For more information about the input for each command, see Commands .

Output

The output for the activity is the following.

Output Item	Datatype	Description
<commandOutput>	varies	The output for this activity varies depending upon the command selected in the Command to Execute field on the Configuration tab. For more information about the output for each command, see Commands .

Commands

The following are the commands that can be executed by this activity and the corresponding input and output for each command.

GetActivityStats

Retrieves information about the activities that have been executed for a given process definition since the engine was started. The activity information is cumulative. A single activity name represents all executions of that activity.

The ExecutionTime computation for the **Call Process** Activity includes the sum of the execution times for all activities in the called process, not just the execution time for the **call process** activity itself.

Input Item	Datatype	Description
ProcessDefinitionName	String	Name of the process definition. Specifying a value for this element overrides the process definition specified on the Configuration tab.

Output Item	Datatype	Description
ProcessDefinitionName	String	Name of the process definition.

Output Item	Datatype	Description
ActivityName	String	Name of the activity.
ActivityClass	String	Name of the class that implements the activity.
ExecutionCount	integer	Number of times the activity has been executed.
ElapsedTime	integer	Total clock time (in milliseconds) used by all executions of this activity. This includes waiting time for Sleep, Call Process, and Wait For activities.
ExecutionTime	integer	Total clock time (in milliseconds) used by all executions of this activity. This does not include waiting time for Sleep, Call Process, and Wait For activities.
ErrorCount	integer	Total number of executions of the activity that have returned an error.
LastReturnCode	String	Status code returned by most recent execution of this activity. This can be either OK, DEAD, or ERROR.
TracingEnabled	boolean	True if tracing is enabled for this activity, false if tracing is disabled.
MinElapsedTime	integer	Elapsed clock time (in milliseconds) of the activity execution that has completed in the shortest amount of elapsed time.
MaxElapsedTime	integer	Elapsed clock time (in milliseconds) of the activity execution that has completed in the longest amount of elapsed time.
MinExecutionTime	integer	Execution time (in milliseconds) of the

Output Item	Datatype	Description
		activity execution that has completed in the shortest amount of execution time.
MaxExecutionTime	integer	Execution time (in milliseconds) of the activity execution that has completed in the longest amount of execution time.
MostRecentElapsedTime	integer	Elapsed clock time (in milliseconds) of the most recently completed activity execution.
MostRecentExecutionTime	integer	Execution time (in milliseconds) of the most recently completed activity execution.
TimeSinceLastUpdate	integer	Time (in milliseconds) since the statistics have been updated.
ExecutionCountSinceReset	integer	A comma-separated list of names of process definitions called by this activity.
CalledProcessDefs	String	Number of activity executions that have completed since the last reset of the statistics.

GetProcessDefinitionStats

Retrieves information about process definitions. |

Output Item	Datatype	Description
ProcessDefinitionName	String	Name of the process definition.
ProcessStarterName	String	Name of the process starter for the process.
NumberCreated	integer	Number of process instances created for this process definition.

Output Item	Datatype	Description
NumberSuspended	integer	Number of times process instances have been suspended.
NumberSwappedToDisk	integer	Number of times process instances have been swapped to disk.
NumberQueued	integer	Number of times process instances have been queued for execution.
NumberAborted	integer	Number of times process instances have been aborted.
NumberCompleted	integer	Number of process instances that have been successfully completed.
NumberCheckpointed	integer	Number of times process instances have executed a checkpoint.
TotalExecutionTime	integer	Total execution time (in milliseconds) for all successfully completed process instances.
AverageExecutionTime	integer	Average execution time (in milliseconds) for all successfully completed process instances.
TotalElapsedTime	integer	Total elapsed time (in milliseconds) for all successfully completed process instances.
AverageElapsedTime	integer	Average elapsed clock time (in milliseconds) for all successfully completed process instances.
MinElapsedTime	integer	Elapsed clock time (in milliseconds) of the process instance that has completed in the shortest amount of elapsed time.

Output Item	Datatype	Description
MaxElapsedTime	integer	Elapsed clock time (in milliseconds) of the process instance that has completed in the longest amount of elapsed time.
MinExecutionTime	integer	Execution time (in milliseconds) of the process instance that has completed in the shortest amount of execution time.
MaxExecutionTime	integer	Execution time (in milliseconds) of the process instance that has completed in the longest amount of execution time.
MostRecentExecutionTime	integer	Execution time (in milliseconds) of the most recently completed process instance.
MostRecentElapsedTime	integer	Elapsed clock time (in milliseconds) of the most recently completed process instance.
TimeSinceLastUpdate	integer	Time (in milliseconds) since the statistics have been updated.
NumberCompletedSinceReset	integer	Number of process instances that have completed since the last reset of the statistics.

GetProcessInstanceExceptions

Retrieves error information reported by the specified process.

Input Item	Datatype	Description
ProcessId	integer	ID for the process instance. If not specified, or if 0 is specified, exceptions for all process instances are returned.

Output Item	Datatype	Description
ExceptionSequenceNumber	integer	Sequence number of the exception, with the most recent exception first.
ProcessId	integer	ID for the process instance.
ExceptionMessage	String	Exception message.
StackTrace	String	Exception stack trace.
ExceptionClass	String	Exception class name.
ProcessStack	String	Process stack at exception. This displays the [ProcessName/GroupName/ActivityName] of the activity issuing the exception. If the activity is in a called sub-process, then the calling activity's process stack plus a '>' separator character gets pre-pended to the normal information to produce the process stack of the activity issuing the exception.
TrackingID	integer	Tracking ID for the process instance.
ProcessDefinitionName	String	Name of the process definition.
State	String	State of the process.

GetProcessInstanceInfo

The values of the input elements for this command specify for which process instances you want to retrieve information. If you specify more than one input element, the values are treated as an AND condition. For example, If you specify the process definition name and the minimum duration, information for process instances for the specified process definition that meet the minimum duration time will be returned.

Input Item	Datatype	Description
ProcessId	integer	ID for the process instance.
ProcessDefinitionName	String	Name of the process definition used by the process instance. Specifying a value for this element overrides the process definition specified on the Configuration tab.
MinimumDuration	integer	Minimum time (in milliseconds) in elapsed clock time since the process instance started. All process instances with greater elapsed times get retrieved.

Output Item	Datatype	Description
ProcessId	integer	ID for the process instance.
ProcessInstanceName	String	Name of the process definition used by the process instance.
TrackingId	String	Tracking ID for the process instance.
CustomId	String	Custom ID for the process instance.
Status	String	Status of the process.
StartTime	integer	Time (in milliseconds) when the process instance started.
ElapsedTimeSinceStarted	integer	Elapsed clock time (in milliseconds) since the process instance started.
MainProcessName	String	Name of the main process definition.
CurrentActivityName	String	Name of the currently executing activity in the process instance.
ProcessStarterName	String	Name of the process starter that started this

Output Item	Datatype	Description
		process instance.
SubProcessName	String	Name of the process definition for the sub-process.

GetProcessStarterStats

Retrieves information about either active or inactive process starters. The information is cumulative. A single process starter name represents all executions of that process starter.

Output Item	Datatype	Description
ProcessDefinitionName	String	Name of the process definition.
ProcessStarterName	String	Name of the process starter.
Status	String	Status of the process starter. The status can be INACTIVE, ACTIVE, or READY.
TotalNumberCreated	integer	Number of process instances created by this process starter.
NumberCreatedPerHour	integer	Number of process instances per hour created by this process starter.
NumberRunning	integer	Number of process instances currently executing.
NumberCompleted	integer	Number of process instances that have completed.
StartTime	String	Time (in milliseconds) at which the process starter was started.
ElapsedTimeSinceStarted	integer	Elapsed clock time since the process starter was started.

Output Item	Datatype	Description
RestartedFromCheckpoint	boolean	True if the process was restarted from a checkpoint.
TracingEnabled	boolean	True if tracing is enabled for this process starter, false if tracing is disabled.

GetRecoverableProcesses

Retrieves the process instances that can be recovered. For more information about recoverable process instances, see *TIBCO ActiveMatrix BusinessWorks™ Administration*. Use the returned process ID in the RestartRecoverableProcess or RemoveRecoverableProcess commands.

Output Item	Datatype	Description
ProcessId	integer	Process ID of the process instance that can be restarted.
Status	string	Status of the process instance. Can be one of the following: - faulted – this status occurs when the process instance is terminated due to an unhandled exception. - interrupted – this status occurs when the process instance is terminated due to engine failure (and the instance did not automatically restart when the engine restarted) or a manual termination.
TrackingId	string	Tracking ID for the process instance.
CustomId	string	Custom ID for the process instance.
ProcessDefinitionName	string	Process definition name for this process instance.

Output Item	Datatype	Description
RestartActivityName	string	Name of the last executed Checkpoint activity in the process instance. This is the point at which the process instance begins executing when it is restarted.

KillProcessInstance

Kills the specified process instance. The process instance is stopped immediately and permanently removed from the engine.

Input Item	Datatype	Description
ProcessId	integer	Process ID of the process instance you want to kill.

ListAllRoles

Returns a list of all roles, along with the current state (enabled or disabled) of each role.

Output Item	Datatype	Description
Role	string	Name of the role.
Enabled	boolean	True if the role is enabled, false if the role is disabled.

ListUserRoles

Returns a list of all user roles, along with the current state (enabled or disabled) of each role.

Output Item	Datatype	Description
Role	string	Name of the role.
Enabled	boolean	True if the role is enabled, false if the role is disabled.

RemoveRecoverableProcess

Removes the specified recoverable process instance from the list of potential recoverable processes. After executing this command, the checkpoint data of the specified process instance is removed and the process instance can no longer be able to be recovered. Obtain the process ID of the recoverable process with [GetRecoverableProcesses](#) the command.

Input Item	Datatype	Description
ProcessId	integer	The name or process ID of the process instance you want to remove.

RestartRecoverableProcess

Restarts the specified recoverable process instance. Obtain the process ID of the recoverable process with [GetRecoverableProcesses](#) the command.

Input Item	Datatype	Description
ProcessId	integer	The name or process ID of the process instance you want to restart.

ResumeProcessInstance

Resumes the specified process instance.

Input Item	Datatype	Description
ProcessId	integer	The name or process ID of the process instance you want to resume. You can retrieve the process ID for a process instance by using the GetProcessInstanceInfo command.

ResumeProcessStarter

Resumes the specified process starter.

Input Item	Datatype	Description
ProcessDefinitionName	string	The name of the process definition whose process starter you want to resume. Specifying a value for this element overrides the process definition specified on the Configuration tab.

StartStatsCollector

Enables collection of statistics for each executed activity. For more information about collecting activity statistics, see *TIBCO ActiveMatrix BusinessWorks™ Administration*.

Output Item	Datatype	Description
FileName	string	Name of the file containing the collected data.

StopStatsCollector

Disables collection of statistics for each executed activity. For more information about collecting activity statistics, see *TIBCO ActiveMatrix BusinessWorks™ Administration*.

Shutdown

Shuts down the process engine. If no input is provided, this command shuts down the engine immediately. You can optionally specify an amount of time to delay the shut down of the engine or you can specify that the engine should wait for any checkpointed process instances to complete before shutting down.

Input Item	Datatype	Description
MaxDelayInSeconds	integer	Specifies the amount of time (in seconds) to wait before shutting down the process engine.
WaitForCheckpoints	boolean	When true is specified, the engine waits for any checkpointed process instances to complete before shutting down.

SuspendProcessInstance

Suspends the specified process instance.

Input Item	Datatype	Description
ProcessId	integer	The name or process ID of the process instance to suspend. You can retrieve the process ID for a process instance by using the GetProcessInstanceInfo command.

SuspendProcessStarter

Suspends the specified process starter.

Input Item	Datatype	Description
ProcessDefinitionName	string	The name of the process definition whose process starter you want to suspend. Specifying a value for this element overrides the process definition specified on the Configuration tab.

External Command

Activity



The **External Command** activity allows you to execute an operating system command. This activity waits for the command to complete before transitioning to the next activity. Optionally, the command output and any errors can be included in this activity's output schema, written to a file, or both.

Long-Running Commands

If you want to execute long-running commands (such as daemons), it may not be practical to execute the command directly. Because the specified command must

terminate before control is passed to the next activity, the process instance must run until the external command completes. To avoid this problem, you may want to create a script that runs the desired commands in the background.

The syntax of executing commands as background processes differs by operating system. For example, on UNIX, to run a command in the background, you append the ampersand character (&) to the command. On Microsoft Windows, you use the START command in a batch file to run a command in a different process.

After creating a script to run commands in the background, specify the script as the command to run in the **External Command** activity. The script runs the desired command and returns the process ID of the process it started. The return code of the script is stored in the `returnCode` item of the **External Command** activity's output.

If you want to start a long-running command in the background and then later terminate the background process, you should store the process ID returned by the script in a file or database table. Another process instance can then read the process ID and kill the process with the appropriate operating system command when necessary.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Command to Execute	Yes	The command line to execute. You can also specify any input and command-line arguments to the command in this field. You cannot specify more than one command by piping output from one command to another (for example, <code>ls more</code> is not allowed). You can specify a global variable for this configuration field. This allows you to easily specify different commands on different operating systems. For example, you may specify

Field	Global Var?	Description
		%%EXTERNAL_COMMAND%% as the value for this field. On MS Windows, the value of the global variable may be the following: <pre>cmd /c dir</pre> On UNIX, the global variable may have the following value: <pre>ls</pre> If the command to execute is not located in one of the directories specified in the PATH environment variable (on UNIX or Windows), you can specify the full path for the executable command or you can use the environment input element to specify the environment for the command.
Make Output Available as Activity Output	Yes	When checked, this field specifies that the output sent to standard output and standard error by the command should be available in the output schema of this activity. For commands that produce a large amount of output, it is recommended to uncheck this field and write the output to a file. This saves memory and allows you to use other activities, such as Read File and Parse Data, to handle the output file more efficiently.
Remove Parameter Quotes	Yes	When checked, this field specifies that the quotes surrounding the parameters should be removed. The quotes get removed even if the parameter contains space(s). By default, the check box is not selected.
Output Filename	Yes	Name and location of the file for storing any output or errors produced by the command. If this field is blank, no output file is created.
Output Line	Yes	Controls whether the command output and errors are split into

Field	Global Var?	Description
Splitting		multiple strings. This field can have the following values: • None – The activity produces a single string containing the command's output or errors. Line end characters are included in this string. • At Operating System Line End – The activity produces a repeating string element with one string element per output line. The lines are split by default on the carriage return/line feed character, depending on the operating system. The line end characters are not included in the output. • At Specified Token – The lines are split wherever the specified token occurs. The token is specified as an activity input element, and the token is not included in the output.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
command	string	The command to execute, including any input or command-line arguments. Specifying this input item overrides the command specified on the Configuration tab.
input	string	An input string to supply to the command's standard input. This schema item accomplishes the same goal as entering the command from a command prompt and then typing in input without specifying a carriage

Input Item	Datatype	Description
		return. For example, on UNIX, typing <code>sed</code> into a command prompt allows you to enter more text as input into the <code>sed</code> command. Use <code>&crf;</code> in the string for this item to specify a carriage return (that is, <code>&crf;</code> simulates pressing the Enter key).
environment	string	The environment variable settings used for running the command. This element is specified as a comma-separated list of <code><name>=<value></code> pairs where <code><name></code> is the name of the environment variable and <code><value></code> is the value of the environment variable. If no value is specified for this element, the command is run with the environment variable settings that are specified in the login scripts for the user account used to execute the ActiveMatrix BusinessWorks process engine. If a value is specified, the value of this element replaces any environment settings for the user account used to run the ActiveMatrix BusinessWorks process engine. That is, the user environment is ignored and this element's value is used instead.
workingDirectory	string	The working directory for the command process. If <code>workingDirectory</code> is not specified, the command process inherits its working directory from the process engine.
token	string	Command output can be split into multiple strings based on the token specified in this element. This field is only available when <code>At Specified Token</code> is chosen in the <code>Line Split</code> field on the <code>Configuration</code> tab. The following escape sequences are supported in this element:

Input Item	Datatype	Description
		• \r – newline carriage return character • \n – newline character • \t – tab character • \ – backslash character For example, \r\n signifies to split at a carriage return/linefeed, \ – signifies to split when a backslash is encountered.

Output

The output for the activity is the following.

Output Item	Datatype	Description
returnCode	integer	The numeric return code returned by the command. Typically, a value of 0 in this element signifies the command executed successfully. A non-zero value usually indicates termination of the command due to an error.
output	string	Any output produced by the command and sent to stdout. This output item is only available if the Make Output Available as Activity Output field is checked on the Configuration tab. This element can be repeating if the Line Split configuration setting causes the command's output to be split into several lines.
error	string	Any errors produced by the command and sent to stderr. This element is only available if the "Make Output Available as Activity Output" field is checked on the Configuration tab. This element can be repeating if the Line Split configuration setting causes the command's errors to be split into several lines.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
InvalidInputException	There is an error in the activity's configuration or input mapping that is preventing the command from being executed.
CommandExecutionError	The command could not be executed.
FileIOError	There was an error when attempting to write the output to a file. Make sure the user that is running the ActiveMatrix BusinessWorks engine has permission to write to the output file.

Generate Error

Activity

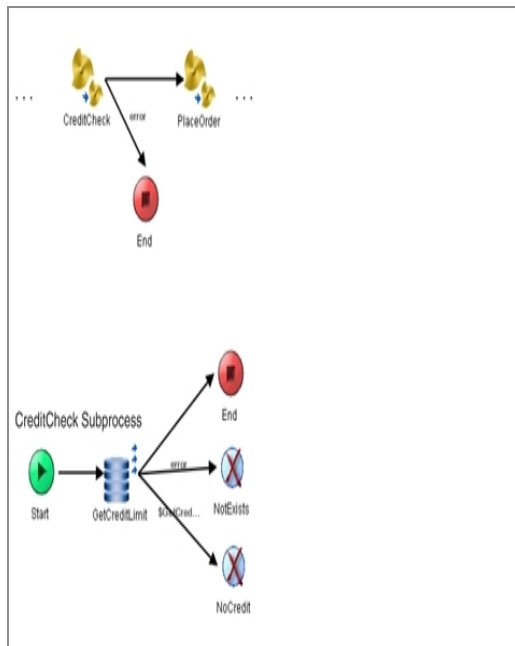


The **Generate Error** activity generates an error and causes an immediate transition to any error transitions. If there are no error transitions, the process instance halts execution. This activity is useful in a group or in a called process. If you would like to catch and raise your own error conditions, you can use this activity to do so.

For example, for a process that calls a subprocess to check the credit of a customer, you can use the **Generate Error** activity to raise an error for any error conditions, such as the customer does not exist, or the customer has no credit available. In the called subprocess, create a transition to the **Generate Error** activity after detecting the error condition and the called process terminates. Processing continues with an error in the calling process. An error transition on the **Call Process** activity can handle the error.

[Using the Generate Error activity](#) illustrates the example described above.

Figure 9: Using the Generate Error activity



For more information about handling errors in process definitions, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Select Error Schema	The process error schema to send to the parent process. - Default - specifies that the error should be propagated only in the <code>\$_error</code> process variable. Process error schemas are defined on the Error Schemas tab of the End activity for the process. The schema specified in this field is added to the Input schema for this activity. Data mapped to the error schema is

Field	Description
	propagated to the parent process.
	For more information about specifying process error schemas, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i> .

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
message	string	The text of the error message to generate.
messageCode	string	The error code of the error. This is used as an identifier for the error so that applications can test for the error.
data	complex	The schema specified in the Select Error Schema field of the Configuration tab is contained in this input item. Map data from the list of process variables to this input item to propagate the data to the parent process.

Output

This activity produces no output.

Get Shared Variable

Activity



The **Get Shared Variable** activity retrieves the current value of a [Shared Variable](#) or [Job Shared Variable](#) resource. If you are using this activity to retrieve the value of a Shared Variable resource, you may want to use a critical section group to

ensure that no other process instances are altering the value of the shared variable at the same time.

For more information about shared variables and critical section groups, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Variable Configuration	No	The Shared Variable or Job Shared Variable whose value you would like to retrieve.

Output

The output for the activity is the following.

Output Item	Datatype	Description
<schema>	varies	The value of the shared variable specified in the Variable Configuration field of the Configuration tab is the output for this activity. The schema for the output is determined by the schema specified for the shared variable.

Inspector

Activity



The **Inspector** activity is used to write the output of any or all activities and process variables to a file and/or stdout. This is particularly useful when debugging process definitions and you want to see the entire schema instead of mapping specific elements to the **Write File** activity.

You can use the **inspector** activity to write the output of any activity or process variable in the current process. Activities and process variables in a subprocess are not available to the **inspector** activity (but the output of a **Call Process** activity can be written using the **inspector** activity). If you want to obtain the output from one or more activities or process variables in a subprocess, place the **inspector** activity in the process definition of the subprocess.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Stdout	Specifies whether the output of this activity should be sent to stdout. When checked, the output is sent to stdout, when unchecked, the output is not sent to stdout.
Activity	The output of the process variable you want to output. This is a drop down list of the available items to output. You can select All to output all available output. You can also select global variables, process context, error variables, or user-defined process variables to output. Only activities that have output and are executed prior to the Inspector activity are shown in the drop down list.
Append to File	When checked, specifies that the output should be appended to the file specified in this activity's input. If unchecked, any file that exists with the specified name is overwritten.
Suppress Job	When checked, this suppresses the output of header information, such as

Field	Description
Headers	the process instance ID and the timestamp. If unchecked, header information for the current process instance is written before the output of the activity or process variable.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
fileName	string	The absolute path to the file you want to write. This file contains the output of the specified activity or process variable.

Output

The output for the activity is the following.

Output Item	Datatype	Description
output	string	The output of the process variable or activity specified on the Configuration tab in string form. This can be used if you want to use this output in subsequent activities, in addition to writing the output to a file or stdout. For example, you can use this output as the body of a TIBCO Rendezvous message.

JNDI Configuration

Shared Configuration



The JNDI Configuration shared configuration resource provides a way to specify JNDI connection information that can be shared by other resources.

This resource can be specified in any resource that permits JNDI connections. For example, [JDBC Connection](#) and [JMS Connection](#) can use JNDI connections.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.
JNDI Context Factory	No	The initial context factory class for accessing JNDI. (javax.naming.Context.INITIAL_CONTEXT_FACTORY). You can choose from the drop down list of supported classes. For more information when using IBM WebSphere, see IBM WebSphere and JNDI Context Caching. Note: ActiveMatrix BusinessWorks attempts to find the class. However, you may need to add the JAR file supplied by your JNDI service provider to the CLASSPATH environment variable to use JNDI.
JNDI Context URL	Yes	The URL to the JNDI service provider (javax.naming.Context.PROVIDER_URL). An example URL is provided when one of the supported JNDI context factory classes is selected. For the syntax of the URL, see your JNDI provider documentation.
JNDI User	Yes	User name for logging into the JNDI server

Field	Global Var?	Description
Name		(javax.naming.Context.SECURITY_PRINCIPAL). If the JNDI provider does not require access control, this field can be empty.
JNDI Password	Yes	Password for logging into the JNDI server (javax.naming.Context.SECURITY_CREDENTIALS). If the JNDI provider does not require access control, this field can be empty.

Test Configuration Button

The **Test Configuration** button allows you to test that the specified configuration fields result in a valid connection to a JNDI server.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Validate JNDI Security Context	No	Some application servers store the security context on the thread used to establish the JNDI connection (at the time of this release, only the WebLogic application server does this). In that case, the first activity to use this resource establishes the security context, then subsequent activities use the same security context, unless this field is checked. Checking this field ensures that each activity that uses this resource examines the security context to determine if the activity uses the same security context as the security context established on the thread. If they are different, the activity's configured security context is used.

Field	Global Var?	Description
		Checking this field causes additional overhead for activities that use this resource. Therefore, only check this field when necessary.
Optional JNDI Properties	No	The table in this field contains optional properties to pass to the JNDI server. Use the +, X, and arrow keys to add, delete, and move properties in the list. Each property requires the property name, the datatype for the property, and the value for the property. For more information about properties that can be passed to the JNDI server, see the documentation for your JNDI provider

IBM WebSphere and JNDI Context Caching

The IBM WebSphere application server provides a mechanism for caching the JNDI context object to improve the performance of JNDI lookup operations. ActiveMatrix BusinessWorks also caches the JNDI context object when resources such as EJB activities, JMS activities, JDBC activities, or JTA UserTransactions use JNDI. ActiveMatrix BusinessWorks performs best when using its own caching mechanism. Therefore, you must disable the IBM WebSphere caching mechanism when using IBM WebSphere with ActiveMatrix BusinessWorks. For a complete description of its caching features, see the IBM WebSphere documentation .

You can disable the IBM WebSphere JNDI context caching feature by performing the following procedure:

Procedure

1. Configure a JNDI Configuration resource.
2. On the Advanced tab, click the + button next to the table in the Optional JNDI Properties field to add a property.
3. In the Name field of the property, enter `com.ibm.websphere.naming.jndicache.cacheobject`.
4. Leave the Type field as the default value of string.
5. In the Value field, enter none.

6. Click **Apply** to accept the changes.
7. Use the JNDI Configuration resource in an EJB Connection, JMS Connection, or JTA UserTransaction shared configuration resource.

Job Shared Variable

Shared Configuration



A Job Shared Variable shared configuration resource allows you to store data for use by each process instance. A copy of the variable is created for each new process instance. This resource is useful for passing data to and from sub-processes without creating an input or output schema for the called process.

You can use the [Get Shared Variable](#) and [Set Shared Variable](#) activities to access the data instead of mapping data to a called processes input or output schemas. New process instances receive a copy of the variable, so data cannot be shared across process instances. Therefore, if a called process has the Spawn configuration field checked, a new process instance is created and the process instance receives a new copy of the job shared variable.

For more information about shared variables, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.

Schema

The Schema tab allows you to define a custom schema to hold the data of the variable. You can define your own datatype on this tab, and you can reference XML schema

stored in the project. For more information about how to define a schema, see [Specifying Data Schema](#).

Initial Value

The Initial Value tab allows you to specify an initial value for the variable. You can choose one of the following:

- **None** – specifies that no initial value is set for the shared variable. You must use the [Set Shared Variable](#) activity to set a value before you can read the value of the variable with the [Get Shared Variable](#) activity.
- **Select Value** – causes the Select Value field to appear. This field allows you to choose a stored XML Instance resource containing data that matches the schema specified on the Schema tab.
- **Build Value** – causes the Build Value field to appear. The **Edit** button brings up a smaller version of the XML Instance configuration. In this dialog you can construct an XML document that contains the initial value of the shared variable.

For more information on creating XML Instance resources, see *TIBCO Designer™ Palette Reference*.

Label

Graphic Tool



The Label resource allows you to create a generic label so that you can provide documentation and comments in your process definition. The Label is not an activity, because you cannot draw transitions to or from it, and it does not perform any action. You use the label to create a descriptive tag that you can place anywhere in a process definition.

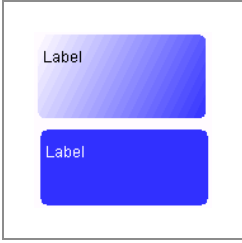
Labels can also be resized to cover an area. You can apply a border to the area by selecting a value in the Border Type and Border Thickness fields. You can also apply a background color to the area by selecting a color in the Background color field. This allows you to annotate your process definition and provide colors and borders to visually group related activities in your process definition.

The text of the label remains in the top left of the label area, but you can edit the text either in the design panel or by editing the Description field.

Configuration

The Configuration tab has the following fields.

Field	Description
Description	Descriptive text that you would like to add to the label. You can add comments here about the process diagram. Use this field to provide documentation for the process.
Label Color	Allows you to pick a color for the text and border of the label. Use the Chose Color button to bring up a dialog to select a color.
Label Font	Allows you to pick a font for the text of the label. Use the Choose Font button to bring up the Font Panel dialog.
Background Color	Allows you to pick a background color for the area the label covers. Use the Chose Color button to bring up a dialog to select a color.
Border Type	Allows you to select the type of border (either a dashed line or a solid line) for the label.
Border Thickness	Allows you to specify the thickness of the border line. 0 indicates no border.
Gradient Background Color	Specifies whether the background color changes from lighter to darker. When this check box is cleared, the background is a solid color. The following illustrates the difference between a gradient and solid background color. The top label uses a gradient background color and the bottom label uses a solid background color.

Field	Description
	 By default, the background color is gradient.

Lock Object

Shared Configuration



Lock Object shared configuration resources are used by Critical Section groups to ensure that only one process instance executes the activities in a critical section group at a time.

The Lock Object resource can be used to synchronize process instances from more than one process definition in the same process engine, or it can be used to synchronize process instances across multiple process engines.

For more information about Critical Section groups and Lock Object resources, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

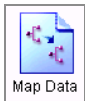
The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.

Field	Global Var?	Description
Multi-Engine	Yes	When checked, this field specifies to use the resource to synchronize process instances across more than one process engine. If you are using this resource to synchronize shared variable access, make sure that the Lock Object and Shared Variable resources have the same setting for this field. When synchronization is performed across process engines, a database transaction is performed to ensure proper execution order. Therefore, you must configure your process engine to use a database for storage, if you want to synchronize Critical Section groups across process engines. For more information about specifying a database for storing process engine information, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i>.

Mapper

Activity



The **Mapper** activity adds a new process variable to the process definition. This variable can be a simple datatype, a TIBCO ActiveEnterprise schema, an XML schema, or a complex structure. You can map data values from the current list of process variables to the elements of the variable added with the **Mapper** activity.

Input Editor

The Input Editor tab defines the structure of the process variable to add to the process definition.

You can use a simple datatype, or you can define a group of data elements on this tab. You can also reference XML schema or ActiveEnterprise classes stored in the project. Once defined, the data specified on the Input Editor tab becomes the input and output schema of the **Mapper** activity. This data then becomes available to other activities in the process definition.

For more information about using the Input Editor tab, see [Specifying Data Schema](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
subject	string	The subject to send the request on. The default subject is displayed on the Advanced tab, but you can override the default by specifying a subject here.
requestTimeout	integer	The time to wait (in milliseconds) for the synchronous request-reply operation to complete. An error is returned if the operation does not complete in this time limit.
replySubject	string	The Subject/Destination that is used to send the response when using the RV/JMS transport.
businessKey	string	This string is appended to the AE tracking info applInfo field. The AE tracking info for messages published is a concatenation of bw + <i><processName></i> + <i><jobID></i> + <i><businessKey></i>. Note: It is a good idea to have all publishers in a process use the same business key. Only the most recent publisher's business key is used in the tracking info for the message.
OtherProperties	complex	This element is only available when JMS Topic or JMS Queue is the transport and a JMS Application Properties resource is specified on the Transport tab. This element contains the message properties for the message. For more information about message

Input Item	Datatype	Description
		properties, see JMS Properties . Only properties applicable to this type of message are displayed.
body	complex	The input parameters for the specified operation. The input parameters are defined by the operation class for the specified Adapter Service for this activity.

Output

The output for the activity is defined by the specified data elements on the Input Editor tab.

Notify

Activity



The **Notify** activity allows a process instance to send data to a corresponding process instance containing a [Wait](#) activity or [Receive Notification](#) process starter. The Notify Configuration resource specified on the Configuration tab and the key specified on the Input tab create the relationship between the **Notify** activity and the corresponding Wait or Receive Notification. The data specified in the Notify's input is sent to the Wait or Receive Notification that specifies the same Notify Configuration resource and has the same value for the key.

The Wait, Receive Notification, and Notify activities allow running process instances to communicate. For more information on inter-process communication, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Shared Configuration	The Notify Configuration shared configuration resource that defines a data schema for this activity. For more information, see Notify Configuration. The Notify Configuration resource is required, but the schema can be empty if you do not want to pass data between processes.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
key	string	The key that coordinates a Notify activity with the corresponding Wait or Receive Notification. For example, in an order handling process, the order ID might be the key that Wait and Notify use to determine which Wait activity corresponds to the Notify activity. The key can be an XPath expression, but it must evaluate to a string. For more information and examples of choosing the key for Wait, Notify, and Receive Notification, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
timeout	integer	The timeout (number of milliseconds) to keep the information for this Notify activity. The activity executes immediately and transitions to the next activity, but the Notify information is kept by the engine until a corresponding Receive Notification or

Input Item	Datatype	Description
		Wait occurs. If the corresponding activity does not occur before this timeout is reached, the Notify information is removed.
schema	complex	The schema specified by the Shared Configuration on the Configuration tab. This schema is used to pass data from the process instance into the process instance containing a corresponding Receive Notification or Wait.

Output

This activity produces no output.

Notify Configuration

Shared Configuration



The Notify Configuration resource specifies a schema to use for passing data between executing process instances. Corresponding [Receive Notification](#), [Notify](#), and [Wait](#) activities use the same Notify Configuration resource to define the data for inter-process communication. The schema can be empty, if you do not want to pass data between processes. For more information about inter-process communication, see *TIBCO ActiveMatrix BusinessWorks™ process Design*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the resource.
Description	Short description of the shared resource.

Field	Description
Local Only	When checked, this field specifies to perform the notification in memory only. When unchecked, a database is used to perform the notification.

Data

The Data tab allows you to define a custom schema. The schema can be empty, if you do not want to pass data between processes.

You can define your own datatype on this tab, and you can reference XML schema or ActiveEnterprise classes stored in the project. Once defined, the data specified on the Data tab appears on the Input or Output tab of the [Receive Notification](#), [Wait](#), or [Notify](#) activity where this shared configuration resource is used.

For more information about how to define a schema, see [Specifying Data Schema](#).

Null

Activity



The Null activity is an activity with no action performed. This activity has a name and a description specified on the Configuration tab, but there is no input or output for the activity.

This activity is useful if you want to join multiple process flows. That is, when you have multiple transitions out of an activity and each transition takes a different path in the process definition, you can create a transition from the activity at the end of each path to a Null activity to resume a single flow of execution in the process.

On Event Timeout

Process Starter



The On Event Timeout process starter specifies a process to execute when a Wait For activity discards an incoming event due to a timeout. A Wait For

activity's event timeout is specified by the Event Timeout field on the Event tab of the activity.

The default behavior for an event timeout is to confirm and then discard the event. You can override the default behavior by creating a process definition using the On Event Timeout process starter.

You can specify an On Event Timeout process definition for a specific event source (that is, a specific Wait For activity). You can specify one or more On Event Timeout process definitions for specific event sources, but you should not create more than one process definition for the same event source. Once a Wait For activity experiences a timeout, that timeout can only apply to one On Event Timeout process definition.

You can also specify that an On Event Timeout process definition applies to any event source that experiences a timeout. If there is no On Event Timeout specified for a particular event source, the process engine calls the On Event Timeout process definition whose Any Event Source field is checked.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the resource.
Description	Short description of the process starter.
Any Event Source	Specifies that this process starter should execute when any event source experiences an event timeout, and there is no On Event Timeout process definition specified for the event source.
Event Source	Only appears when the Any Event Source field is unchecked. This field allows you to specify for which event source this process should execute when there there is an event timeout. Only one On Event Timeout process definition can correspond to any event timeout. That is, you cannot create multiple process definitions and use the same Wait For activity the Event Source field.

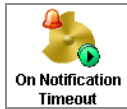
Output

The output for the activity is the following.

Output Item	Datatype	Description
originatingProcess	string	The name of the process definition containing the activity that had the event timeout.
originatingActivity	string	The name of the activity that had an event timeout.
<event>	complex	The schema of the discarded event. If the Any Event Source field is checked on the Configuration tab, this appears as an element of type any. If a specific activity is supplied in the Event Source field, the output schema of that activity is provided.

On Notification Timeout

Process Starter



The **On Notification Timeout** process starter specifies a process to execute when a timeout is reached for storing notification data for a **Notify** activity. For more information about the **Notify** activity and how to specify a timeout for storing notification information, see [Notify](#).

You can specify an On Notify Timeout process definition for a specific Notify Configuration. The process definition with the On Notification Timeout process starter executes when a **Notify** activity with a matching Notify Configuration experiences a timeout. You can specify one or more On Notify Timeout process definitions for specific Notify Configurations, but you should not create more than one process definition for the same Notify Configuration. Once a **Notify** activity experiences a timeout of its stored notification information, that timeout can only apply to one On Notification Timeout process definition.

You can also specify that an On Notification Timeout process definition applies to any **Notify** activity that experiences a timeout. If there is no On Notification Timeout specified for the Notify Configuration of the **Notify** activity, the process engine calls the On Notification Timeout process definition that has the **Any Notify Config** field checked.

i Note: Notify timeouts cause the notify information to be marked for removal, but the information is removed at regular intervals. The default interval for checking Notify timeouts is 60 seconds, and therefore, the OnNotificationTimeout process starter only starts processes after each interval. If you want to reduce the interval, you can do so by altering the Engine.WaitNotify.SweepInterval custom engine property to the desired number of seconds. However, as you decrease the number of seconds in the interval you incur further engine overhead.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the resource.
Description	Short description of the process starter.
Any Notify Config	Specifies that this process starter should execute when any Notify activity experiences a timeout, and there is no On Notify Timeout process definition specified for the Notify Configuration for the activity. When this field is checked, the Notify Configuration field becomes disabled. Uncheck this field when you want to specify a process definition for a specific Notify Configuration.
Notify Configuration	Only appears when the Any Notify Config field is unchecked. This field allows you to specify the Notify Configuration that this process applies to. Only timeouts for Notify activities with a matching Notify Configuration start a process instance. Only one On Notification Timeout process definition can correspond to any Notify Configuration. That is, you cannot create multiple process definitions and use the same Notify Configuration.

Output

The output for the activity is the following.

Output Item	Datatype	Description
notificationData	Object	This element contains the schema specified in the Notify Configuration of the Notify activity that has timed out.

On Shutdown

Process Starter



The On Shutdown process starter specifies a process to execute when the process engine shuts down, after all process instances are executed.

The On Shutdown process starter can be useful for specifying any post-processing that must be done before shutting down the engine. For example, your application may back up a database table then purge the data. Also, you may want to send email to administrators notifying them that the process engine is shutting down.

Typically, you create one process definition that uses the On Shutdown process starter and place all post-processing activities into that process definition. However, you can have more than one process definition that uses the On Shutdown process starter. All process definitions with the On Shutdown process starter execute in no specific order when the process engine shuts down.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the resource.
Description	Short description of the process starter.

On Startup

Process Starter



The On Startup process starter specifies a process to execute when the process engine starts, before any incoming events are processed. In the event that the process engine is restarting and attempting to recover checkpointed process instances, the On Startup process definition must complete its execution before any recovered process instances execute.

The On Startup process starter can be useful to specify any pre-processing that must be done before regular processing starts. For example, your application may check that the required database tables exist and create them necessary. Also, you may want to send email to administrators notifying them that the process engine is starting.

Typically, you create one process definition that uses the On Startup process starter and place all pre-processing activities into that process definition. However, you can have more than one process definition that uses the On Startup process starter. All process definitions with the On Startup process starter execute in no specific order when the process engine starts.



Note: All On Startup process definitions must complete successfully before normal processing can begin.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the resource.
Description	Short description of the process starter.

Receive Notification

Activity



The **Receive Notification** process starter starts a process when another process executes a **Notify** activity with a matching key and Notify Configuration resource. The key specified in the Key field of the Configuration tab creates a relationship between the Receive Notification process starter and the corresponding **Notify** activity.

The same Notify Configuration shared configuration resource must be specified by corresponding Receive Notification and Notify activities so that data can be passed from the process containing the **Notify** activity to the process started with Receive Notification. The schema in the Notify Configuration resource can be empty, if you do not want to pass data between processes.

The Wait, Receive Notification, and Notify activities allow running process instances to communicate. For more information on inter-process communication, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Shared	The Notify Configuration shared configuration resource that defines a

Field	Description
Configuration	data schema for this activity. For more information, see Notify Configuration. The Notify Configuration resource is required, but the schema can be empty if you do not want to pass data between processes.
Key	The key that coordinates a Receive Notification process starter with the corresponding Notify activity. For example, in an order handling process, the order ID might be the key that Receive Notification uses to determine which Notify activity corresponds to this process starter. The key must be a fixed string. You may use global variables, but XPath expressions cannot be used. For more information and examples of choosing the key for Wait, Notify, and Receive Notification, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output

The output for the activity is the following.

Output Item	Datatype	Description
schema	complex	The schema specified by the Shared Configuration field on the Configuration tab. This schema is used to pass data from the process instance containing the Notify activity into this process instance.

Rethrow

Activity



The **Rethrow** activity throws the exception caught by the [Catch](#) activity again. Use this activity when you want to propagate the exception to the next level. For more information about using the **Rethrow** activity, see the description of error handling in *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.

Input

This activity has no Input.

Output

This activity has no Output.

Set Shared Variable

Activity



The **Set Shared Variable** activity allows you to change the value of a shared variable. If you are using this activity to set the value of a Shared Variable resource, you may want to use a critical section group to ensure that no other process instances are altering the value of the shared variable at the same time.

For more information about shared variables and critical section groups, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Variable Configuration	No	The Shared Variable or Job Shared Variable whose value you would like to change.
Output New Value	No	When checked, the new value of the shared variable is placed into this activity's output. The value is then available to subsequent activities in the process definition. If not checked, this activity has no output.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
<schema>	varies	The schema of the shared variable specified in the Variable Configuration field on the Configuration tab is the input schema for this activity.

Output

The output for the activity is the following.

Output Item	Datatype	Description
<schema> or no output	varies	When the Output New Value field on the Configuration tab is unchecked, this activity has no output. When the field is checked, the new value of the shared variable is output. The schema for the output is determined by the schema specified for the shared variable.

Shared Variable

Shared Configuration



A Shared Variable resource allows you to share data across process instances. All process instances can read and update the data stored in a shared variable.

This resource is useful if you want to marshal data across process instances or if you want to make a common set of information available to all process instances. For

example, you may have a set of approval codes for incoming orders that change daily for security purposes. You can create a shared variable to hold the approval codes and create one process definition for setting the codes. You can then retrieve the shared variable in all processes that require the current approval codes.

For more information about shared variables, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.
Persistent	Yes	The current value of the shared variable is stored in memory for fast access by process instances. However, if a process engine crashes, the current state of the shared variable may be lost. To preserve the current state of a Shared Variable resource, you can check this field to store the current value of the shared variable in the process engine's storage location. The current state of the shared variable is only updated in the process engine's persistent storage when the value of the variable changes.
Multi-Engine	Yes	Checking this field specifies that you want to make the value of a Shared Variable resource available to process instances across multiple process engines. If you choose this option, a database must be used to store process engine data. Also, only engines that are in the same deployment and use the same database to store process information can share variables. For more information on specifying a database for process engine storage during deployment, see <i>TIBCO ActiveMatrix BusinessWorks™</i>

Field	Global Var?	Description
		<i>Administration.</i> For more information on specifying a database for process engine storage during testing, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>. Because multiple process engines access the shared variable when this field is checked, the current value of the variable is not stored in memory. Both retrieving the current value of the variable and assigning a new value to the variable requires I/O to the process engine's database storage. If this field is not checked, each process engine operates on its own set of shared variable resources. Process instances on one engine cannot retrieve the value of the shared variable set by a different process engine, when this field is unchecked.

Schema

The Schema tab allows you to define a custom schema to hold the data of the shared variable. You can define your own datatype on this tab, and you can reference an XML schema stored in the project. For more information about how to define a schema, see [Specifying Data Schema](#).

Initial Value

The Initial Value tab allows you to specify an initial value for the shared variable. You can choose one of the following:

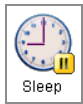
- None – specifies that no initial value is set for the shared variable. You must use the [Set Shared Variable](#) activity to set a value before you can read the value of the variable with the [Get Shared Variable](#) activity.
- Select Value – causes the Select Value field to appear. This field allows you to choose a stored XML Instance resource containing data that matches the schema specified on the Schema tab.
- Build Value – causes the Build Value field to appear. The **Edit** button brings up a

smaller version of the XML Instance configuration. In this dialog you can construct an XML document that contains the initial value of the shared variable.

For more information on creating XML Instance resources, see *TIBCO Designer™ palette Reference*.

Sleep

Activity



The **Sleep** activity suspends the process on the current transition for the given amount of time. If you have multiple control flows in your process, only the current execution branch of the process is suspended.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
IntervalInMillisec	integer	The amount of time, in milliseconds, to suspend the current branch of the process.

Timer

Process Starter



The Timer process starter starts a process at a specific time. You can also specify that processes are to be started periodically.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Start Time	Yes	The day and time to start the process. If the process is to be run periodically, then the start time indicates the first time to run the process. The local machine's time zone is used to determine when the process should start. If the project is run on machines in different time zones, the time zone of the machine where the project was saved is used. Use a global variable in this field to supply start times relative to the time zone where the project is deployed. If you supply a global variable for this field, the format of the specified time must be the following: <pre>yyyy MMM dd HH:mm:ss z</pre> For example, "2004 Apr 28 20:35:11 EST". The z at the end of the date format is the name of the time zone. For a list of time zone names, see Time Zone Names.

Field	Global Var?	Description
Run once	No	Indicates this process should be run only once at the day and time indicated by the Start Time field. If unchecked, the Time Interval and Interval Unit fields appear to allow you to specify the frequency of the process.
Time Interval	Yes	Integer indicating the number of units specified in the Interval Unit field. For example, if a value of 1 is specified and the Interval Unit field is specified as Hour, a new process is started every hour beginning at the time specified in the Start Time field. If a value of 2 is specified and the Interval Unit field is specified as Month, a process is started every other month, beginning at the time specified in the Start Time field. If the value specified is less than one second (1000 milliseconds), the timer ignores the specified interval and is triggered once every second. Note: If you use a global variable for this field, the global variable specifies the time interval in milliseconds and the Interval Unit field is not displayed.
Interval Unit	No	Unit of time to use with the Time Interval field to determine how often to start a new process. The units can be: Millisecond, Second, Minute, Hour, Day, Week, Month, Year.

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate

Field	Description
	to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i> .
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the \$_processContext process variable.

Output

The output for the activity is the following.

Output Item	Datatype	Description
TimerOutputSchema		The time the process instance started. The time is represented by several items in the output schema indicating the hour, minute, second, and so on. Each item uses the appropriate datatype.

Wait

Activity



The **Wait** activity suspends execution of the process instance and waits for a **Notify** activity with a matching key to be executed in another process instance. The key specified in the Notify Configuration resource specified on the Configuration tab and the Key field of the Input tab creates a relationship between the **Wait** activity and the corresponding **Notify** activity.

The same Notify Configuration shared configuration resource must be specified by corresponding Wait and Notify activities so that data can be passed from the process instance containing the **Notify** activity to this process instance. The schema in the Notify

Configuration resource can be empty, if you do not want to pass data between processes.

The Wait, Receive Notification, and Notify activities allow running process instances to communicate. For more information about inter-process communication, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Shared Configuration	The Notify Configuration shared configuration resource that defines a data schema for this activity. For more information, see Notify Configuration. The Notify Configuration resource is required, but the schema can be empty if you do not want to pass data between processes.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
key	string	The key that coordinates a Wait activity with the corresponding Notify . For example, in an order handling process, the order ID might be the key that Wait and Notify use to determine which

Input Item	Datatype	Description
		Wait activity corresponds to the Notify activity. The key can be an XPath expression, but it must evaluate to a string. For more information and examples of choosing the key for Wait, Notify, and Receive, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
timeout	integer	The timeout (number of milliseconds) for this Wait activity. If a Notify activity that has a matching key is not executed in this specified number of seconds, the Wait times out and transitions to the next activity. If no value or if zero is specified in this field, this activity waits indefinitely.

Output

The output for the activity is the following.

Output Item	Datatype	Description
schema	complex	The schema specified by the Shared Configuration on the Configuration tab. This schema is used to pass data from the process instance containing the Notify into this process instance.

Write To Log

Activity



This activity writes a message to the log. There is one log file for each process engine.

Depending upon whether you are using a process engine in testing mode or a deployed process engine, the logs are stored in different locations. During test mode, the logs are stored in the ActiveMatrix BusinessWorks working directory under the logs subdirectory. For more information about specifying the working directory, see *TIBCO Designer™ User's Guide*. For deployed process engines, the log location is specified by custom engine properties. For more information about custom engine properties, see *TIBCO ActiveMatrix BusinessWorks™ Administration*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Role	The user-defined role name for this log entry. Roles can be used to filter the log entries when displaying them. For example, if you set the role as Debug, you can choose to view or not view all log entries that have the role set to Debug. You can enable or disable system or user-defined roles using custom engine properties at deployment time. For more information, see Custom Properties for the Write To Log Activity.
Suppress Job Info	If checked, no additional information is added to the log entry. If unchecked, each log message has the Job number, process definition name, and activity name prepended to the message text.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
message	string	The message that you would like to appear in the log. Logs are stored in the ActiveMatrix BusinessWorks installation directory under the logs subdirectory. Note: When the message contains non-ASCII data, the default encoding of the Java Virtual Machine used by the ActiveMatrix BusinessWorks process engine is used to encode the text when writing to the log file.
msgCode	string	The error code of the message. This is used as an identifier for the message so that applications can scan the log for the message.

Output

This activity produces no output.

Custom Properties for the Write To Log Activity

In some situations, you may want to specify which messages are sent to the log file. You can use custom engine properties to specify which roles are enabled or disabled for log messages. You may configure the following properties:

- `Trace.<roleName>.Log` – Setting this property to false disables writing messages for the specified role name to the log.
- `Trace.<roleName>.*` – Setting this property to false disables all messages for the specified role. The specified role can be one of the following system roles: ERROR, WARN, DEBUG, or INFO.
- `Trace.Role.<roleName>` – Setting this property to false disables all messages for the specified user-defined role. Setting `Trace.Role.*` to false disables all messages to all user-defined roles.
- `Trace.Role.*` – Setting this property to false disables all messages for all user-defined roles.

You can also specify the character encoding of the log file by using the `log.file.encoding` property.

For more information about setting custom engine properties, see *TIBCO ActiveMatrix BusinessWorks™ Administration*.

OnError

Activity



This activity provides error handling mechanism for all the errors that happen outside the job boundaries of all processes associated with the service resource.

You are recommended to create only one process definition with this event source for the whole project. If you create multiple processes, then all of them get triggered and may not be in sequence.

Whenever an error occurs for messages triggered through Service Resource outside of Job boundaries, the Error handler process gets triggered with the appropriate data.

This error handler process can be used only for Logging purposes. It does not affect The actual Service MEP.

Note: The Error handler handles the failure in Pre-process, Post-process (failure while sending the response), and Fault Sending scenarios. It does not catch the error during the execution process.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.

Misc

The Misc tab has the following fields.

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom ID	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output

The Output-tab of the onError event source contains the following schema:

- `errorCode` - Contains the error code useful for filtering the errors.
- `errorDesc` - Contains description of errors.
- `errorStack` (optional) - Contains stack of error logs.
- `msgData` - Contains the Soap-Envelope of the message.
- `MessageHeaders` (optional) - Contains either JMS or HTTP transport headers.
This depends on the transport for which the error is captured.
- `mimeEnvelopeElement` (optional) - Contains mime data.

HTTP Palette

The HTTP palette allows you to send and receive HTTP requests.

HTTP Connection

Shared Configuration



The HTTP Connection resource describes the characteristics of the connection used to receive incoming HTTP requests. This resource is used when ActiveMatrix BusinessWorks expects to receive an HTTP request on a specific port where a process engine is running. For example, the process starters [HTTP Receiver](#) and [SOAP Event Source](#) and the signal-in activity [Wait for HTTP Request](#) receive HTTP requests.

Note: There can be at most one process with an HTTP Receiver or Wait for HTTP Request that uses the same HTTP Connection resource. This restriction allows the HTTP server listening for incoming requests to dispatch the request to the correct process. There can be more than one SOAP Event Source that uses the same HTTP Connection. Also, SOAP Event Source and HTTP activities can use the same HTTP Connection resource and the HTTP server correctly dispatches the incoming request to the correct process.

Two types of servers are available for the HTTP Connection resource: Tomcat and HTTPComponent.

Tomcat has a synchronous request response paradigm and can be used in scenarios where high throughput is important. To achieve a good throughput with Tomcat, the maxprocessor value should be almost equal to the number of concurrent requests the server can handle. Note that increasing the maxprocessor count also increases the memory footprint.

HTTPComponent is a light-weight and scalable server based on NIO which can be useful in scenarios where handling thousands of requests in a resource efficient manner is more important than the throughput. HTTPComponent server gives a consistent throughput for any number of concurrent requests with little or no increase in its worker thread (maxprocessor thread).

Note: You can choose the server type at runtime by setting the global property `bw.plugin.http.server.serverType` in the file `bwengine.tra`. Using this property, you can use the HTTPComponent server at run-time for older projects without having to modify the server type at design-time. The value set for this property in the `bwengine.tra` file overrides the value that is set in design-time.

The HTTP Connection resource can specify that the HTTPS (secure sockets layer or SSL) protocol must be used by clients. If this is enabled, you can configure the SSL parameters for the HTTP server using the **Configure SSL Button**. For more information, see [Configure SSL Button](#).

Warning: If you have multiple HTTP Connection resources specified by multiple HTTP Receiver process starters, the HTTP servers require that all of the connections must be valid to initialize all HTTP Receivers. Therefore, make certain that all HTTP Connection resources have valid configurations before testing or deploying the project.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.
Host	Yes	Specifies the name of the host that accepts the incoming requests. For machines that have only one network card, the default value <code>localhost</code> specifies the current machine. For machines that have more than one network card, this field specifies the host name of the card that can be used to accept incoming HTTP requests. If there is more than one network card on the machine, and you specify <code>localhost</code> in this field, all network cards on the

Field	Global Var?	Description
		machine can listen for incoming HTTP requests on the specified port. Note: Only one HTTP server can be started on each port. Therefore, if you have a machine with multiple network cards, make certain that all HTTP Connection resources that use the same host name specify different port numbers. Note: Note: When the hostname is localhost, ActiveMatrix BusinessWorks considers the machine as a non multi home environment. Hence it is not required to set the <code>bw.plugin.http.server.defaultHost</code> property in <code>bwengine.tra</code> file. However, when the hostname is anything other than the localhost, then ActiveMatrix BusinessWorks considers the machine as a multi home environment. Set the <code>bw.plugin.http.server.defaultHost</code> property in <code>bwengine.tra</code> file to the same value as has been set in the host field of HTTP Shared Connection for default host.
Port	Yes	Port number on which to listen for incoming HTTP requests.
SSL	No	Specifies whether incoming requests must use the HTTPS (secure socket layer or SSL) protocol. This protocol authenticates the server to the client, and optionally authenticates the client to the server. Enabling this field allows you to specify SSL parameters with the Configure SSL button (see Configure SSL Button).
Server Type	No	Specifies the server type to be used. The drop-down lists the available options: Tomcat and HTTPComponent.

Configure SSL Button

The **Configure SSL** button allows you to specify the SSL parameters for the HTTP connection. The following are the fields in the SSL Configuration for HTTPS Connections dialog:

i Note: The HTTPComponent server type does not support Entrust based SSL. Set property `java.property.TIBCO_SECURITY_VENDOR` to `j2se` in the `bwengine.tra` file for HTTP SSL to work with HTTPComponent server type.

Field	Description
Requires Client Authentication	Checking this field requires clients to present their digital certificate before connecting to the HTTP server. When this field is checked, the Trusted Certificates Folder becomes enabled so that you can specify a location containing the list of trusted certificate authorities.
Trusted Certificates Folder	This field is only applicable when the Requires Client Authentication field is checked. This field specifies a folder in the project containing one or more certificates from trusted certificate authorities. This folder is checked when a client connects to ensure that the client is trusted. This prevents connections from rogue clients.
Identity	This is an Identity resource that contains the HTTP server's digital certificate and private key. For more information, see <i>TIBCO Designer™ Palette Reference</i>.
Strong Cipher Suites Only	When checked, this field specifies that the minimum strength of the cipher suites used can be specified with the <code>bw.plugin.security.strongcipher.minstrength</code> custom engine property. For more information about this property, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i>. The default value of the property disables cipher suites with an effective key length below 128 bits. When this field is unchecked, only cipher suites with an effective key length of up to 128 bits can be used.

Advanced

The Advanced tab has the following fields

Field	Global Var?	Description
Enable DNS Lookups	No	Checking this field enables a Domain Name System (DNS) lookup for HTTP clients so that the IP address is resolved to a DNS name. This can adversely affect throughput. Uncheck this field by default and check this field only when required.
Connection Properties		
maxPostSize	Yes	This property is available when the Server Type Tomcat is selected. Specifies the maximum size in bytes of the POST that the container FORM URL parameter parsing can handle. By default, the value of this field is set to 2M or 2097152 bytes. Setting the value less than or equal to zero disables the limit for maxPostSize.
maxSavePostSize	Yes	This property is available when the Server Type Tomcat is selected. Specifies the maximum size in bytes of the POST that the container can save/buffer during FORM or CLIENT-CERT authentication. During FORM authentication, the request message is saved while the user is redirected to the login page. It is retained until the user authentication succeeds or the session associated with the authentication request expires. During CLIENT-CERT authentication, the

Field	Global Var?	Description
		request message is buffered for the entire duration of the SSL handshake. Once the request is processed, the buffer is emptied. However, in both the cases the buffering happens before the user is authenticated. By default, the value of this field is set to 4K or 4096 bytes. Setting the value to -1 disables the limit for this field.
URI Encoding	Yes	Specifies the character encoding used to decode the URI bytes. If you do not set any value for uriEncoding, the default encoding set in HTTP Receiver is used. Specifying this property automatically sets the Tomcat property useBodyEncodingForURI to false.
acceptCount	Yes	This property is available when the Server Type Tomcat is selected. Specifies the maximum number of incoming connection requests that can be accepted when all HTTP processors are in use. The default value for acceptCount is 100. Incoming requests received after the acceptCount limit is reached are rejected. If the value is not set in TIBCO Designer, you can set the value for either design-time or run-time. Specify the value in a .config file. Provide the file name and location while starting TIBCO Designer to set the value of acceptCount in design-time

Field	Global Var?	Description
		Specify the value in bwengine.tra to set the value in run-time.
compressableMimeType	Yes	Specifies the list of MIME types for which HTTP compression may be used. The default value for this comma separated list is text/html, text/xml, text/plain.
compression	Yes	Specifies if the output of the HTTP Connection is compressed using HTTP/1.1 GZIP compression. The acceptable values for this field are - off: By default, the compression is disabled. on: Allows compression for text data. force: Forces compression in all cases. Note that if the content length is not known and compression is enabled, the output still be compressed. <i>any numerical integer value:</i> Any numerical value equivalent to "on", and specifies the minimum amount of data before the output can be compressed.
connectionTimeOut	Yes	Specifies the number of milliseconds that the Connector waits for the request URI line to be presented after the connection is accepted. The default value is 60000 that is, 60 seconds.

Custom Properties for the HTTP Palette

In some situations, you may want to alter the configuration of the HTTP server that receives incoming HTTP requests for ActiveMatrix BusinessWorks. You can configure the following custom properties:

Property	Tomcat or HTTPComponent	Description
bw.plugin.http.server.minProcessors	Tomcat	The minimum number of threads available for incoming HTTP requests. The HTTP server creates the number of threads specified by this parameter when it starts up. The default value of this property is 10.
bw.plugin.http.server.maxProcessors	Tomcat	The maximum number of threads available for incoming HTTP requests. The HTTP server does not create more than the number of threads specified by this parameter. The default value of this property is 75.
bw.plugin.http.server.maxSpareProcessors	Tomcat	The maximum number of unused request processing threads that can exist until the thread pool starts stopping the unnecessary threads. The default maximum number of spare threads is 50.

Property	Tomcat or HTTPComponent	Description
<code>bw.plugin.http.server.acceptCount</code>	Tomcat	The maximum queue size for incoming requests. Incoming requests that are not handled by available threads are placed on the queue until they can be processed. If the queue is full, new incoming requests are refused with an error. The default value of this property is 100.
<code>bw.plugin.http.server.restrictIPAddresses</code>	Both	A comma-separated list of regular expression patterns that is compared with the remote client's IP address before accepting or rejecting requests from the client. The remote IP address of the client must not match for any request from this client to be accepted.
<code>bw.plugin.http.server.serverType</code>	Both	The type of server to be used for the HTTP connection resource. Two

Property	Tomcat or HTTPComponent	Description
		server types are available: Tomcat and HTTP Component. The default value of this property is Tomcat.
bw.plugin.http.server.httpcomponents.workerThread	HTTPComponent	The maximum number of web server threads available to handle HTTP requests for the HTTPComponents server type. The default value of this property is 50.
bw.plugin.http.server.restrictHttpMethods	Tomcat	Specific HTTP methods can be disabled. By default, none of the HTTP methods are restricted by the server. You can specify a comma-separated list of methods that are to be restricted. These restrictions are then applicable to all resources accessed on this server, for all roles. You cannot disable methods selectively for a

Property	Tomcat or HTTPComponent	Description
		particular service or for a particular server.

i Note: If the same IP address is allowed and restricted, the restriction overrides the allowance and restricts the IP address.

If you have a large number of incoming requests, you may want to change the values of these properties to handle more incoming requests concurrently.

For more information about setting custom engine properties, see *TIBCO ActiveMatrix BusinessWorks™ Administration*.

i Note: When a client sends a request that cannot be processed because no threads are available, ActiveMatrix BusinessWorks returns a ConnectionRefused exception to the client. The HTTP Receiver uses the minProcessors/maxProcessors properties to control the flow of incoming HTTP requests. If you set the Flow Limit deployment property for a process definition with the HTTP Receiver process starter, maxProcessors is set to <flowLimitValue>-1 and minProcessors is set to <maxProcessorValue>/2. Therefore, the Flow Limit value can not be reached because the maxProcessors property prevents new requests from being accepted before the Flow Limit value is reached.

HTTP Receiver

Process Starter



HTTP Receiver

Starts a process based on the receipt of a HTTP request.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
HTTP Connection	No	The HTTP Connection resource describes the characteristics of the connection used to receive incoming HTTP requests. For more information about specifying shared resources, see HTTP Connection.
Output Style	No	Type of output. Can be either String or Binary.
Parse Post Method Data	No	Specifies to parse the message body of the HTTP request into a schema for the activity's output. When this check box is selected, the parameters specified in the Parameters field are used to validate the incoming request and parse it into the output schema. Note: In the case of an incoming GET request, there is no message body, so this field has no effect on incoming GET requests.
Parameters	No	The parameters of the incoming HTTP request. Specifying parameters in this table allows the incoming request to be parsed and represented as a schema in this activity's output. For each parameter, you must provide a name, datatype, and whether the field is required, optional, or repeating. Use the + and x buttons to the right of the parameter table to add and remove parameters. Use the up and down

Field	Global Var?	Description
		arrows to move parameters to new positions in the table. Note: Specifying parameters in this field parses the query string of the request into the output schema, regardless of whether the Parse Post Method Data field is checked. Checking the Parse Post Method Data field and specifying parameters here is required to parse the message body along with the query string of the request.
HTTP Authentication	No	Specifies that the client sending the HTTP request must be authenticated. The user name and password specified in the incoming request must exist in the domain. Users are created and managed in the domain using TIBCO Administrator.
Expose Security Context	No	Places the information from the user's security context (either authentication or SSL certificate information) into the Context/SecurityContext output element. This is a fairly expensive operation and SSL certificates can consume memory resources, so check this option only if you require information from the user's security context for later use in your process definition.
Default Encoding	Yes	Specifies the encoding to use if no charset is specified in the Content-Type header of the message. This encoding is used for the message body and URL. However, when URLEncoding is specified in the HTTP Connection shared resource, the Default Encoding parameter specified here applies to the contentType alone and not to the URLEncoding property.
Upgrade Configuration	No	This field is available only when projects created in ActiveMatrix BusinessWorks 5.2.x or earlier versions are migrated to a higher version.

Field	Global Var?	Description
		Selecting the check box upgrades the schema in the input/output tab for the activity. You can revert back to the old schema by clearing the check box. However, if you create new processes in the older projects, the activities always show the new schema in their input/output tabs. In this case, it is not possible to revert to the old schema.

Advanced

The Advanced tab contains the following fields:

Field	Global Var?	Description
Write to File	No	Checking this field specifies to write incoming requests that exceed the specified threshold size to a file instead of storing the request in memory. This allows you to accept large incoming requests without consuming a great deal of memory. When this field is checked, the Directory and Threshold Size fields appear. Note: This option is not intended to be used with the Parse Post Method Data option on the Configuration tab. When Write to File is specified, the PostData output element becomes a choice element containing either the output FileName or the PostData depending upon whether the data exceeds the size specified in the Threshold Size field. It is recommended to use either the Write to File option or the Parse Post Method data option, but not both at the same time. Leaving this field unchecked specifies to keep incoming requests in memory.

Field	Global Var?	Description
Note: Once written, the files created by using this option are not deleted automatically. You must manage the storage used by these files and delete them when they are no longer used.		
Directory	Yes	Create Non-Existing Directories
Threshold Size (bytes)	Yes	The maximum size (in bytes) of an incoming request that can be kept in memory. Requests larger than the specified size are written to a file in the specified directory. The file's name is output so that subsequent activities in the process definition can access the file and read its contents. Specifying zero (0) in this field causes all incoming requests to be saved to a file.

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Special Characters in HTTP Requests

Depending upon the content type of the data for the request, the request can contain URL-encoded data and the server is expected to decode the data. For example, an incoming request may have the following message body:

```
name=John%20Smith&address=500%201%2F2%20Main%20Street
```

If the message body is not parsed (Parse Post Method Data is unchecked on the Configuration tab), the body of the message remains URL-encoded in the PostData output element. If the message body is parsed, the data is decoded before being placed into the PostData output element, and that element would appear as follows:

```
name=John Smith&address=500 1/2 Main Street
```



Note: The order of the parameters may not be the same in the PostData output element as the order in the original HTTP request sent by the client when the message body is parsed.

Because the PostData output element can contain different data and be a different length depending upon whether the message body is parsed, it is recommended that you obtain data from the parameters output element when the message body is parsed.

For more information about the URL specification, see <http://www.rfc-editor.org/rfc/rfc1738.txt>.

Output Headers

The Output Headers tab describes the data structure for the headers of the HTTP request. You can use the default structure, or you can alter the structure, if the incoming request has a specific data structure for the header of the request. This tab uses the mechanism described in [Specifying Data Schema](#) to specify the data structure for the headers. For more information about creating a customized data structure, see that appendix.

Header structure is defined by the HTTP protocol. For more information about the fields and content of the header of a HTTP request, see the HTTP Protocol specification. You can obtain this specification at www.w3.org.

The default header fields are the following.

Header	Datatype	Description
Accept	string	This field specifies media types that are acceptable for response messages for the incoming request. For example, <code>text/*,text/html</code>. Media types are described in the HTTP specification. If no Accept header field is present, then it is assumed that the client accepts all media types.
Accept-Charset	string	This field specifies the character sets that are acceptable for response messages for the incoming request. For example, <code>iso-8859-5, unicode-1-1</code>. Character sets are described in the HTTP specification. If no Accept-Charset header is present, then it is assumed that the client accepts any character set.
Accept-Encoding	string	This field specifies the content-coding values that are acceptable for response messages. For example, <code>compress, gzip</code>. For more information about this header field, see the HTTP specification .
Content-Type	string	This field indicates the media type of the entity body sent to the receiver. For example, <code>text/html; charset=ISO-8850-4</code>. Media types are described in the HTTP specification.
Content-Length	string	This field indicates the size of the entity body (in decimal number of OCTETs) sent to the receiver. This field takes into account the encoding of the message body. For more information on when the message body is URL encoded, see Special Characters in HTTP Requests.
Connection	string	This field allows the requestor to specify options desired for this connection. For example, the option <code>close</code> specifies that the requestor would like the connection to be closed when the request is complete.

Header	Datatype	Description
Cookie	string	For more information about this field, see the HTTP specification. If you want to receive more than one cookie, set the Cardinality for this field to Repeating (*).
Pragma	string	This field is used to include implementation-specific directives that might apply to the receiver. For more information about using this field, see the HTTP specification.

Output

The output for the activity is the following.

Output Item	Datatype	Description
Method	string	The method specified in the request. For example, GET or POST.
RequestURI	string	The address portion of the request. This is the portion before the question mark (?).
HTTPVersion	string	Version field of the HTTP request.
PostData or BinaryContent or FileName	string	The message body of the HTTP request. The content of this element depends upon whether the message body is parsed. Parsing behavior is controlled by the Parse Post Method Data field on the Configuration tab. For more information, see Special Characters in HTTP Requests. If the Write to File field is checked on the Advanced tab, this field can also contain the file name of the file containing the message, if the incoming message

Output Item	Datatype	Description
		exceeds the specified threshold size.
Command	string	Command portion of the HTTP request.
QueryString	string	The query string portion of the request. This is the portion after the question mark (?).
Header	string	The header of the HTTP request.
DynamicHeaders	complex	The dynamic header is an additional header parameter to add runtime headers to the Outgoing Http Messages. This element is specified in the Input tab. Dynamic header consists of the following information: • Name – name of the header. • Value – value of the header. The following overriding conditions can be considered: • Overrides the value of HeaderName with the value found in DynamicHeaders if it is a non-repeating header. If more than one occurrence of this header is found under DynamicHeaders, it throws the following exception: <pre>"The header [headerName] is defined as non-Repeating Header in Input Headers. This header cannot have multiple occurrences in DynamicHeaders."</pre> • If it is repeating element, add the respective name value pairs under dynamic headers, which can be added to the existing list maintained for this element. • For a repeating element, if the new header name is not found under Headers section declared via TIBCO Designer, following is

Output Item	Datatype	Description
		added into the HTTP Headers. ◦ Shows only one value, if found once in DynamicHeaders. ◦ Shows array of values, if found repeating in DynamicHeaders.
Protocol	string	Can be either HTTP or HTTPS depending upon the protocol used by the request.
Port	string	Port number on which the request was received. This is the port number configured in the HTTP Connection resource.
mimeEnvelope Element	complex	This element contains the message attachments. This element contains a repeating element named mimePart that contains each mime attachment.
mimeHeaders	complex	This element contains the mime header for each mimePart. Mime headers can contain the following information: • content-disposition – To suggest a filename for an attachment, use "*,filename=<filename>" in this element. Note: HTTP servers may alter or choose to ignore the suggested name. • content-type • content-transfer-encoding • content-id • other mime header information For more information about MIME headers and their

Output Item	Datatype	Description
		syntax, see http://www.faqs.org/rfcs/rfc2045.html. Note: When the content type is specified as "text/*" (for example, "text/xml"), the attachment content is expected to be in either the textContent input element or the file name storing the attachment is expected to be in the fileName input element. When the content type is anything other than "text/*", the attachment content is expected to be in either the binaryContent input element or the file name storing the attachment is expected to be in the fileName input element.
binaryContent textContent fileName	choice	This element contains the mime attachment. The element can be one of the following: • binaryContent – content of the attachment when the attachment is binary data. • textContent – content of the attachment when the attachment is text data. • fileName – the file name of the attachment written to the disk.
Headers	complex	The fields of the header specified on the Output Headers tab. For more information about these items, see the description of the Output Headers tab.
parameters	complex	Parameters for the incoming request. These parameters are configured in the Parameters field on the Configuration tab. When the Parse Post Method Data field on the Configuration tab is checked, the parameters include the parameters in the query string and message body of the incoming HTTP request. When the Parse Post Method Data field is unchecked, the parameters only include the query

Output Item	Datatype	Description
		string of the incoming request.
Context	complex	Contains information about the client's environment. For more information, see the description of the following elements contained in the Context element.
RemoteAddress	String	The IP address of the client that submitted the HTTP request.
Context	complex	This element holds the optional SecurityContext element and the RemoteAddress and RemoteHost elements described below.
SecurityContext	complex	This output element is only available when the Expose Security Context field is checked on the Configuration tab. This element contains information about the requestor's authentication or certificate information, depending upon the security scheme used. For more information, see the description of the following elements contained in the SecurityContext element.
Username PasswordToken	complex	The username and password of the HTTP client that made the request. The elements in this element are populated only when the client uses HTTP authentication.
CertificateToken	complex	This element contains the SSL certificate and certain useful information taken from the certificate, such as the CipherSuite, the certificate issuer, and so on. If the client has more than one certificate in a certificate chain, all certificates are contained in the repeating Certificate element.
RemoteAddress	String	IP address of the remote host that sent the HTTP request.

Output Item	Datatype	Description
RemoteHost	String	Host name of the remote host that sent the HTTP request. This element is only available when the Enable DNS Lookups field is checked on the Advanced tab of the HTTP Connection used by this activity.

Proxy Configuration

Shared Configuration



The Proxy Configuration resource is used to specify a proxy HTTP server when HTTP requests are sent outside of a firewall.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.
Proxy Host	Yes	Host name or IP address of the proxy server.
Proxy Port	Yes	Port number of the proxy host.
Proxy Authentication	No	Specifies what authentication should be used . An Authentication can be of the following three types: • NONE - is default. • BASIC - to continue with the current implementation of using Username Password credentials.The

Field	Global Var?	Description
		Identity field appears when this field is selected. NTLM - Identity and Domain fields appear when NTLM is selected. Identity is a reference of the Username and Password combination.
Identity	No	This is an Identity resource that contains the username and password for connecting to the proxy server. For more information, see <i>TIBCO Designer™ Palette Reference</i>.
Domain	No	Specifies the domain to be used for Authentication. The user name and password specified in the Identity field must exist in the domain.

Send HTTP Request

Activity



The **Send HTTP Request** activity sends a HTTP request to a web server.

This activity can send a request to a server that complies with HTTP 1.1 specification.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.

Field	Description
Description	Short description of the activity.
Host	The host machine name or IP address to send the request to. For example, www.tibco.com.
Port	The port on the host machine to send the request to. The default port is 80.
Use Proxy Setting	Specifies to use a proxy server to gain access outside of a firewall. The Proxy Configuration shared configuration resource specifies the configuration of the proxy server. For more information, see Proxy Configuration .
Accept Redirects	Checking this field indicates that the request should be automatically redirected when the HTTP server sends the redirection status code (302) in response to this request and the remote host redirects the request to the same host and port. If the remote host redirects the request to a different host or a different port on the same host, the request is not automatically redirected. The maximum number of redirections is 100. When this field is unchecked, the request is not redirected.
Parameters	The parameters of the HTTP request. For each parameter, you must provide a name, datatype (must be type string), and whether the parameter is required, optional, or repeating. These parameters are specified in the parameters element on the Input tab. For more information about this field, see Sending Data in the HTTP Request.
HTTP Authentication	Specifies what authentication should be used . An authentication can be of the following three types: • NONE - is default. • BASIC - to continue with the current implementation of using

Field	Description
	username and password credentials. The Identity field appears when this field is enabled. • NTLM - Identity and Domain fields appear when NTLM is selected. Identity is a reference of the Username and Password combination. • Kerberos - Identity, Krb Conf and Login Conf fields appear when Kerberos is selected. This includes information describing the default Kerberos realm, and the location of the Kerberos key distribution centers for known realms.
Identity	This is an Identity resource that contains the client's username and password. This identity is used to perform only Basic, NTLM authentication and Kerberos authentication. When using Kerberos authentication, the identity configuration type must be of type username and password only. For more information, see TIBCO Designer™ Palette Reference.
Domain	Specifies the domain to be used for Authentication. The username and password specified in the Identity must exist in the domain.
SSL	When the Use SSL? check box is selected, this specifies to use the HTTPS (secure socket layer, or SSL) for the request. This protocol authenticates the server to the client, and optionally, the server can require that the client authenticate itself to the server. The Configure SSL button becomes enabled when this field is checked. For more information, see Configure SSL Button.
Upgrade Configuration	This field is visible only when projects created in ActiveMatrix BusinessWorks 5.2.x or earlier versions are migrated to a higher version. Selecting the check box upgrades the schema in the input/output tab for the activity. You can revert back to the old schema by clearing the check box.

Field	Description
	However, if you create new processes in the older projects, the activities always show the new schema in their input/output tabs. In this case, it is not possible to revert to the old schema.
Krb Conf	This field is visible only when Kerberos is specified in the HTTP Authentication field. Provide the Krb conf file from the disk or from the network. The krb5.conf file contains Kerberos configuration information, including the locations of KDCs and admin servers for the Kerberos realms of interest, defaults for the current realm and for Kerberos applications, and mappings of hostnames to Kerberos realms.
Login Conf	This field is visible only when Kerberos is specified in the HTTP Authentication field. Login configuration file entry consists of a name followed by one or more LoginModule-specific items. The name used for specifying LoginModule-specific items is "com.sun.security.jgss.login". Each LoginModule-specific item specifies a LoginModule, a flag value, and options to be passed to the LoginModule. Use the option, refreshKrb5Config=true for LoginModule. Example: com.sun.security.jgss.login { com.sun.security.auth.module.Krb5LoginModule required client=TRUE doNotPrompt=false refreshKrb5Config=true useTicketCache=true; };

Sending Data in the HTTP Request

There are several HTTP methods that can be used in an HTTP request. Each method sends data in the request in a different manner. For example, the GET method uses the query string of the request URI to pass parameter/value pairs. Other methods use the HTTP message body to send data in the request.

The **Send HTTP Request** activity has three input elements for sending data in a request:

- **PostData** – corresponds to the body of the HTTP message. All methods except the GET method accept data in this element.
- **QueryString** – corresponds to the query string of the request URI. You can use this input element to dynamically construct the query string using an XPath expression when you do not know the names or number of input parameters for the request until the activity executes.
- **Parameters** – corresponds to parameters defined in the Parameters field on the Configuration tab. This is useful if you have a fixed set of parameters that you send with the request. For requests that use the GET method, these parameters are passed as the query string of the request URI. For requests that use the POST method, these parameters are usually sent as the body of the HTTP message, but they can also be included in the query string.

For some methods, these input elements are mutually exclusive. For example, for POST requests, you can specify parameters on the Configuration tab and in the parameters input element or you can specify a PostData input element. However, you should not specify both input elements. In the case of a POST request, the PostData input element is ignored when parameters are specified on the Configuration tab.

For GET requests, you can specify parameters on the Configuration tab and in the parameters input element or you can specify a QueryString input element. Typically, if you know the list of parameters for the request, you should configure the parameters on the Configuration tab. If the list of parameters is not known until the activity executes, you should use the QueryString element. However, when all parameters on the Configuration tab are specified as Optional, you can use the QueryString input element instead of the parameters input element (but if any elements in the parameters element contain an expression, the QueryString element is ignored).

Special Characters in HTTP Requests

Depending upon the content type of the data for the request, the request can contain URL-encoded data and the server is expected to decode the data. If this is the case and you want to send special characters such as +, /, or = in your HTTP request, your data string must be URL-encoded if you send the data using the PostData or QueryString input elements. If you send the data using the parameters specified on the Configuration tab, encoding is done automatically.

For example, you want to specify the following PostData:

```
name=John Smith&address=500 1/2 Main Street
```

Your postData input element should result in the following string:

```
name=John%20Smith&address=500%201%2F2%20Main%20Street
```

For more information about the URL specification, see <http://www.rfc-editor.org/rfc/rfc1738.txt>.

Configure SSL Button

The **Configure SSL** button allows you to specify the SSL parameters for the HTTP request. The following are the fields in the SSL Configuration for HTTPS Client Requests dialog:

Field	Description
Trusted Certificates Folder	Folder in the project containing one or more certificates from trusted certificate authorities. This folder is checked when a client connects to the HTTP server to ensure that the server is trusted. This prevents connections to rogue servers.
Identity	This is an Identity resource that contains the client's digital certificate and private key. For more information, see <i>TIBCO Designer™ Palette Reference</i>.
Verify Host Name	This field specifies to check the host name of the HTTP server against the host name listed in the server's digital certificate. This provides additional verification that the host name you believe you are connecting to is in fact the desired host. If the host name specified in the Host field on the Configuration tab is not an exact match to the host name specified in the server's digital certificate, the connection is refused.

Field	Description
Note: If you specify an equivalent hostname (for example, an IP address) in the Host field, but the name is not an exact match of the hostname in the host's digital certificate, the connection is refused.	
Strong Cipher Suites Only	When checked, this field specifies that the minimum strength of the cipher suites used can be specified with the <code>bw.plugin.security.strongcipher.minstrength</code> custom engine property. For more information about this property, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i>. The default value of the property disables cipher suites with an effective key length below 128 bits. When this field is unchecked, only cipher suites with an effective key length of up to 128 bits can be used.



Note: Server Name Indication (SNI) is available only when the security provider is J2SE and is enabled by default with this provider. Set the property `jsse.enableSNIExtension` to false to disable the SNI extension.

Advanced

The Advanced tab contains the following fields:

Field	Global Var?	Description
Write to File	No	Checking this field specifies to write incoming requests that exceed the specified threshold size to a file instead of storing the request in memory. This allows you to accept large incoming requests without consuming a great deal of memory. When this field is checked, the Directory and Threshold Size fields appear.

Field	Global Var?	Description
		Note: This option is not intended to be used with the Parse Post Method Data option on the Configuration tab. When Write to File is specified, the PostData output element becomes a choice element containing either the output FileName or the PostData depending upon whether the data exceeds the size specified in the Threshold Size field. It is recommended to use either the Write to File option or the Parse Post Method data option, but not both at the same time. Leaving this field unchecked specifies to keep incoming requests in memory. Note: Once written, the files created by using this option are not deleted automatically. You must manage the storage used by these files and delete them when they are no longer used.
Directory	Yes	Create Non-Existing Directories
Threshold Size (bytes)	Yes	The maximum size (in bytes) of an incoming request that can be kept in memory. Requests larger than the specified size are written to a file in the specified directory. The file's name is output so that subsequent activities in the process definition can access the file and read its contents. Specifying zero (0) in this field causes all incoming requests to be saved to a file.

Input Headers/Output Headers

The Input Headers and Output Headers tabs describe the data structure for the headers of the HTTP request and the HTTP reply message. You can use the default structure, or you can alter the structure, if the outgoing request or the reply to the request has a specific data structure for the header. This tab uses the same mechanism described [Specifying Data Schema](#) to specify the data structure for the headers. For more information about creating a customized data structure, see that section.

Header structure is defined by the HTTP protocol. For more information about the fields and content of the header of a HTTP request, see the HTTP Protocol specification. You can obtain this specification at www.w3.org.

The default header fields are the following.

Header	Datatype	Description
Accept (input header)	string	This field specifies media types that are acceptable for response messages for the request. For example, <code>text/*,text/html</code>. Media types are described in the HTTP specification. If no Accept header field is specified, then the server assumes that all media types are acceptable.
Accept-Charset (input header)	string	This field specifies the character sets that are acceptable for response messages for the request. For example, <code>iso-8859-5, unicode-1-1</code>. Character sets are described in the HTTP specification. If no Accept-Charset header is specified, then the server assumes that any character set is acceptable.
Accept-Encoding (input header)	string	This field specifies the content-coding values that are acceptable for response messages. For example, <code>compress, gzip</code>. For more information about this header field, see the HTTP specification.
Content-Type (input/output header)	string	This field indicates the media type of the entity body for the outgoing message and the incoming response. Media types are described in the HTTP specification. An example of the media type is <code>text/html; charset=ISO-8850-4</code>.
Content-Length (output header)	string	This field indicates the size of the entity body (in decimal number of OCTETs) of the response message.
Content-Encoding (output header)	string	This field is used as a modifier to the content-type. When present, its value indicates what additional content encodings have been applied to the entity-body, and thus

Header	Datatype	Description
header)		what decoding mechanisms must be applied in order to obtain the media-type referenced by the Content-Type header field. Content-Encoding is primarily used to allow a document to be compressed without losing the identity of its underlying media type. For more information about this field, see the HTTP specification.
Date (output header)	string	The date and time when the response message was sent.
Server (output header)	string	The server sending the original response message.
Location (output header)	string	This field is used to redirect the receiver to a location other than the Request-URI for completion of the request or for identification of a new resource.
Connection	string	This field allows the requestor to specify options desired for this connection. For example, the option close specifies that the requestor would like the connection to be closed when the request is complete.
Set-Cookie (output header)	string	For more information about this field, see the HTTP specification. If you want to receive more than one cookie, set the Cardinality for this field to Repeating (*).
Cookie (input header)	string	A name/value pair (also known as a cookie) containing information that the HTTP server may be expecting. You can set the Cardinality field for this element to Repeating (*) to specify more than one cookie. You can also specify multiple name/value pairs in a single non-repeating element by separating each pair with a comma (for example, "name1=value1, name2=value2"). If you set the custom engine property

Header	Datatype	Description
		bw.plugin.http.protocol.single-cookie-header to true, multiple name/value pairs are sent as a single header element. For more information about custom engine properties, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i> .
Pragma (input/output header)	string	This field is used to include implementation-specific directives that might apply to the receiver. For more information about using this field, see the HTTP specification.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Host	string	The HTTP host you want to connect to. Specifying a value for this input item overrides any value specified on the Configuration tab.
Port	integer	The port number that the HTTP server uses for incoming requests. Specifying a value for this input item overrides any value specified on the Configuration tab.
Method	string	The HTTP method to use for the request. All HTTP 1.1 methods are supported, but the CONNECT method is unnecessary because ActiveMatrix BusinessWorks automatically uses the connect method when connecting by way of a proxy server. If no method is specified in this element, the

Input Item	Datatype	Description
		GET method is used by default.
RequestURI	string	The address portion of the request. This is the portion of the URI before the question mark (?).
PostData	string	The message body of the HTTP request. Do not specify this element when the method of the request is GET. For more information about this input element, see Sending Data in the HTTP Request.
QueryString	string	The query string portion of the request. This is the portion after the question mark (?). For more information about this input element, see Sending Data in the HTTP Request.
Timeout	integer	The amount of time (in milliseconds) to wait for a response from the HTTP server.
Headers	complex	The header fields to send for the request. This element is specified on the Input Headers tab, and you can use this input item to supply values for the header when sending the request.
DynamicHeaders	complex	The dynamic header is an additional header parameter to add runtime headers to the Outgoing Http Messages. This element is specified in the Input tab. Dynamic header consists of the following information: • Name – name of the header. • Value – value of the header. The following overriding conditions can be considered:

Input Item	Datatype	Description
		- Overrides the value of HeaderName with the value found in DynamicHeaders if it is a non-repeating header. If more than one occurrence of this header is found under DynamicHeaders, it throws the following exception: "The header [headerName] is defined as non-Repeating Header in Input Headers. This header cannot have multiple occurrences in DynamicHeaders." - If it is repeating element, add the respective name value pairs under dynamic headers, which can be added to the existing list maintained for this element. - For a repeating element, if the new header name is not found under Headers section declared via TIBCO Designer, following is added into the HTTP Headers. - Shows only one value, if found once in DynamicHeaders. - Shows array of values, if found repeating in DynamicHeaders.
parameters	complex	Parameters for the incoming request. These parameters are configured in the Parameters field on the Configuration tab. This element is only available when there are items specified in the Parameters field. For more information about this input element, see Sending Data in the HTTP Request.
mimeEnvelope	complex	This element contains the message

Input Item	Datatype	Description
Element		attachments. This element contains a repeating element named mimePart that contains each mime attachment. Note: Only HTTP POST method can send messages with attachments.
mimeHeaders	complex	This element contains the mime header for each mimePart. Mime headers can contain the following information: • content-disposition – To suggest a filename for an attachment, use <code>"*;filename=<filename>"</code> in this element. Note: HTTP servers may alter or choose to ignore the suggested name. • content-type • content-transfer-encoding • content-id • other mime header information For more information about MIME headers and their syntax, see http://www.faqs.org/rfcs/rfc2045.html.

Input Item	Datatype	Description
Note: When the content type is specified as "text/*" (for example, "text/xml"), the attachment content is expected to be in either the textContent input element or the file name storing the attachment is expected to be in the fileName input element. When the content type is anything other than "text/*", the attachment content is expected to be in either the binaryContent input element or the file name storing the attachment is expected to be in the fileName input element.		
binaryContent textContent fileName	choice	This element contains the mime attachment. The element can be one of the following: • binaryContent – content of the attachment when the attachment is binary data. • textConetnt – content of the attachment when the attachment is text data. • fileName – the file name of the attachment written to the disk.

Output

The output for the activity is the following.

Output Item	Datatype	Description
Header	string	The header of the HTTP request.
statusLine	complex	This field is the first line of the response message. This consists of the protocol version, a numeric status code, and the text phrase explaining the status code. For more information about status codes in HTTP

Output Item	Datatype	Description
		responses, see the HTTP specification.
binaryContent	binary	The binary content of the response to the request from the HTTP server.
asciiContent	string	The ASCII content of the response to the request from the HTTP server.
Headers	complex	The header fields of the reply. The structure of this output item is specified on the Output Headers tab.
DynamicHeaders	complex	The dynamic header is an additional header parameter to add runtime headers to the Outgoing Http Messages. This element is specified in the Input tab. Dynamic header consists of the following information: • Name – name of the header. • Value – value of the header. The following overriding conditions can be considered: • Overrides the value of HeaderName with the value found in DynamicHeaders if it is a non-repeating header. If more than one occurrence of this header is found under DynamicHeaders, it throws the following exception: "The header [headerName] is defined as non-Repeating Header in Input Headers. This header cannot have multiple occurrences in DynamicHeaders." • If it is repeating element, add the respective name value pairs under dynamic headers, which can be added to the existing list maintained for this element. • For a repeating element, if the new header

Output Item	Datatype	Description
		name is not found under Headers section declared via TIBCO Designer, following is added into the HTTP Headers. ◦ Shows only one value, if found once in DynamicHeaders. ◦ Shows array of values, if found repeating in DynamicHeaders.
mimeEnvelope Element	complex	This element contains the message attachments. This element contains a repeating element named mimePart that contains each mime attachment.
mimeHeaders	complex	This element contains the mime header for each mimePart. Mime headers can contain the following information: • content-disposition – To suggest a filename for an attachment, use ";filename=<filename>" in this element. Note: HTTP servers may alter or choose to ignore the suggested name. • content-type • content-transfer-encoding • content-id • other mime header information For more information about MIME headers and their syntax, see http://www.faqs.org/rfcs/rfc2045.html.

Output Item	Datatype	Description
Note: When the content type is specified as "text/*" (for example, "text/xml"), the attachment content is expected to be in either the textContent input element or the file name storing the attachment is expected to be in the fileName input element. When the content type is anything other than "text/*", the attachment content is expected to be in either the binaryContent input element or the file name storing the attachment is expected to be in the fileName input element.		
binaryContent textContent fileName	choice	This element contains the mime attachment. The element can be one of the following: • binaryContent – content of the attachment when the attachment is binary data. • textConetnt – content of the attachment when the attachment is text data. • fileName – the file name of the attachment.

Persistent Connections

A **Send HTTP Request** activity requires a connection to the HTTP server. The activity exclusively uses the connection until the HTTP server sends the response message. If you have many process instances connecting to a HTTP server, each Send HTTP Request opens a connection, holds the connection, and then closes the connection when the activity completes. Opening and closing a large number of connections causes a significant overhead. Persistent connections play a major role here. Persistent connections maintain a pool of connections that can be reused by Send HTTP Request activities so that each activity does not need to open and close the connection. Once the connection is released by the process instance, it is returned to the pool maintained.



Note: Not all HTTP servers support the use of persistent connections. To determine if the HTTP Server supports the use of persistent connections, read the documentation of that HTTP Server.

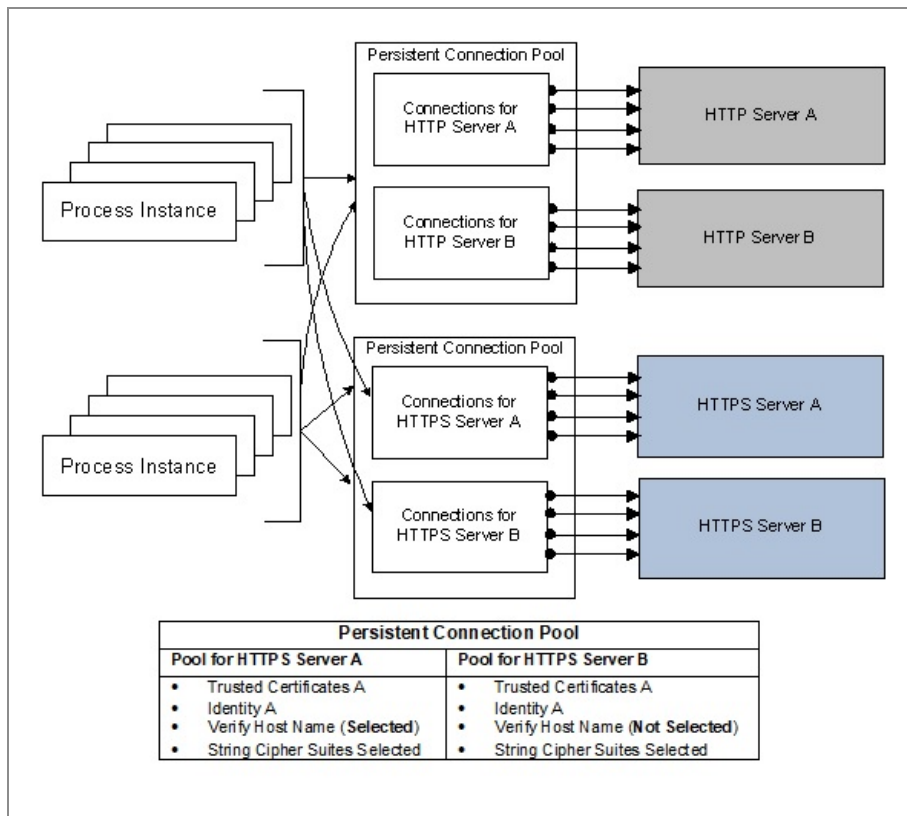
Persistent HTTP Connection illustrates the persistent connection pool.

Persistent Connection Manager allows you to specify the total number of connections as well as the maximum number of connections per HTTP Server. The total number of connections is a total of connections to all HTTP Servers handled by that connection pool.

For each SSL configuration, a different Persistent Connection pool is created and maintained. All connections using the same SSL Configuration, irrespective of the HTTPS Server they are connecting to, is maintained by this pool.

When a **Send HTTP Request** activity requires a connection, the pool is requested for a connection that corresponds to the HTTP server. If an idle connection for that HTTP Server is found, it is used. If no idle connection is found, the pool tries to create one depending on whether the maximum connections limit for that HTTP Server has been reached. If the maximum connections limit is reached, the request waits for a connection to be released back to the pool. If the maximum connections limit is not reached, the pool creates a connection to use.

Figure 10: Persistent HTTP Connection



Persistent connections are managed by custom engine properties. For more information about setting custom engine properties, see *TIBCO ActiveMatrix BusinessWorks™ Administration*.

The following section describe the engine properties that control persistent connections.

bw.plugin.http.client.usePersistentConnectionManager

This property specifies that a pool of HTTP connections to each HTTP server must be created for the connections to be reused by the Send HTTP Request activities. This property enables persistent connections for all non-SSL Send HTTP Request in that ActiveMatrix BusinessWorks engine.

When this property is set to `true`, a pool of connections is created for each HTTP server that Send HTTP Request connects to. The total number of connections in the pool is limited by the `bw.plugin.http.client.maxTotalConnections` property. The number of connections for each host (each HTTP server) is limited by the `bw.plugin.http.client.maxConnectionsPerHost` property.

The default value of this property is `false`.

bw.plugin.http.client.usePersistentConnectionManagerForSSL

This property specifies that a pool of HTTPS connections to each HTTP server must be created for connections to be reused by the Send HTTP Request activities. This property enables persistent connections for all SSL Send HTTP Request in that ActiveMatrix BusinessWorks engine.

When this property is set to `true`, a pool of connections is created for each HTTPS server that Send HTTP Request connects to. The total number of connections in the pool is limited by the `bw.plugin.http.client.maxTotalConnections` property. The number of connections for each host is limited by the `bw.plugin.http.client.maxConnectionsPerHost` property.

The default value of this property is `false`.

- i Note:** SSL Configuration - consists of Trusted Certificates folder, Identity, verifyHostName flag, and strongCipherSuites flag.
NTLM Authentication Configuration - consists of Identity and domainName.

Different connections pools get created for the following combinations:

HTTP

HTTP and NTLM

HTTPS (using SSL) and NTLM

HTTPS (using SSL)

All persistent connection properties like `maxTotalConnections`, `maxConnectionsPerHost`, idle connections check, and timeout value are applied on each connection pool.

- i Note:** To enable Persistent Connections for both HTTP and HTTPS, set `bw.plugin.http.client.usePersistentConnectionManager` and `bw.plugin.http.client.usePersistentConnectionManagerForSSL` properties to true.

- i Note:** Define the engine property `bw.plugin.http.client.authentication.preemptive` to False, if you don't need the preemptive authentication applied for the HTTP Requests. The default value of this property is True.

`bw.plugin.http.client.connectionTimeout`

The value of this property is ignored unless the `bw.plugin.http.client.usePersistentConnectionManager` OR `bw.plugin.http.client.usePersistentConnectionManagerForSSL` property is set to true. This property specifies that the timeout period (in milliseconds) for which persistent connections should be alive to each remote HTTP server.

The default value of this property is 3000 ms.

`bw.plugin.http.client.maxConnectionsPerHost`

The value of this property is ignored unless the `bw.plugin.http.client.usePersistentConnectionManager` OR `bw.plugin.http.client.usePersistentConnectionManagerForSSL` property is set to true. This property specifies the maximum number of persistent connections to each remote HTTP server.

The default value of this property is 20.

bw.plugin.http.client.maxTotalConnections

The value of this property is ignored unless the `bw.plugin.http.client.usePersistentConnectionManager` or `bw.plugin.http.client.usePersistentConnectionManagerForSSL` property is set to true. This property specifies the maximum number of persistent connections for all HTTP servers.

The default value of this property is 200.

bw.plugin.http.client.checkForStaleConnectionsValidationTime

The value of this property is ignored unless the `bw.plugin.http.client.usePersistentConnectionManager` or `bw.plugin.http.client.usePersistentConnectionManagerForSSL` property is set to true. For

persistent connections, the HTTPComponent Library 4.5.5 defines the period of inactivity in milliseconds after which persistent connections must be re-validated before being allocated to the consumer. This check helps to detect connections that have become stale when the connections were kept inactive in the pool.

A non-positive value passed to this method disables connection validation. The default value of this property is 2000 ms.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective actions to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
HttpClientException	The HTTP server replied with a message that has the 4XX status code.
HttpServerException	The HTTP server replied with a message that has the 5XX status code.

Exception	Thrown When...
HttpException	An HTTP exception occurred when trying to execute the specified method, or when trying to read the response.
ActivityTimeoutException	A timeout has been reached.

Send HTTP Response

Activity



Sends a response to a previously received HTTP request. This activity is used in conjunction with the HTTP Receiver process starter or the **Wait for HTTP Request** activity.

The default status line returned by this activity is "200 OK".

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Reply For	The Wait for HTTP Request activity or HTTP Receiver process starter that received the request. This is a selection list of available activities that can receive HTTP requests.
Flush Response	Allows you to specify if the response is to be flushed after each Send HTTP Response activity. By default, this property is not checked. This field is useful when there are a large number of Send HTTP

Field	Description
	Response activities in a job. In such cases, the Close Connection field is checked in the last activity to indicate the response is complete. However, checking the Flush Response field overrides this behavior and causes the response to be flushed after each Send HTTP Response activity.
Close Connection	Specifies that this activity contains the last part of an HTTP response. This field is used when you have more than one Send HTTP Response activity in a process definition, each activity sending a portion of the response to the client. All Send HTTP Response activities in a process definition should have this field unchecked, except the last Send HTTP Response activity. This field should be checked in the last Send HTTP Response activity to indicate the response is complete. If the entire response is sent by only one Send HTTP Response activity in the process definition, this field should be checked.

Input Headers

The Input Headers tab describes the data structure for the headers of the HTTP response. You can use the default structure, or you can alter the structure, if the outgoing response has a specific data structure for the header of the request. This tab uses the same mechanism described [Specifying Data Schema](#) to specify the data structure for the headers. For more information about creating a customized data structure, see that appendix.

Header structure is defined by the HTTP protocol. For more information about the fields and content of the header of a HTTP request, see the HTTP Protocol specification. You can obtain this specification at www.w3.org.

The default header fields are the following.

Header	Datatype	Description
StatusLine	string	This field is the first line of a response message. This consists of the protocol version, a numeric status code, and the text phrase explaining the status code. For more

Header	Datatype	Description
		information about status codes in HTTP responses, see the HTTP specification.
Content-Type	string	This field indicates the media type of the entity body sent to the receiver. Media types are described in the HTTP specification. An example of the media type is <code>text/html; charset=ISO-8850-4</code>. By default this item is set to <code>text/html</code>. If you are using this activity and the Retrieve Resources activity to retrieve a WSDL file, you should set the type to <code>text/xml</code> when specifying values on the Input tab.
Set-Cookie	string	For more information about this field, see the HTTP specification.
Pragma	string	This field is used to include implementation-specific directives that might apply to the receiver. For more information about using this field, see the HTTP specification.
Location	string	This field is used to redirect the receiver to a location other than the Request-URI for completion of the request or for identification of a new resource.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
binaryContent	binary	The binary content of the response to the request.

Input Item	Datatype	Description
asciiContent	string	The ASCII content of the response to the request.
Headers	complex	The fields of the header specified on the Input Headers tab. For more information about these items, see the description of the Input Headers tab. Note: Only the first Send HTTP Response activity in a process definition can set the Headers element. If your process definition contains multiple Send HTTP Response activities, the headers are set by the first activity and this element is ignored for all subsequent activities.
DynamicHeaders	complex	The dynamic header is an additional header parameter to add runtime headers to the Outgoing Http Messages. This element is specified in the Input tab. Dynamic header consists of the following information: • Name – name of the header. • Value – value of the header. Consider the following overriding conditions: • Overrides the value of the HeaderName with the value found in DynamicHeaders if it is a non-repeating header. If more than one occurrence of this header is found under DynamicHeaders, it throws the following exception: "The header [headerName] is defined as non-Repeating Header in Input Headers. This header cannot have multiple occurrences in DynamicHeaders." • If it is repeating element, add the respective name value pairs under dynamic headers, which can be added to the existing list maintained for this element.

Input Item	Datatype	Description
		- For a repeating element, if the new header name is not found under Headers section declared via TIBCO Designer, following is added into the HTTP Headers. - Shows only one value, if found once in DynamicHeaders. - Shows array of values, if found repeating in DynamicHeaders.
mimeEnvelope Element	complex	This element contains the message attachments. This element contains a repeating element named mimePart that contains each mime attachment.
mimeHeaders	complex	This element contains the mime header for each mimePart. Mime headers can contain the following information: - content-disposition – To suggest a filename for an attachment, use "*,filename=<filename>" in this element. Note: HTTP servers may alter or choose to ignore the suggested name. - content-type - content-transfer-encoding - content-id - other mime header information For more information about MIME headers and their syntax, see http://www.faqs.org/rfcs/rfc2045.html.

Input Item	Datatype	Description
Note: When the content type is specified as "text/*" (for example, "text/xml"), the attachment content is expected to be in either the textContent input element or the file name storing the attachment is expected to be in the fileName input element. When the content type is anything other than "text/*", the attachment content is expected to be in either the binaryContent input element or the file name storing the attachment is expected to be in the fileName input element.		
binaryContent textContent fileName	choice	This element contains the mime attachment. The element can be one of the following: • binaryContent – content of the attachment when the attachment is binary data. • textContent – content of the attachment when the attachment is text data. • fileName – the file name of the attachment written to the disk.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
HttpCommunicationException	An HTTP exception occurred when trying to send the response.

Wait for HTTP Request

Activity



Waits for an incoming HTTP request in a process definition. The process instance suspends until the incoming HTTP request is received.

For information on running the "Wait for" activities in a multi-engine mode, see [Scalability With Incoming Events](#) in *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
HTTP Connection	The HTTP Connection shared resource that describes the host name and the port number on which the process waits for the incoming message. For more information about specifying shared resources, see HTTP Connection.
Parse Post Method Data	Specifies that the message body of the HTTP request should be parsed into a schema for the activity's output. When this check box is selected, the parameters specified in the Parameters field are used to validate the incoming request and parse it into the output schema. Note: In the case of an incoming GET request, there is no message body, so this field has no effect on incoming GET requests.

Field	Description
Parameters	The parameters of the incoming HTTP request. Specifying parameters in this table allows the incoming request to be parsed and represented as a schema in this activity's output. For each parameter, you must provide a name, datatype, and whether the field is required, optional, or repeating. Use the + and x buttons to the right of the parameter table to add and remove parameters. Use the up and down arrows to move parameters to new positions in the table. Note: Specifying parameters in this field parses the query string of the request into the output schema, regardless of whether the Parse Post Method Data field is checked. Checking the Parse Post Method Data field and specifying parameters here is required to parse the message body along with the query string of the request.
HTTP Authentication	Specifies that the client sending the HTTP request must be authenticated. The user name and password specified in the incoming request must exist in the domain. Users are created and managed in the domain using TIBCO Administrator.

Event

The Event tab has the following fields.

Field	Description
Candidate Event Key	Expression built from the data of the incoming file change event. This expression should evaluate to a string and it is compared to the "key" field of the activity's input. If the Candidate Event Key and the activity's key match, then the process accepts the incoming message. For example, you may have sent an HTTP request earlier in the process, and you may be waiting for a response from the HTTP server. You may specify a requestID as an attribute of the HTTP request and responseTo attribute on the Wait for HTTP request activity. The HTTP server

Field	Description
	presumably set the <code>responseTo</code> attribute of its response to the same value as the <code>requestID</code> attribute of the request. You would then specify the <code>responseTo</code> attribute as the Candidate Event Key and the <code>requestID</code> attribute of the HTTP request as the "key" in the Input tab. This expression is specified in XPath, and only data from the incoming event is available for use in this XPath expression. For more information about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Event Timeout (msec)	A message may arrive before this activity is executed. This field specifies the amount of time (in milliseconds) a message waits if it is received before this activity is executed. If the event timeout expires, an error is logged and the event is discarded. If no value is specified in this field, the message waits indefinitely. If zero is specified, the event is discarded immediately, unless this activity has already been reached.

i Note: When building an expression in the Candidate Event Key field, only data from the incoming event is available. This is because you want to place an expression containing incoming event data in the Candidate Event Key field. When the results of this expression match the results of the expression in the "key" item on the Input tab, the **Wait For HTTP Request** activity proceeds.

Output Headers

The Output Headers tab describes the data structure for the headers of the HTTP request. You can use the default structure, or you can alter the structure, if the incoming request has a specific data structure for the header of the request. This tab uses the same mechanism described [Specifying Data Schema](#) in to specify the data structure for the headers. For more information about creating a customized data structure, see that section.

Header structure is defined by the HTTP protocol. For more information about the fields and content of the header of a HTTP request, see the HTTP Protocol specification. You can obtain this specification at www.w3.org.

The default header fields are the following.

Header	Datatype	Description
Accept	string	This field specifies media types that are acceptable for response messages for the incoming request. For example, <code>text/*,text/html</code>. Media types are described in the HTTP specification. If no Accept header field is present, then it is assumed that the client accepts all media types.
Accept-Charset	string	This field specifies the character sets that are acceptable for response messages for the incoming request. For example, <code>iso-8859-5, unicode-1-1</code>. Character sets are described in the HTTP specification. If no Accept-Charset header is present, then it is assumed that the client accepts any character set.
Accept-Encoding	string	This field specifies the content-coding values that are acceptable for response messages. For example, <code>compress, gzip</code>. For more information about this header field, see the HTTP specification.
Content-Type	string	This field indicates the media type of the entity body sent to the receiver. Media types are described in the HTTP specification. An example of the media type is <code>text/html; charset=ISO-8850-4</code>.
Content-Length	string	This field indicates the size of the entity body (in decimal number of OCTETs) sent to the receiver. This field takes into account the encoding of the message body, For more information on when the message body is URL encoded, see Special Characters in HTTP Requests.
Connection	string	This field allows the requestor to specify options desired for

Header	Datatype	Description
		this connection. For example, the option close specifies that the requestor would like the connection to be closed when the request is complete.
Cookie	string	For more information about this field, see the HTTP specification.
Pragma	string	This field is used to include implementation-specific directives that might apply to the receiver. For more information about using this field, see the HTTP specification.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
key	string	The value that is compared to the Candidate Event Key field of the Event tab to determine if this message applies to this process. For an example of using the Candidate Event Key field with the key input item, see Event .
processTimeout	integer	The time (in milliseconds) to wait for the incoming message. An error is returned if the message is not received in this time limit.

Output

The output for the activity is the following.

Output Item	Datatype	Description
Method	string	The HTTP method to use for the request. All HTTP 1.1 methods are supported.
RequestURI	string	The address portion of the request. This is the portion before the question mark (?).
HTTPVersion	string	Version field of the HTTP request.
PostData or BinaryContent	string	The message body of the HTTP request. The content of this element depends upon whether the message body is parsed. Parsing behavior is controlled by the Parse Post Method Data field on the Configuration tab). For more information, see Special Characters in HTTP Requests.
Command	string	Command portion of the HTTP request.
QueryString	string	The query string portion of the request. This is the portion after the question mark (?).
Header	string	The header of the HTTP request.
DynamicHeaders	complex	The dynamic header is an additional header parameter to add runtime headers to the Outgoing Http Messages. This element is specified in the Input tab. Dynamic header consists of the following information: • Name – name of the header. • Value – value of the header. The following overriding conditions can be considered: • Overrides the value of HeaderName with the value found in DynamicHeaders if it is a non-

Output Item	Datatype	Description
		repeating header. If more than one occurrence of this header is found under DynamicHeaders, it throws the following exception: "The header [headerName] is defined as non-Repeating Header in Input Headers. This header cannot have multiple occurrences in DynamicHeaders." • If it is repeating element, add the respective name value pairs under dynamic headers, which can be added to the existing list maintained for this element. • For a repeating element, if the new header name is not found under Headers section declared via TIBCO Designer, following is added into the HTTP Headers. ◦ Shows only one value, if found once in DynamicHeaders. ◦ Shows array of values, if found repeating in DynamicHeaders.
Protocol	string	Can be either HTTP or HTTPS depending upon the protocol used by the request.
Port	string	Port number on which the request was received. This is the port number configured in the HTTP Connection resource.
mimeEnvelope Element	complex	This element contains the message attachments. This element contains a repeating element named mimePart that contains each mime attachment.
mimeHeaders	complex	This element contains the mime header for each mimePart. Mime headers can contain the following information: • content-disposition – To suggest a filename for

Output Item	Datatype	Description
		an attachment, use <code>"";filename=<filename></code> in this element. Note: HTTP servers may alter or choose to ignore the suggested name. • content-type • content-transfer-encoding • content-id • other mime header information For more information about MIME headers and their syntax, see http://www.faqs.org/rfcs/rfc2045.html. Note: When the content type is specified as <code>"text/*"</code> (for example, <code>"text/xml"</code>), the attachment content is expected to be in either the <code>textContent</code> input element or the file name storing the attachment is expected to be in the <code>fileName</code> input element. When the content type is anything other than <code>"text/*"</code>, the attachment content is expected to be in either the <code>binaryContent</code> input element or the file name storing the attachment is expected to be in the <code>fileName</code> input element.
binaryContent textContent fileName	choice	This element contains the mime attachment. The element can be one of the following: • binaryContent – content of the attachment when the attachment is binary data. • textConetnt – content of the attachment when the attachment is text data. • fileName – the file name of the attachment when the attachment is stored in the file system.

Output Item	Datatype	Description
Headers	complex	The fields of the header specified on the Headers tab. For more information about these items, see the description of the Headers tab.
parameters	complex	Parameters for the incoming request. These parameters are configured in the Parameters field on the Configuration tab. When the Parse Post Method Data field on the Configuration tab is checked, the parameters include the parameters in the query string and message body of the incoming HTTP request. When the Parse Post Method Data field is unchecked, the parameters only include the query string of the incoming request.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ActivityTimeoutException	A timeout has been reached.

Java Palette

The **Java** palette has activities and shared configuration resources for executing Java code as well as converting between Java objects and XML documents.

Show Class Browser

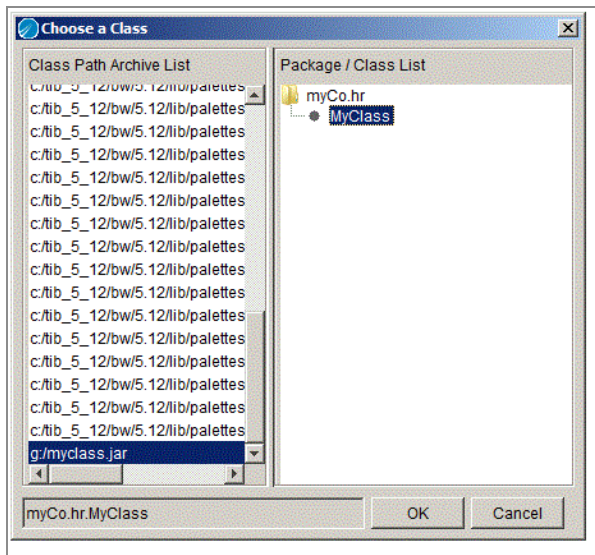
Button

The Choose a Class or Choose a Class and Method dialog appears when you click the **Show Class Browser** button. This dialog allows you to pick a Java class (and method of that class for the [Java Method](#) activity) from a list of archives available in the ActiveMatrix BusinessWorks classpath. The **Show Class Browser** button is used in the following resources:

- EJB Home
- [Java Method](#)
- [Java Schema](#)
- [Java To XML](#)
- [XML To Java](#)

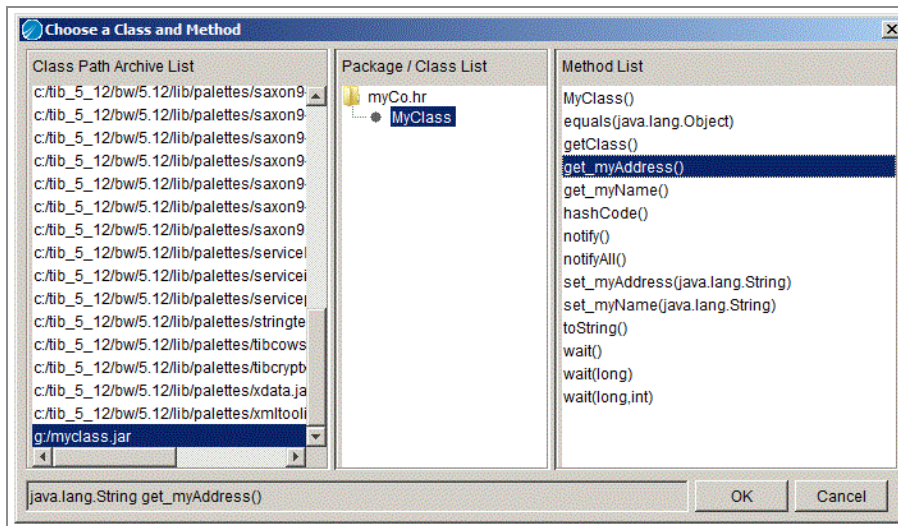
[Selecting a class with the Choose a Class dialog](#) illustrates selecting a class in the Choose a Class dialog. First you select the archive, then you select the class. Packages that contain classes are displayed as folders and the contained classes are nested below the package folder.

Figure 11: Selecting a class with the Choose a Class dialog



The [Java Method](#) and EJB Home activities allow you to select a class and a method in that class to execute. The **Show Class Browser** button brings up the Choose a Class and Method dialog. [Selecting a method in a class from the Choose a Class and Method dialog](#) illustrates this dialog.

Figure 12: Selecting a method in a class from the Choose a Class and Method dialog



Common Features of Java Code and Java Method

The [Java Code](#) and [Java Method](#) activities both execute Java. Therefore, there are some features that both activities share. This section describes some of the common features between the two activities.

Accessing ActiveMatrix BusinessWorks Global Variables and Java System Properties

You can use the `com.tibco.pe.plugin.PluginProperties.getProperty()` method to retrieve any Java system property or global variables defined in your project.

Global variables are contained in Java System properties that are prefixed with `tibco.clientVar`. For example, if you define a global variable named `myVar`, you can access the value of the global variable with the following Java code:

```
String var1 =  
    com.tibco.pe.plugin.PluginProperties.getProperty(  
        "tibco.clientVar.myVar");
```

You must specify the full path for global variables contained in variable groups to access the variable's value. Use the slash character (`/`) to separate each element in the path. For example, if you have a global variable named `item1` contained in a group named `myGroup`, you can access the value of the global variable with the following Java code:

```
String var1 =  
    com.tibco.pe.plugin.PluginProperties.getProperty(  
        "tibco.clientVar.myGroup/item1");
```

Any Java System property can also be accessed with the `PluginsProperties.getProperty()` method. Supply a string containing the property name as the input parameter to the method to obtain the property's value.

i Note: The `com.tibco.pe.plugin.PluginProperties` class is contained in the `lib/engine.jar` file within the ActiveMatrix BusinessWorks installation directory. This directory is in the ActiveMatrix BusinessWorks classpath, and therefore Java Code activities can compile code referencing this class. If you compile your code outside of ActiveMatrix BusinessWorks, you must ensure your development environment has access to the `engine.jar` file.

Sending Messages to The ActiveMatrix BusinessWorks log4j File

The standard log4j logging service is available for components of ActiveMatrix BusinessWorks. For more information about using the ActiveMatrix BusinessWorks log4j properties file, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Code in the Java Code or Java Method activities can send messages to the log4j file by using the `bw.logger` class. For example, the following code obtains the `bw.logger` class and uses the class to send a message to the log file.

```
org.apache.logging.log4j.Logger logger =
    org.apache.logging.log4j.LogManager.getLogger("bw.logger");
logger.warn("This is a warning message from the either a
Java Code or Java Method activity");
```

If you use your own logger class, you can configure the ActiveMatrix BusinessWorks log4j properties file to include your logger. This involves adding `log4j.logger` and `log4j.appender` properties to the properties file. For more information about log4j properties, see the log4j documentation . Once the properties for your custom logger class are added to the ActiveMatrix BusinessWorks log4j file, you can obtain your logger class and write messages to the log file in your Java Code and Java Method activities.

Java Code

Activity



You can add custom code to your process definition with the **Java Code** activity. This activity allows you to write standard Java code that can manipulate any of the process data or perform any action you choose.

The **Java Code** activity automatically creates an `invoke()` method in which you place the code you want to execute. This method is called when the engine processes the **Java Code** activity.

When you specify input and output parameters for the **Java Code** activity, get/set method code is automatically generated for the activity. You can use the get/set methods in your Java code, and you can display the code for the get/set methods when you select the **Full Class** radio button on the Code tab.

✔ **Tip:** Use **Edit>Preferences>Other** to specify the Java editor you want to use to edit source code. If you do not specify an editor, a simple popup editing window is provided.

For more information about features available in the **Java Code** activity, see [Common Features of Java Code and Java Method](#).

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Input Parameters	No	The input parameters for the Java Code activity. These parameters appear on the Input tab after they are specified. Each input parameter has three fields: - Field Name – Name of the parameter. - Type – Datatype of the parameter. Specify one of the supplied Java primitive types, or specify Object Reference if you are accepting a Java object from another Java Code activity as input. For more information about passing Java objects between Java

Field	Global Var?	Description
		Code activities, see Passing Java Objects Between Java Code Activities. • Occurrence – Whether the parameter is required, optional, or repeating.
Output Parameters	No	The output parameters for the Java Code activity. These parameters appear on the Output tab after they are specified. Each output parameter has three fields: • Field Name – Name of the parameter. • Type – Datatype of the parameter. Specify one of the supplied Java primitive types, or specify Object Reference if you are passing a Java object to another Java Code activity as output. For more information about passing Java objects between Java Code activities, see Passing Java Objects Between Java Code Activities. • Occurrence – Whether the parameter is required, optional, or repeating.

Code

You can choose to display the full Java class or display only the `invoke()` method body of the class. Choose the appropriate option at the top of the Code tab for the code you want to display.

The code tab has the following fields/buttons.

Field/Button	Description
Java Code	The source code of the Java class. Specify the code you want to execute in the automatically generated <code>invoke()</code> method. The get/set methods for the input and output parameters

Field/Button	Description
	are automatically generated and displayed when you choose the Full Class option above the code window. When the Java code is run by the engine, the automatically generated get/set methods are called, then the <code>invoke()</code> method in the Java class is called.
Compile Button	Compiles the current source code. When the code in the activity needs compilation (for example, the input or output parameters change or the code changes), a yellow warning icon appears on this button. The code either compile successfully or the View Errors, dialog appears to display any errors in the code. After compilation, the source and compiled java code are placed into the <code>lib/palettes/javaCode</code> directory in the ActiveMatrix BusinessWorks installation directory.
View Errors Button	Displays the errors encountered when the code was compiled.
Edit Class Button	Displays the source code in the editor specified with the TIBCO Designer preferences. If no editor is specified, a simple popup editing window is used. Use Edit>Preferences>Other to specify the Java editor you want to use to edit source code. Note: On MS Windows platforms, a temporary file is created to hold the Java code when you edit your code with this button. This file is located in either <code>C:\temp</code> or <code>C:\tmp</code> or your working directory. If your working directory contains a space (for example <code>C:\Documents and Settings\...</code>), there may be problems opening the temporary file. Make certain either the <code>C:\temp</code> or <code>C:\tmp</code> directory exists to avoid this issue.

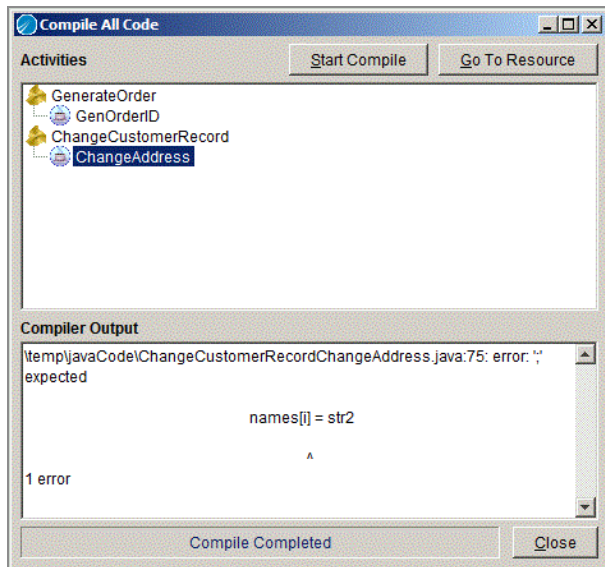
Compiling Java Code

You can compile each **Java Code** activity individually with the **Compile** button on the Code tab, or you can compile all Java Code activities in your project at once with the

Tools > Java Code > Compile All Code menu item.

When you use the Compile All Code menu item, the Compile All Code dialog appears. [Compile All Code dialog](#) illustrates this dialog.

Figure 13: Compile All Code dialog



All process definitions containing **Java Code** activities that do not compile successfully are listed in the Activities area. Each **Java Code** activity is listed under the process definition in which it is located. You can select any **Java Code** activity in the Activities area to display the compilation errors for that activity.

The dialog has the following buttons:

- **Start/Cancel Compile** – Click **Start Compile** button to begin compiling all Java Code activities in your project. When the code is compiling, this button is renamed **Cancel Compile** to allow you to cancel the compile all operation.
- **Go To Resource** –You can select any process definition or **Java Code** activity, then click the **Go To Resource** button to display that resource in the design and configuration panels.
- **Close** – Closes the Compile All Code dialog.

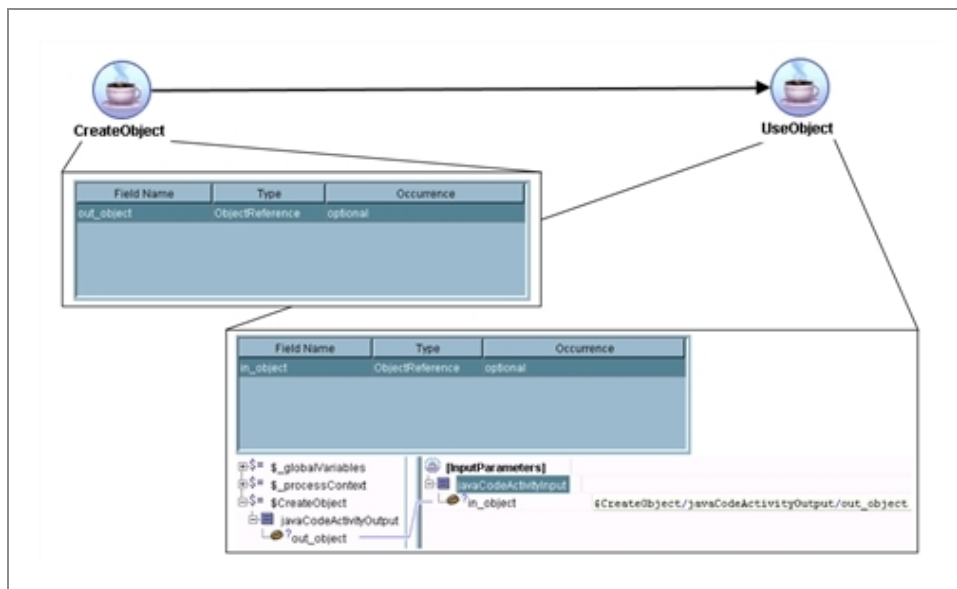
Passing Java Objects Between Java Code Activities

You may create instances of Java objects in your Java code or by using the [Java Method](#) or [XML To Java](#) activities. You can pass these Java objects using an output parameter to another activity later in the process definition. The **Java Code** activity receiving a Java object accepts the object into an input parameter and you must map the output Java object to the input object of the receiving Java Code object.

Note: Any Java objects passed by input and output parameters between activities must be serializable.

[Passing objects between Java Code activities](#) illustrates two **Java code** activities. The **CreateObject** activity creates a Java object and passes a reference to the object in the output parameter named `out_object`. The **UseObject** activity defines an input parameter named `in_object` of type `ObjectReference` and maps the **CreateObject** activity's output object reference to its input parameter of type `ObjectReference`.

Figure 14: Passing objects between Java Code activities



The **UseObject** activity can invoke methods on the input object as it would for any other object. For example, if you want to call a method named `getInteger()` that returns an integer, you would use the following code in the **UseObject** activity:

```

MyClass myObject = (MyClass) in_object;
int var = 0;
if (myObject != null) {
    var = myObject.getInteger();
}

```

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
parameters	varies	The Input tab lists all input parameters specified on the Configuration tab.

Output

The output for the activity is the following.

Output Item	Datatype	Description
parameters	varies	The Output tab lists all output parameters specified on the Configuration tab.

Error Output

The Error Output tab lists the exceptions that can be thrown by the **Java Code** activity. Because the activity can return any Java exception, all potential exceptions cannot be listed here. However, the error schema for all errors is the same. The following describes the schema for exceptions returned by the **Java Code** activity.

Error Schema Element	Datatype	Description
msg	string	The ActiveMatrix BusinessWorks error message. For more information, see <i>TIBCO ActiveMatrix BusinessWorks™ Error Codes</i> .
msgCode	string	The ActiveMatrix BusinessWorks error code. For more information, see <i>TIBCO ActiveMatrix BusinessWorks™ Error Codes</i> .
methodName	string	The name of the Java method that raised the exception.
exceptionClassName	string	The Java exception class name. The value of this element is obtained by calling the <code><exception>.getClass().getName()</code> method.
exceptionMessage	string	The Java exception message. The value of this element is obtained by calling the <code><exception>.getMessage()</code> method.

Java Custom Function

Shared Configuration



The Java Custom Function resource allows you to create custom functions to use when mapping data in an activity's input tab. These functions are also displayed when using the XPath Editor to build an XPath expression.

To create a custom function, you must write the function as a method of a Java class and compile the code into a Java class file. You then load that class file into the project using this resource.

When the class is loaded using this resource, ActiveMatrix BusinessWorks inspects the contents of the class and only the class methods that meet the following restrictions are made available in XPath:

- Only methods declared as `public` and `static` are loaded.
- The input parameters and return values must be of one of the types described in

Datatypes allows for Java custom functions.

- The return value of the function cannot be void.
- The method cannot be a constructor.
- The method cannot explicitly throw an exception. Runtime exceptions are allowed, however.
- Method names cannot be overloaded in a class or any imported classes in a single Java Custom Function resource. You can load methods of the same name into separate classes in separate Java Custom Function resources and use the Prefix field to differentiate between the methods.
- If you make references to any imported class files, these classes must be available in the classpath configured for ActiveMatrix BusinessWorks. The easiest way to make the imported classes available is to place them in the TIBCO/bw/2.0/lib directory.
- Inner classes are not supported.



Tip: ActiveMatrix BusinessWorks provides detailed online help for each XPath function. To provide online help for Java Custom Functions, create a two-dimensional array named `HELP_STRINGS` in your class. The element containing a string matching the function name is used as the help for that function. For an example of creating the `HELP_STRINGS` array see [Example Java Function](#).

Datatypes allows for Java custom functions

Type of Data	Datatypes Allowed for Input Parameters and Return Types	
Strings and Characters	String	char
Boolean	Boolean	boolean
Numbers	int	Integer
	float	Float
	double	Double
	short	Short
	long	Long



Warning: Static variables are not supported in Java custom functions. Do not write custom functions that use static variables.

If there are different classes that have methods with the same names, the Prefix field allows you to specify a prefix for qualifying which method you want to use in an XPath function. For more information, see the description of the Prefix field below.

For more information about building XPath expressions and using the XPath editor, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the resource.
Description	Short description of the shared resource. The method signature plus the description supplied in this field appear as the documentation of the function in the XPath editor. You can use simple HTML markup to format the text of this field, if you desire.
Suggested Prefix	A folder with the name you specify in this field appears in the Functions tab in the XPath editor. You can drag and drop methods from the loaded class into your XPath expression, just as you would with any standard XPath function. Use this name as a prefix for function names in this class, if multiple classes are loaded and function names are not unique among the classes. For example, if you have <code>class1.method1</code> and <code>class2.method1</code>, you need to specify <code>class1</code> in the Suggested Prefix field when you load <code>class1</code> into a Java Custom Resource. You would specify <code>class2</code> in the Suggested Prefix field when loading <code>class2</code>. When using <code>method1</code> in XPath expressions, you must qualify which <code>method1</code> you are using by specifying <code>class1:method1</code> or <code>class2:method1</code> in the XPath expression.

Field	Description
	The value of this field must be unique across all loaded Java Custom Function resources, and the value must be a valid XPath identifier. Also, you cannot use global variables (that is, %variable%) in this field.
Class Location	The location of the class file you want to load. Use the Browse button to locate the class file. Note: This field is used to locate the file initially, but the file is actually loaded and stored in the repository. Once the file is loaded, it can be removed from its original location in the file system. If you want to change the file stored in the repository, you can use this field to reload a changed file or load a new file for this resource.
Class Data	The size of the class in bytes. Use the Load button after a class is selected in the Class Location field to load the class and determine its size.

Example Java Function

The following Java code implements custom Java functions. The example illustrates which methods of the class meet the restrictions and are therefore available as custom functions in XPath. The example also illustrates methods which do not meet the restrictions, and therefore are not available as custom functions.

```
package com.tibco;
public class Sample
{
    protected int mInternalValue;
    /**
     * The following method is not available because it is
     * a constructor.
     */
    public Sample(int value)
    {
        mInternalValue = value;
    }
    /**
     * This method is used to concat two strings together. It
     * must be declared as public static to be made available in
     * TIBCO ActiveMatrix BusinessWorks.
     */
}
```

```

public static String stringConcat(String s1, String s2)
{
    return s1 + s2;
}
/**
 * This method is used to add two ints together. Note, that
 * it takes both an int type and an Integer object.
 */
public static int intAdd(int lhs, Integer rhs)
{
    return lhs + rhs.intValue();
}
/**
 * The following method is not available because it
 * throws an exception.
 */
public static int badAdd(int lhs, int rhs)
throws Exception
{
    long result = lhs + rhs;
    if (result > Integer.MAX_VALUE) {
        throw new ArithmeticException();
    }
    return new Long(result).intValue();
}
/**
 * The following method is not available because it
 * returns nothing: its void.
 */
public static void returnsNothing(String s)
{
    System.out.println(s);
}
/**
 * The following method is not available because it
 * is not static.
 */
public int add(int rhs)
{
    return mInternalValue + rhs;
}
/**
 * The following method is not available because it
 * is not public.
 */
protected static int protectedAdd(int lhs, int rhs)
{
    return lhs + rhs;
}

```



```

    }
    /**
     * The following is a two-dimensional array that provides the
     * online help for functions in this class. Declare an array
     * named HELP_STRINGS.
     */
    public static final String[][] HELP_STRINGS = {
        {"stringConcat", "Joins two strings.",
        "Example", "stringConcat(\"test/testDict\",
        $input/key)"},
        {"intAdd", "Adds two integers.",
        "Example", "intAdd(5, $input/myInt)"},
    }
}

```

Java Event Source

Process Starter



The Java Event Source allows you to create a custom process starter written in Java. For example, you may want to start a ActiveMatrix BusinessWorks process when an application inserts a row into a database table. Your custom process starter would observe the database for insert events, then call the `onEvent()` method with the desired data as input when an insert occurs. The Java Event Source process starter creates a process when the `onEvent()` method is invoked and the object passed to the method is passed to the process definition.

This process starter uses an abstract class to define the interface. You can either write and compile your custom Java code in your own code editor and upload the class to ActiveMatrix BusinessWorks, or you can use the Code tab to add your implementation code to the appropriate methods. For more information about the methods required by this class, see [JavaProcessStarter Abstract Class](#).



Tip: If you want to use the **Edit Class** button on the Code tab, select **Edit>Preferences>Other** to specify the Java editor you want to use to edit source code. If you do not specify an editor, a simple popup editing window is provided.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Use External Class	No	When this field is checked, the Code tab is disabled, and you must load the class using the Class Library and Class fields. When this field is unchecked, you must add your code to the appropriate methods on the Code tab.
Java	No	This field contains the Class Library, Class, and Init Method fields that allow you to specify which class and method to use.
Class Library	No	Use this field to specify an AliasLibrary resource that is used to locate the Java class files. For more information about AliasLibrary resources, see <i>TIBCO Designer™ Palette Reference</i> .
Class	No	The compiled Java class that implements the JavaProcessStarter abstract class. Use the Show Class Browser button to select the class from the specified Class Library.
Init Method	No	The init() method for the class. This field is automatically populated when you use the Show Class Browser button. For more information about the init() method, see JavaProcessStarter Abstract Class. You can use the drop down list to select a different method in the selected Java class, if desired. Note: You can hover the cursor over this field to display the signature of the selected method.

Field	Global Var?	Description
Init Method Parameters	No	This table allows you to specify default values for the input parameters of the <code>init()</code> method. Each input parameter has three fields: • Parameter Name – Name of the parameter. • Parameter Type – Datatype of the parameter. • Parameter Value – Specify a default value for the parameter.

JavaProcessStarter Abstract Class

The `JavaProcessStarter` abstract class defines the interface of your Java Process Starter with the ActiveMatrix BusinessWorks process engine. You must define an implementation for the following methods:

- `init()` – this method is called when the process engine starts up. This method should initialize any resource connections. You could also specify a Java Global Instance on the Advanced tab that initializes resource connections. Java Global Instances are also loaded and initialized during process engine start up. You can call `this.getJavaGlobalInstance()` to obtain the Java Global Instance resource in your process starter code.
- `onStart()` – this method is called by the process engine to activate the process starter. This method should activate any event notifier or resource observer code. The notifier or observer code can then call the `onEvent()` method to start a process instance.
- `onStop()` – this method is called by the process engine to deactivate the process starter. This method should deactivate any event notifier or resource observer code.
- `onShutdown()` – this method is called by the process engine when the engine shuts down. This method should release any resources and resource connections and perform any required clean up operations.

The following methods are already implemented and can be used in your code:

- `onEvent(Object object)` – this method is called when a listener or resource observer catches a new event. The input to this method is a Java object containing the event data.
- `getGlobalInstance()` – this method returns an object reference to the Java Global Resource specified on the Advanced tab of the process starter. This is useful if you want to place initialization code or other shared information in a Java Global Resource instead of in the `init()` method of this class.
- `onError()` – this method throws the exception specified in the input parameter. Use this method to propagate an error to the ActiveMatrix BusinessWorks process instance when a listener or resource observer fails to generate an event.

Code

This tab is only enabled if the Use External Class field is unchecked. This tab allows you to add implementation code to the appropriate methods of the `JavaProcessStarter` class. You can choose to display the full Java class or display a particular method by selecting the appropriate value in the Show field at the top right of the Code tab.

The code tab has the following fields/buttons.

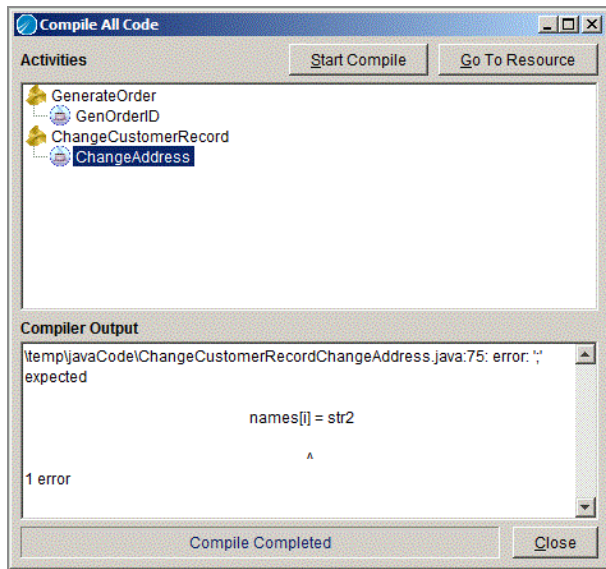
Field/Button	Description
Source Code	The source code of the <code>JavaProcessStarter</code> implementation class.
Show	Displays the selected method or the entire class. The following are valid options to select: • <code>init</code> • <code>onStart</code> • <code>onStop</code> • <code>onShutdown</code> • Entire Class
Compile Button	Compiles the current source code. When the code in the activity needs compilation (for example, the code changes), a yellow warning icon appears on this button.

Field/Button	Description
	The code either compile successfully or the View Errors dialog appears to display any errors in the code. After compilation, the source and compiled java code are placed into the lib/palettes/javaCode directory in the ActiveMatrix BusinessWorks installation directory.
View Errors Button	Displays the errors encountered when the code was compiled.
Edit Class Button	Displays the source code in the editor specified with the TIBCO Designer preferences. If no editor is specified, a simple popup editing window is used. Use Edit>Preferences>Other to specify the Java editor you want to use to edit source code. Note: On MS Windows platforms, a temporary file is created to hold the Java code when you edit your code with this button. This file is located in either C:\temp or C:\tmp or your working directory. If your working directory contains a space (for example C:\Documents and Settings\...), there may be problems opening the temporary file. Make certain either the C:\temp or C:\tmp directory exists to avoid this issue.

Compiling Java Code

You can compile each **Java Code** activity and Java Process Starter resource individually with the **Compile** button on the Code tab, or you can compile all Java Code and Java Process Starter resources in your project at once with the **Tools > Java Code > Compile All Code** menu item.

When you use the Compile All Code menu item, the Compile All Code dialog appears. [Compile All Code dialog](#) illustrates this dialog.

Figure 15: Compile All Code dialog

All process definitions containing Java Code activities and Java Process Starters that do not compile successfully are listed in the Activities area. Each Java Code activity or Java Process Starter is listed under the process definition in which it is located. You can select any Java Code activity or Java Process Starter in the Activities area to display the compilation errors for that activity.

The dialog has the following buttons:

- **Start/Cancel Compile** – Click **Start Compile** to begin compiling all Java Code activities and Java Process Starters in your project. When the code is compiling, this button is renamed **Cancel Compile** to allow you to cancel the compile all operation.
- **Go To Resource** –You can select any process definition, Java Process Starter, or **Java Code** activity, then click the **Go To Resource** button to display that resource in the design and configuration panels.
- **Close** – Closes the **Compile All Code** dialog.

Advanced

The Advanced tab has the following fields:

Field	Global Var?	Description
Java Global Instance	No	A Java Global Instance resource. Specify a value in this field if you want to reference a Java Global Instance resource in your JavaProcessStarter implementation class. You can use the <code>getGlobalInstance()</code> method to obtain a reference to the Java Global Instance resource. For more information about Java Global Instance resources, see Java Global Instance.
Convert Output Object To XML	No	When this field is unchecked, the object passed to the <code>onEvent()</code> method is output by this process starter as a Java object reference. You can use the Java To XML activity to render the Java object as an XML tree, if desired. Check this field if you want the data of the input object to the <code>onEvent()</code> method to be converted to an XML tree in this process starter's output. For the output to be returned as an XML tree, the output object type must be specified. To achieve this, you must add the <code>getOutputObjectType()</code> class to your implementation class. For example, the output can be a string: <pre>public Class getOutputObjectType() { return String.class; }</pre> Then, the object you pass into the input parameter of the <code>onEvent()</code> method must be a String.

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output

The output for the activity is the following.

Output Item	Datatype	Description
javaObject or Object	varies	The Java object that was passed to the <code>onEvent()</code> method. By default, this element is a Java object reference. If the Convert Output Object to XML field is checked on the Advanced tab and the output class is specified in your implementation class, the output is represented as an XML tree.

Error Output

The Error Output tab lists the exceptions that can be thrown by the Java Event Source resource. Because the resource can return any Java exception, all potential exceptions cannot be listed here. However, the error schema for all errors is the same. The following describes the schema for exceptions returned by the Java Event Source resource.

Error Schema Element	Datatype	Description
msg	string	The ActiveMatrix BusinessWorks error message. For more information, see <i>TIBCO ActiveMatrix BusinessWorks™ Error Codes</i> .
msgCode	string	The ActiveMatrix BusinessWorks error code. For more information, see <i>TIBCO ActiveMatrix BusinessWorks™ Error Codes</i> .

Java Global Instance

Shared Configuration



The Java Global Instance shared configuration resource allows you to specify a Java object that can be shared across all process instances in a Java Virtual Machine (JVM). When the process engine is started, an instance of the specified Java class is constructed. When the process engine is shut down, if specified, a cleanup method is invoked on the object and the object is released before the engine shuts down. Any [Java Method](#) activity can be configured to access the shared Java Global Instance when the process engine runs. Any [Java Code](#) activity can access the shared Java Global Instance by invoking the static methods of the configured Java class.

If multiple process instances access the shared Java Global Instance, you may want to ensure that only one process instance can access the object at a time. You can accomplish this by either declaring the methods of the configured class as synchronous or by using a critical section group. For more information about using critical section groups, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

i Note: Java class files or archives referenced by this resource must be available to the Java VM in which TIBCO Designer and the ActiveMatrix BusinessWorks process engine are running. This is done either by adding the correct path to the classpath in the `designer.tra` and `bwengine.tra` files (For more information about modifying the classpath, see the comments in the `.tra` files) or by using an Alias Library resource. If you add the path to the classpath, the system class loader loads the class into the Java VM. If you use an Alias Library resource, the ActiveMatrix BusinessWorks application class loader loads the class into the Java VM. If both techniques are used, the class loaded by the system class loader takes precedence.

Objects created from classes loaded by different class loaders are not able to interact. For example, an Alias Library resource points to the location of Java ClassA and a **Java Method** activity references ClassA. Java class ClassB is placed in the classpath and referenced by a **Java Code** activity. The **Java Method** activity can pass the ClassA object to the **Java Code** activity, but the ClassB object in the **Java Code** activity may not be able to interact with the ClassA object. If you want objects to interact correctly, ensure that the activities use the same class loader.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the shared configuration resource.
Description	No	Short description of the resource.
Java	No	This field contains the Library, Class, and Method fields that specify the Java object to create.
Library	No	Use this field to specify an AliasLibrary resource that is used to locate the Java class files. For more information about AliasLibrary resources, see <i>TIBCO Designer™ Palette</i>

Field	Global Var?	Description
<i>Reference.</i>		
Class	No	Use the Show Class Browser button to locate the class you want to use.
Method	No	Select a constructor from the drop down list of constructors available for this class. This can be used to construct the class when the engine is started. This field displays only the default constructor (that is, the constructor with no parameters) or constructors that have only parameters of Java primitive types. You cannot create an object using a constructor that requires an object, array, or non-primitive type for a parameter.
Parameters Input	No	The input parameters for the constructor selected in the Method field. Each input parameter has three fields: • Parameter – Name of the parameter. • Type – Datatype of the parameter. • Value – Supply a value for the parameter in this field. Note: When string is the datatype for the parameter, do not use quotes around the string when entering the value.

Advanced

The Advanced tab has the following fields:.

Field	Global Var?	Description
Invoke Cleanup	No	When checked, this specifies that you want to invoke a method on the object before the object is released from memory when

Field	Global Var?	Description
Method		the process engine terminates. This allows you to perform cleanup activities, such as closing connections, releasing resources, and so on. When this field is checked, the Cleanup Method field appears.
Cleanup Method	No	This field is only available when the Invoke Cleanup Method field is checked. This field allows you to specify the name of the method to invoke before the object in memory is released. The method selected cannot contain any parameters, therefore only methods without parameters are listed in this field. Any values returned by the selected method are ignored, but exceptions raised by the selected method are sent to the log.

Java Method

Activity



The **Java Method** activity allows you to invoke a method contained in a Java class. You can construct an instance of the specified Java class, if you choose to invoke the constructor for the class.

The Java class file must be located in the classpath for TIBCO Designer and the ActiveMatrix BusinessWorks process engine. Update the `designer.tra` and `bwengine.tra` file to contain the directory where your Java class files are located.

For more information about features available in the **Java Code** activity, see [Common Features of Java Code and Java Method](#).

Configuration

The Configuration tab has the following fields:

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Java Global Instance	No	You can execute methods on shared Java objects created by Java Global Instance resources by specifying the resource in this field. For more information about the shared Java objects created by Java Global Instance resources, see Java Global Instance. When a resource is specified in this field, the Library and the Class fields are automatically populated with the correct values from the Java Global Instance resource, and the fields on the Advanced tab become unavailable. You can select the method to execute in the drop down list in the Method field. Note: If the configuration of the Java Global Instance resource changes, you must select and view any Java Method activities in your process definitions that reference the Java Global Instance to automatically propagate the changes. An exception is raised if you attempt to test your process definitions before refreshing the reference to the Java Global Instance. You may need to manually update the mappings of any input or output if you select a new method with a different signature in the Java Global Instance resource.
Java	No	This field contains the Library, Class, and Method fields that allow you to specify which class and method to execute.
Library	No	Use this field to specify an AliasLibrary resource that is used to locate the Java class files. For more information about AliasLibrary resources, see <i>TIBCO Designer™ Palette Reference</i> .
Class	No	The declared Java class whose method you want to execute.

Field	Global Var?	Description
		Use the Show Class Browser button to select the class and method, or use the Java Global Instance field to specify a shared Java object.
Method	No	The method in the selected Java class you want to execute. This field is automatically populated when you use the Show Class Browser button. You can use the drop down list to select a different method in the selected Java class, if desired. Note: You can hover the cursor over this field to display the signature of the selected method.

Advanced

The Advanced tab has the following fields:

Field	Global Var?	Description
Construct Declared Class	No	This field controls whether the declaring class must be constructed when the activity is executed or whether an object reference must be supplied in the activity's input schema. When this field is checked, the declaring class is constructed using the default constructor when the activity executes. In this case, the declaring class is not included in the activity's input schema. When this field is unchecked, you must supply an object reference for the Java class specified on the Configuration tab. The object must be created and contained in the output schema of a previously executed activity, such as a Java Code activity, another Java Method activity, a RMI Server

Field	Global Var?	Description
		process starter, or an XML To Java activity. In this case, the declaring class is included in the activity's input schema. This field is not available when a static method is selected.
Cache Constructed Class	No	This field is only available when the Construct Declared Class is checked. When checked, this specifies that all instances of this Java Method activity share the instance of the constructed declaring class. If the constructed object is cached, it is kept in memory until the process engine terminates. If this field is unchecked, each instance of this Java Method activity creates its own copy of the declaring class. For more information about caching the declared class, see Caching the Java Class.
Invoke Cleanup Method	No	This field is only available when the Cache Constructed Class field is checked. When checked, this specifies that you want to invoke a method on the cached object before the object is released from memory when the process engine terminates. This allows you to perform cleanup activities, such as closing connections, updating database tables, and so on. When this field is checked, the Cleanup Method field appears.
Cleanup Method	No	This field is only available when the Invoke Cleanup Method field is checked. This field allows you to specify the name of the method to invoke before the cached class in memory is released. The selected method cannot contain any parameters, therefore only methods with no parameters are listed in this field. Any values returned by the selected method are ignored, but any exceptions raised are sent to the log.

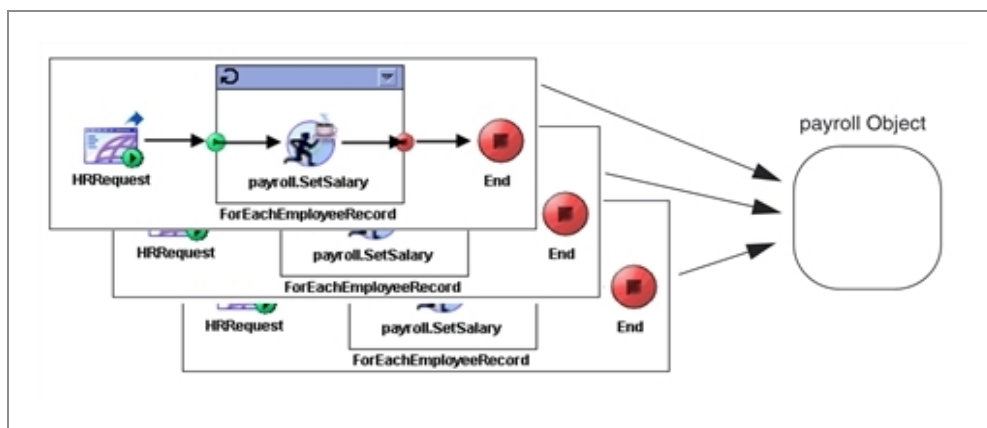
Caching the Java Class

When the Cache Constructed Class check box is selected on the Advanced tab, all instances of the same **Java method** activity share the same Java object. For example, each process instance that executes this **Java Method** activity shares the same instance of the class.

Also, if this activity is executed multiple times in a loop, each execution of this activity uses the same cached instance of the object. [Sharing a cached class](#) illustrates multiple processes and Java Methods in a group all sharing the same Java class.

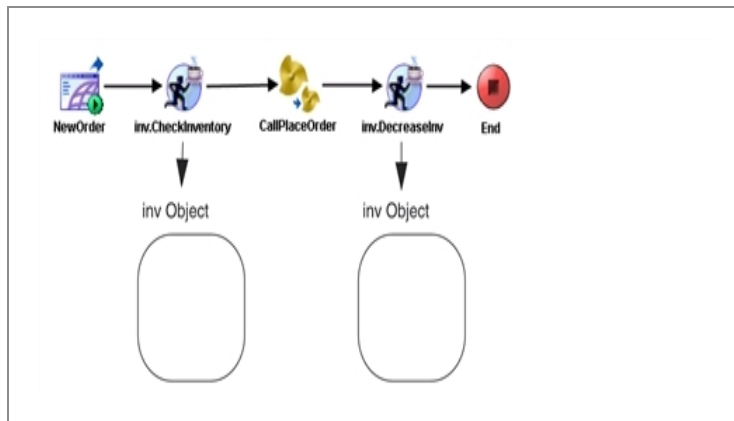
Warning: If you choose to cache the constructed instance of the class, the same Java object is shared by Java Method activities that invoke methods on the class. Therefore, be aware of potential concurrency issues, if your class or its methods are not threadsafe.

Figure 16: Sharing a cached class



However, different instances of a **Java Method** activity that invoke methods on the same class do not share the instance of the object. For example, [Different Java Method activities in the same process do not share the class](#) illustrates two Java Method activities in the same process definition, each executing different methods on the same Java object. However, each of these **Java Method** activities operate on their own instances of the inv class, if they both select the Cache Constructed Class check box.

Figure 17: Different Java Method activities in the same process do not share the class



In the process definition illustrated in [Different Java Method activities in the same process do not share the class](#), for each process instance that is created, each time the `inv.CheckInventory` activity executes, it shares one instance of the `inv` object that has been cached. Similarly, each time the `inv.DecreaseInv` activity executes, it shares the instance of the `inv` object that the activity has cached. However, the `inv.CheckInventory` and `inv.DecreaseInv` activities do not share their cached Java objects with each other.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
<declaringClass>	Java Object	An object reference to the Java Class specified on the Configuration tab. This element is not shown when the method you want to invoke it static. Also, this element is only available when the Construct Declared Class field on the Advanced tab is unchecked.
MethodParameters	varies	This element contains a list of input parameters that are required for the Java method selected on the

Input Item	Datatype	Description
		Configuration tab.
		This element is not available when the selected method has no parameters.

Output

The output for the activity is the following.

Output Item	Datatype	Description
MethodReturnValue	varies	The Output tab have the return value specified for the Java method selected on the Configuration tab.
		This element is not available for methods that return void.

Error Output

The Error Output tab lists the exceptions that can be thrown by the invoked method. Once a method to invoke is selected, the exceptions that the method can throw are listed on the Error Output tab. Because a method can return any Java exception, the potential exceptions cannot be listed here.

While all potential errors cannot be known, the error schema for all errors is the same. The following describes the schema for exceptions returned by the **Java Method** activity.

Error Schema Element	Datatype	Description
<exceptionName>	complex	The fully-qualified name of the exception, including the Java package name. This is a container for the other elements.

Error Schema Element	Datatype	Description
<exceptionName>	Java Object	The Java exception object.
exceptionClassName	string	The exception class name. The value of this element is obtained by calling the <exception>.getClass().getName() method.
exceptionMessage	string	The exception message. The value of this element is obtained by calling the <exception>.getMessage() method.

Java Schema

Shared Configuration



The Java Schema shared configuration resource allows you to specify a Java class that is used to configure a [Java To XML](#) or [XML To Java](#) activity.

The Java class is converted to an XML schema using the following rules:

- Only the public data members of the Java class are mapped to XML elements. Private and protected members are not included.
- The Java class member public variable name is mapped to an XML element with the same name. For example, a Java class member variable declared as public int ZipCode is mapped to an XML element named ZipCode.
- Java Bean accessors and modifiers are mapped to appropriate XML element names. For example, a Java class method public int getBalance() or public void setBalance(int Balance) are mapped to an XML element named Balance.
- Only one XML element is created regardless of how many members of the Java class share the same name. For example, there may be an attribute named MySalary and accessors named getMySalary() and setMySalary(). This translates to one element named MySalary in the resulting XML document.
- Conversion is case-sensitive. For example, if you have a data member declared as public int zipCode and an accessor method declared public void setZipCode(int

NewZipCode), there are two elements in the resulting XML schema, zipCode (for the data member) and ZipCode (for the accessor). You must make sure to use the correct element when using the resulting XML in input mappings of subsequent activities.

- All Java primitive types are supported. Datatypes that extend `java.util.Collections` are also supported (for example, `List`, `ArrayList`, and `Vectors`). Arrays (for example, `int[]` and `string[]`) are also supported.
- The datatype `java.util.Map` or any types that extend `java.util.Map` are not supported. For example, `HashMap` is not supported.

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the resource.
Description	Short description of the shared resource.
Java	This field contains the Library and Class fields that allow you to specify the Java class for this resource.
Library	Use this field to specify an AliasLibrary resource that is used to locate the Java class files. For more information about AliasLibrary resources, see <i>TIBCO Designer™ Palette Reference</i> .
Class	The Java class that specifies the data of the schema. This class must implement or be a subtype of a class that implements the <code>Serializable</code> interface. You can use the Show Class Browser button to browse Java classes in the ActiveMatrix BusinessWorks classpath.
Schema Namespace	The namespace to use for the schema when converting to or from XML.

Schema Text

The Schema Text tab displays the resulting XML schema from the Java class selected on the Configuration tab.

Java To XML

Activity



The **Java To XML** activity allows you to convert a Java object's data members into an XML document. For more information about the conversion rules, see [Java Schema](#).



Note: If the class does not have a public data member and only has a Java bean modifier that sets the data, the input schema contains an element for the modifier, but the resulting XML document has no value set for the corresponding element.

For example, the Java object has a method declared as `public int setID()`, but there is no method for getting the ID, and the data member ID is not public. In this case, there is an element named ID in this activity's output schema, but that element has no value because there is no public mechanism for getting the data.

Enumeration

Java enum type elements can be mapped to a simple type schema with enumeration facets. You can use the following annotations in your Java classes to use Java enums :

- `javax.xml.bind.annotation.XmlEnum`
- `javax.xml.bind.annotation.XmlEnumValue`
- `XmlEnum` and `XmlEnumValue`

together provide a mapping of type enum to the XML representation.

For more information and for the usage of these annotations, see <https://jaxb.java.net/nonav/2.2.4/docs/api/javax/xml/bind/annotation/XmlEnum.html>

and

<https://jaxb.java.net/nonav/2.2.4/docs/api/javax/xml/bind/annotation/XmlEnumValue.htm>
|

Java to Schema Data Type Binding

Field	Description
xsd:string	java.lang.String
xsd:boolean	java.lang.Boolean
xsd.byte	java.lang.Byte
xsd:short	java.lang.Boolean
xsd:int	java.lang.Integer
xsd:long	java.lang.Long

Example 1

```
@XmlEnum(String.class)
public enum Card { CLUBS, DIAMONDS, HEARTS, SPADES }
<xs:simpleType name="Card">
  <xs:restriction base="xs:string"/>
  <xs:enumeration value="CLUBS"/>
  <xs:enumeration value="DIAMONDS"/>
  <xs:enumeration value="HEARTS"/>
  <xs:enumeration value="SPADES"/>
</xs:simpleType>
```

Example 2

```
@XmlEnum(Integer.class)
public enum Coin {
  @XmlEnumValue("1") PENNY(1),
  @XmlEnumValue("5") NICKEL(5),
  @XmlEnumValue("10") DIME(10),
  @XmlEnumValue("25") QUARTER(25) }
<!-- Example: XML Schema fragment -->
<xs:simpleType name="Coin">
  <xs:restriction base="xs:int">
  <xs:enumeration value="1"/>
```

```

<xs:enumeration value="5"/>
<xs:enumeration value="10"/>
<xs:enumeration value="25"/>
</xs:restriction>
</xs:simpleType>

```

Sequenced Elements

For a Java schema to generate the schema of complex type element having a sequence order indicator in the right sequence, use XMLType annotations in the Java classes.

This change in behavior is enabled by a system property

```
java.property.com.tibco.xml.conversion.SequencingViaAnnotation=true.
```

ActiveMatrix BusinessWorks tries to get the order of the element through annotations when the property is set to true. The default value is false.

Use the following annotation:

```
javax.xml.bind.annotation.XmlType
```

Properties and fields mapped to elements are mapped to a content model within a complex type.

The annotation element `propOrder()` can be used to customize content model to `xs:sequence`. This is used to specify the order of the XML elements in `xs:sequence`.

For example, map a class to a complex type element with `xs:sequence` with a customized ordering of JavaBean properties.

```

@XmlType(propOrder={"street", "city", "state", "zip", "name" })
public class USAddress {
    private String name;
    private String street;
    private String city;
    private String state;
    private long zip;

    public String getName() {..};
    public void setName(String) {..};

    public String getStreet() {..};
    public void setStreet(String) {..};
    public String getCity() {..};
    public void setCity(String) {..};

    public String getState() {..};

```

```

public void setState(String) {..};
public long getZip() {..};
public void setZip(long zip) {..};
}
<!-- XML Schema mapping for USAddress -->
<xs:complexType name="USAddress">
<xs:sequence>
<xs:element name="street" type="xs:string"/>
<xs:element name="city" type="xs:string"/>
<xs:element name="state" type="xs:string"/>
<xs:element name="zip" type="xs:decimal"/>
<xs:element name="name" type="xs:string"/>
</xs:sequence>
</xs:complexType>

```

An exception is thrown if any of the following conditions are true:

All the field members are not specified in the proporder element. Or, proporder contains the name of a field member that does not exist.

Configuration

The Configuration tab has the following fields.

XML Schema Type	Global Var?	Java Data Type
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the activity.
Use Java Schema		Specifies whether you want to use a Java Schema shared configuration resource to configure this activity or whether you want to specify a Java class file. When unchecked, the Java Class field appears. When checked, the Java Schema: and Class: fields appear.
Java Schema:	No	Only available when the Use Java Schema field is checked. This field specifies a Java Schema shared configuration resource. This resource is used to specify the schema used to

XML Schema Type	Global Var?	Java Data Type
		convert the Java object to the XML schema.
Java Class:		When Use Java Schema is checked, this is the class in the Java Schema to use. Use the Show Class Browser button to bring up the dialog to choose th Java class.

Advanced

The Advanced tab has the following fields:.

Field	Global Var?	Description
Construct Declared Class	No	This field controls whether the declaring class must be constructed when the activity is executed or whether an object reference must be supplied in the activity's input schema. When this field is checked, the declaring class is constructed using the default constructor when the activity executes. In this case, the declaring class is not included in the activity's input schema. When this field is unchecked, you must supply an object reference for the Java Class specified on the Configuration tab. The object must be created and contained in the output schema of a previously executed activity, such as a Java Code activity, Java Method, or an XML To Java activity. In this case, the declaring class is included in the activity's input schema.
Cache Constructed Class	No	This field is only available when Construct Declared Class is checked.

Field	Global Var?	Description
		When checked, this specifies that all instances of this Java To XML activity share the instance of the constructed declaring class. If the constructed object is cached, it is kept in memory until the process engine terminates. If this field is unchecked, each instance of this Java To XML activity creates its own copy of the declaring class. For more information about caching the declared class, see Caching the Java Class.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
<declaringClass>	Java Object	An object reference to the Java Class specified on the Configuration tab. This element is only available when the Construct Declared Class field on the Advanced tab is unchecked.

Output

The output for the activity is the following.

Output Item	Datatype	Description
XML Schema	varies	The XML schema from the converted Java object is output. The elements appear in the schema in alphabetical order.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
JavaToXMLConversionException	An error occurred when converting the Java object to XML.

XML To Java

Activity



The XML to Java activity allows you to create an instance of a Java object based on data from an XML document. The XML schema for providing input to the Java object is created from the Java object or Java Schema specified on the Configuration tab of this activity. The specified Java class must meet the following requirements:

- The Java class must have a public default constructor (that is, a constructor with no arguments).
- The Java class must be serializable (that is, the class must implement or be a subclass of a class that implements `java.io.Serializable`).

For more information about description of the conversion rules, see [Java Schema](#).



Note: If the class does not have a public data member and only has a Java bean accessor that retrieves the data, the input schema contains an element for the accessor, but the resulting Java object has no value set for the member.

For example, the object has a method declared as `public int getID()`, but there is no method for setting the ID and the data member ID is not public. In this case, there is an element named ID in this activity's input schema, but mapping a value to the element does not result in setting the ID member of the output Java object because there is no public mechanism for setting the data.

Sequenced elements and mapping of enumeration to simple type schemas is supported in the **XML to Java** activity.

For more information, see [Sequenced Elements](#) and [Java To XML](#) .

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the activity.
Use Java Schema	No	Specifies whether you want to use a Java Schema shared configuration resource to configure this activity or whether you want to specify a Java class file. When unchecked the Java Class field appears. When checked, the Java Schema: and Class: fields appear.
Java Schema:	No	Only available when the Use Java Schema field is checked. This field specifies a Java Schema shared configuration resource. This resource is used to specify the schema used to convert the XML schema to the Java object.
Java Class:	No	When Use Java Schema is checked, this is the class in the Java Schema to use. Use the Show Class Browser button to bring up the dialog to choose the Java class.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
XML Schema	varies	The XML schema that you want to convert to a Java object. The content of this schema depends upon the Java class or Java Schema defined on the Configuration tab. The elements appear in the schema in alphabetical order.

Output

The output for the activity is the following.

Output Item	Datatype	Description
<declaringClass>	Java Object	A Java object that is created out of the XML schema.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
XMLToJavaConversionException	An error occurred when converting the XML schema to a Java object.

JDBC Palette

The JDBC palette contains activities and shared configuration resources for querying, updating, or calling stored procedures in a database.

JDBC Call Procedure

Activity



The **JDBC Call Procedure** activity calls a database procedure using the specified JDBC connection.

If this activity is not part of a transaction group, the SQL statement is committed after the activity completes. If this activity is part of a transaction group, the SQL statement is committed or rolled back with the other JDBC activities in the group at the end of the transaction. For more information about creating groups for transactions, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

If you want to override the default behavior of transaction groups for certain JDBC activities in a transaction group, you can check the Override Transaction Behavior field on the Advanced tab. This specifies that the activity is outside of the transaction and the SQL statement is committed when the activity completes, even if the activity is in a transaction group.

The **Refresh** button on this activity allows you to synchronize the activity with the contents of the database. This is useful if you make a change to the database while you are editing a process definition containing this activity in ActiveMatrix BusinessWorks.



Note: When using the jConn or Data Direct 3.3 driver and connecting to a Sybase database, stored procedures that are supposed to return multiple result sets do not return any data. To avoid this problem, you can either use the `sun.jdbc.odbc.JdbcOdbcDriver` instead or you can add `set nocount on` to your stored procedure definition.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JDBC Connection	Yes	Path to the shared configuration resource containing the JDBC connection information. For more information about JDBC resources, see JDBC Connection .
Schema	Yes	The schema in which the procedure resides. This name is used to resolve naming conflicts if more than one schema in the database has the selected procedure with the same name. The Select Schema button allows you to choose a schema from the database using the specified JDBC Connection.
Catalog/Package	Yes	The catalog or package in which the procedure resides. This name is used to resolve naming conflicts if more than one catalog or package in the database has the selected procedure with the same name. This field is only applicable to databases that have more than one catalog and package. For more information about catalogs and packages, see your database documentation. The Select Catalog/Package button allows you to choose a catalog or package from the database using the specified JDBC Connection. If a schema is specified in the Schema field, only catalogs or packages in the specified schema are listed.

Field	Global Var?	Description
Procedure/ Function Name	Yes	Name of the database procedure or user-defined function to call. You can use the Select Procedure/Function button to query the database for available procedures or functions once the JDBC Connection field is specified. The list of available procedures and functions are limited by the values supplied in the Schema and Catalog/Package fields. Note: The input parameters and output parameters are displayed when you use the Select Procedure/Function button. TIBCO Designer retrieves the signature of each stored procedure/function from the database once. If you change the signature while editing your project, you must click the Refresh button to retrieve the changes from the database.
Timeout (sec)	Yes	Time to wait for the procedure call to complete. If the call does not complete in the given time limit, an error is returned.
Maximum Rows	Yes	The maximum number of rows to retrieve. To retrieve all rows, specify 0.
Parameter Types	No	This field displays the parameters for the procedure and allows you to override the parameter type (IN, OUT, INOUT) for each parameter.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Override Transaction Behavior	Yes	Overrides the default behavior of a transaction group. If this activity is in a transaction group, the activity is normally committed or rolled back with the other transactional activities. If this check box is selected, this activity is not part of the transaction group and is committed when it completes. Checking this option uses a separate database connection to perform the activity and commit the SQL statement.
Interpret Empty String as Null	No	Specifies how empty strings in the activity's input elements should be handled. When the field is checked, nulls are sent to the database where empty strings are supplied. When the field is unchecked, empty strings are treated as zero-length strings. Whether you use this field or not, you can still use XPath to set input elements explicitly to null. Note: Many databases treat empty strings and nulls as the same, so this field does not affect how the database interprets empty strings.
Resultsets Use Schema	No	Specifies that the design-time must try to fetch the schema when a resultset is expected from a stored procedure call. When selected, TIBCO Designer tries to invoke the stored procedure with default values and get the metadata for the resultset. If the check box is not selected, or if TIBCO Designer does not get any metadata, the results can be displayed as 'unknown resultsets'.

Input

The input for this activity is dependent upon the input parameters of the database procedure.

Input Item	Datatype	Description
ServerTimeZone	string	Specifies the time zone for the database server. for more information about how time zones are constructed using this field, see Time Zone Construction. You can use the Constants tab in the XPath Formula Builder to supply a valid time zone name. The list of time zone names is generated from <code>java.util.TimeZone.getAvailableIDs()</code>. For the list of valid time zone names, see Time Zone Names.
inputSet	complex	The input parameters of the database procedure. The datatype contained in this element varies depending upon the parameters specified for the database procedure.
timeout	number	Time (in seconds) to wait for the procedure call to complete. If the call does not complete in the given time limit, an error is returned.
maxRows	number	The maximum number of rows to retrieve. To retrieve all rows, specify 0.

Output

The output for the activity is dependent upon the output parameters of the database procedure.

Output Item	Datatype	Description
outputSet	complex	The output of the database procedure. The datatype contained in this element varies depending upon what the database procedure returns.
unknownResultSets	complex	The output of the database procedure may return other result sets. These result sets are returned in this

Output Item	Datatype	Description
		output item as an XML string. Create a Schema resource to describe these result sets, then use the Parse XML activity to parse the XML based on the schema. You can then use the parsed output in subsequent activities. For a more detailed description of this procedure, see <i>TIBCO ActiveMatrix BusinessWorks™ Release Notes</i>.
Elapsed Time	long	This field displays the time taken for the execution of the activity in milliseconds. This is an optional element that appears only when the property <code>java.property.bw.activity.output.stats.elapsedTime.JDBCCallActivity</code> is set to true in <code>designer.tra</code>. For run-time, set the property to true in the <code>bwengine.tra</code> file. By default, the property is set to false. If set in the design-time, this property must be carried forward to the run-time; else it can cause run-time errors. Enabling this property causes additional performance overhead. Hence, it is recommended that you use this property only when this information is to be consumed by some other ActiveMatrix BusinessWorks activity. Also, it is recommended that this property be used only in a non-production environment as a diagnostic tool and the property be turned off at run-time. You can turn off the property and not calculate the elapsed time at run-time by setting the property <code>java.property.bw.activity.output.stats.elapsedTime.turnoff</code> in the <code>bwengine.tra</code> file. However, note that setting this property turns off the calculation of elapsed time at run-time for all the JDBC activities.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
JDBCDriverLoadingException	An error occurred when loading the specified JDBC driver.
JDBCConnectionNotFoundException	An error occurred when attempting to connect to the specified database.
InvalidTimeZoneException	An invalid timezone was specified. For a list of valid timezones, see Time Zone Names .
JDBCSQLException	A SQL error occurred.
JDBCTimedOutException	The operation has timed out.
LoginTimedOutException	A timeout has occurred while attempting to connect to the database.

JDBC Connection

Shared Configuration



The JDBC Connection resource describes a JDBC connection. JDBC connections are used when specifying activities from the **JDBC** palette.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.
Connection Type	No	Specifies the kind of JDBC connection you want to create. The connection type can be one of the following: • JDBC • JNDI • XA The type of connection determines the other configuration fields that appear.

JDBC Connection Type Configuration Fields

JDBC Driver	Yes	The name of the JDBC driver class. You can select from a list of supported drivers. If you select a supported driver, the Database URL field is populated with a template for the URL of the driver. For more information about database drivers, see Database Drivers.
Database URL	Yes	The URL to use to connect to the database. A template of the URL is supplied for the selected JDBC driver. You must supply the portions of the URL that are in angle brackets (<>). For example, if you select the <code>oracle.jdbc.driver.OracleDriver</code> (thin), the following appears in the Database URL field: <pre>jdbc:oracle:thin:@<host>:<port#>:<db_instancename></pre> You must supply the host, port number, and database instance name in the URL.

Field	Global Var?	Description
Maximum Connections	No	The maximum number of database connections to allocate. The default maximum is 10. The minimum value that can be specified is 1. For more information about this field, see Connection Pooling.
User Name	Yes	User name to use when connecting to the database.
Password	Yes	Password to use when connecting to the database.
Login Timeout	Yes	Time (in seconds) to wait for a successful database connection. Only JDBC drivers that support connection timeouts can use this configuration field. If the JDBC driver does not support connection timeouts, the value of this field is ignored. Most JDBC drivers support connection timeouts. You can also set this timeout with the <code>Config.JDBC.Connection.SetLoginTimeout</code> custom engine property. The value of the property in the configuration file overrides the value of this field. For more information about setting custom engine properties, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i>.
License Key	Yes	Enter a valid license key provided for the Progress DataDirect driver. This value is necessary only when you are using Progress DataDirect driver.

JNDI Connection Type Configuration Fields

Use Shared JNDI Configuration	No	When this field is checked, the JNDI Configuration field appears, allowing you to choose a JNDI Configuration shared configuration resource. When this field is unchecked, the JNDI Context Factory, JNDI Context URL, JNDI User Name, and JNDI Password fields appear.

Field	Global Var?	Description
JNDI Configuration	No	This field only appears when the Use Shared JNDI Configuration field is checked. This field allows you to choose a JNDI Configuration shared configuration object that specifies the JNDI connection information.
JNDI Context Factory	No	This field only appears when the Use Shared JNDI Configuration field is unchecked. The initial context factory class for accessing JNDI. (javax.naming.Context.INITIAL_CONTEXT_FACTORY). You can choose from the drop down list of supported classes, or you can type in a different InitialContextFactory class name. When using IBM WebSphere, you should use the JNDI Configuration resource instead of specifying the JNDI configuration on this resource. For more information, see IBM WebSphere and JNDI Context Caching. Note: ActiveMatrix BusinessWorks attempts to find the class. However, you may need to add the Java file supplied by your JNDI service provider to the CLASSPATH environment variable to use JNDI.
JNDI Context URL	Yes	This field only appears when the Use Shared JNDI Configuration field is unchecked. This is the URL to the JNDI service provider (javax.naming.Context.PROVIDER_URL). An example URL is provided when one of the supported JNDI context factory classes is selected. For the syntax of the URL, see your JNDI provider documentation.
JNDI User Name	Yes	This field only appears when the Use Shared JNDI Configuration field is unchecked. Specify the user name to use when logging into the JNDI server (javax.naming.Context.SECURITY_PRINCIPAL). If the JNDI provider does not require access

Field	Global Var?	Description
		control, this field can be empty.
JNDI Password	Yes	This field only appears when the Use Shared JNDI Configuration field is unchecked. Specify the password for logging into the JNDI server (javax.naming.Context.SECURITY_CREDENTIALS). If the JNDI provider does not require access control, this field can be empty.
JNDI DataSource Name	Yes	The JNDI name specified for the DataSource.

XA Connection Type Configuration Fields

XA DataSource Class	No	The XA DataSource class. Note: ActiveMatrix BusinessWorks attempts to find the class. However, you may need to add the location of the class to the CLASSPATH environment variable.
Database URL	Yes	The URL to use to connect to the database. A template of the URL is supplied for the selected XA DataSource class. You must supply the portions of the URL that are in angle brackets (<>). For example, if you select the oracle.jdbc.xa.client.OracleXADataSource, the following appears in the Database URL field: <pre>jdbc:oracle:thin:@<host>:<port#>:<db_instancename></pre> You must supply the host, port number, and database instance name in the URL.
Maximum Connections	Yes	The maximum number of database connections to allocate. The default maximum is 10. The minimum value that can be specified is 1. For more information about this field, see Connection

Field	Global Var?	Description
Pooling.		
User Name	Yes	User name to use when connecting to the database.
Password	Yes	Password to use when connecting to the database.

Migrating existing JDBC applications configured with TIBCO Database Drivers Supplement

Starting from ActiveMatrix BusinessWorks 5.15.1 release, TIBCO® Database Drivers Supplement™ is not supported. Migrate any existing JDBC applications configured using the same to either vendor supplied native drivers or Progress DataDirect drivers procured with a valid license key.

Migrating from TIBCO Database Drivers Supplement configured application to native driver

If the Database URL is not configured using a Global Variable, pass the following property in `bwengine.tra` and `designer.tra` to apply the property for all applications deployed in that particular `TIBCO_HOME`:

```
java.property.bw.plugin.jdbc.<DBCCConnectionResourceName>.dburl= <Database Connection URL>
```

You can also apply this property specifically for an application by adding this property under `<TIBCO_HOME>\tra\domain\<domain_name>\application\<application_name>\<application_name>-Process_Archive.tra`.



Warning: Do not redeploy the application as it removes entries from the application's tra file.

When a Database URL is configured using a Global Variable, you can edit the Global variable value accordingly and redeploy the applications.

i Note: If you are working on a TIBCO_HOME which is upgraded from a previous version to TIBCO ActiveMatrix BusinessWorks 5.15.1, to upgrade applications to the latest TIBCO ActiveMatrix BusinessWorks, use Upgrade option on applications in Admin UI. For more information, see "Upgrading an application" in TIBCO BusinessWorks Administration.

i Note: If you are using JCONN3.jar to connect with a Sybase 16 database instance, use the `java.property.bw.plugin.jdbc.<JDBCConnectionResourceName>.driver=<Database Driver URL>` property additionally.

Migrating from TIBCO Database Drivers Supplement configured application to Progress DataDirect Driver:

Pass the following properties in `bwengine.tra` and `designer.tra` to apply the property for all applications deployed in that particular TIBCO_HOME.

- `java.property.bw.plugin.jdbc.<JDBCConnectionResourceName>.driver= <Database Driver URL>`
JDBCConnectionResourceName is the name of JDBC Connection Resource.
- `java.property.bw.plugin.jdbc.<JDBCConnectionResourceName>.dburl= <Database Connection URL>`
- `java.property.bw.plugin.jdbc.licenseKey=<License Key of Progress DataDirect driver>`

You can also apply these properties specifically for an application by adding them under `<TIBCO_HOME>\tra\domain\<domain_name>\application\<application_name>\<application_name>-Process_Archive.tra`.

Database Drivers

The following drivers are supported, but you must obtain, install, and configure these from the appropriate vendor:

- `oracle.jdbc.driver.OracleDriver (oci)`
- `oracle.jdbc.driver.OracleDriver (thin)`
- `org.postgresql.Driver`
- `com.sybase.jdbc2.jdbc.SybDriver`

- `com.timesten.jdbc.TimesTenDriver`
- `com.ibm.db2.jcc.DB2Driver`
- `com.mysql.jdbc.Driver`
- `com.microsoft.sqlserver.jdbc.SQLServerDriver`

The following drivers are JDBC 2.0-compliant and should work correctly, but they are not officially supported with ActiveMatrix BusinessWorks. You must obtain, install, and configure the driver from the appropriate vendor:

- `sun.jdbc.odbc.JdbcOdbcDriver`
- `weblogic.jdbc.mssqlserver4.Driver`

You can download, install, and use other JDBC drivers, but TIBCO does not guarantee the drivers function properly with ActiveMatrix BusinessWorks. To use an unsupported driver, perform the following:

Procedure

1. Download the JDBC driver and copy the required JAR files to a location in the classpath of the ActiveMatrix BusinessWorks Engine. For example, you can copy the files to the following location: `TIBCO_HOME/tpcl/<version>/jdbc`.
2. In the JDBC Connection resource, choose **Select Driver** from the drop-down list in the JDBC Driver field and type in the name of the driver class in the text box within that field. Then, type in the correct URL in the Database URL field and specify the other fields in the resource.

For more information on the supported and tested versions of databases and database drivers, see *TIBCO ActiveMatrix BusinessWorks™ Release Notes*.

You can also use Progress DataDirect drivers with a valid License key. Perform the following steps:

1. Copy the Progress DataDirect driver JAR files to a location in the classpath of the TIBCO ActiveMatrix BusinessWorks Engine. For example, you can copy the files to the following location: `TIBCO_HOME/tpcl/<version>/native-jdbc`.
2. In the JDBC Connection resource, choose **Select Driver** from the drop-down list in the **JDBC Driver** field and type in the name of the driver class in the text box within that field.
3. Type in the correct URL in the **Database URL** field and specify the other fields in the resource.

Connection Pooling

ActiveMatrix BusinessWorks creates a pool of JDBC connections for every JDBC Connection shared resource that uses the JDBC connection type. The maximum size of this pool is specified by the Maximum Connections configuration field.

Activities that use this JDBC Connection resource are given a connection from the pool. Once the maximum number of connections is reached, activities requesting a connection cannot proceed. Once a connection is freed by an activity, the connection is returned to the pool.

Connections that are left open, get eventually time out and be closed. These connections can be reopened at a later time, until the maximum number of connections specified in this field is reached. If you want to configure a timeout value for these connections, you can set the `Engine.DBConnection.idleTimeout` property. For more information, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

If an activity detects a connection in the pool is invalid (for example, the database is restarted), the activity that uses the JDBC connection resource attempts to reestablish the connection.

All activities that are part of the same transaction, use the same connection in the connection pool. The first activity in a transaction attempts to reestablish an invalid connection. If a connection becomes invalid during a transaction, the transaction is rolled back and must be retried, if necessary.

Test Connection Button

The **Test Connection** button allows you to test that the specified configuration fields result in a valid connection to a database.

JDBC Get Connection

Activity



The **JDBC Get Connection** activity retrieves an object reference to a `JavaConnectionAccessor` object for the specified JDBC Connection from the connection pool. This allows you to use the optimized ActiveMatrix BusinessWorks database connection pool instead of specifying database configuration information in your Java code. This database connection can then be used within Java

activities to access the specified database.

For more information on using the **JDBC Get Connection** activity, see [Using The JDBC Connection in Java](#).

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JDBC Connection	Yes	Path to the shared configuration resource containing the JDBC connection information. For more information about JDBC resources, see JDBC Connection .
Override Transaction Behavior	No	Overrides the default behavior of a transaction group. If the activity that uses this JDBC connection is in a transaction group, the activity is normally committed or rolled back with the other transactional activities. If this check box is selected, the database connection is in auto commit mode, where each operation is committed as it is performed. You can disable the auto commit mode, if desired, and your code is then responsible for performing the commit or rollback operation before releasing the connection.

Input

This activity has no input.

Output

The output for the activity is dependent upon the output parameters of the database procedure.

Output Item	Datatype	Description
JavaConnectionAccess sor	Java Object	A Java object reference to a JavaConnectionAccessor object. This object can be passed as input to a Java activity where custom Java code can use the connection to perform database operations.
Elapsed Time	long	This field displays the time taken for the execution of the activity in milliseconds. This is an optional element that appears only when the property <code>java.property.bw.activity.output.stats.elapsedTime.JDBCGetConn ectionActivity</code> is set to true in <code>designer.tra</code>. For run-time, set the property to true in the <code>bwengine.tra</code> file. By default, the property is set to false. If set in the design-time, this property must be carried forward to the run-time; else it can cause run-time errors. Enabling this property causes additional performance overhead. Hence, it is recommended that you use this property only when this information is to be consumed by some other ActiveMatrix BusinessWorks activity. Also, it is recommended that this property be used only in a non-production environment as a diagnostic tool and the property be turned off at run-time. You can turn off the property and not calculate the elapsed time at run-time by setting the property <code>java.property.bw.activity.output.stats.elapsedTime.turnoff</code> in the <code>bwengine.tra</code> file. However, note that setting this property turns off the calculation of elapsed time at run-time for all the JDBC activities.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. for more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
JDBCSQLException	A SQL error occurred.
LoginTimedOutException	A timeout has occurred while attempting to connect to the database.

Using The JDBC Connection in Java

The database connection obtained with the **JDBC Get Connection** activity has the following restrictions:

- Checkpoints should not be taken after a **JDBC Get Connection** activity completes.
- Do not pass the `JavaConnectionAccessor` object reference to a subprocess that spawns a new process.

The **JDBC Get Connection** activity creates a `JavaConnectionAccessor` object reference that can be passed to an input parameter of a [Java Code](#) or [Java Method](#) activity. For more information about how to pass Java object references between activities, see [Passing Java Objects Between Java Code Activities](#).

`JavaConnectionAccessor` has the following methods:

- `java.sql.Connection getDBConnection();` – retrieves a database connection from the connection pool.
- `void releaseConnection();` – releases a database connection so that it can return to the connection pool.
- `boolean isTransactional();` – returns true if the activity using the JDBC connection is in a transaction group and the Override Transaction Behavior on the Configuration tab of the **JDBC Get Connection** activity is unchecked.
- `void releaseConnection(SQLException ex)` – releases the connection obtained by way of

the `getDBConnection()` method. Use this method instead of the standard `releaseConnection` method because this method detects if the SQL Exception passed as input is due to a connection problem. If the connection is no longer usable, it is destroyed and it is not returned to the connection pool.

When the connection is part of a transaction, the `JavaConnectionAccessor` object prevents you from committing or rolling back the transaction. The transaction completes when the transaction group either commits or rolls back. If the connection is part of an XA transaction group, the `enlist` and `delist` operations are performed by the `getDBConnection` and `releaseConnection` methods.

When the connection is not part of a transaction, the database connection is in auto commit mode, where each operation is committed as it is performed. You can disable the auto commit mode, if desired, and your code is then responsible for performing the commit or rollback operation before releasing the connection.



Warning: If your Java code does not release the connection, the connection is automatically released when the process instance terminates. This could have undesirable consequences.

Your code should check for SQL Exceptions returned when executing JDBC statements. If you encounter an unexpected SQL Exception, you can use the `releaseConnection` (`SQLException ex`) method to release the connection. This allows you to check if the connection is no longer valid. If the connection is no longer valid, your Java code should return an appropriate exception for subsequent activities so that transactions can be rolled back, if necessary.

JDBC Query

Activity



The **JDBC Query** activity performs the specified SQL `SELECT` statement.

If this activity is not part of a transaction group, the SQL statement is committed after the activity completes. If this activity is part of a transaction group, the SQL statement is committed or rolled back with the other JDBC activities in the group at the end of the transaction. For more information about creating groups for transactions, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

If you want to override the default behavior of transaction groups for certain JDBC activities in a transaction group, you can check the `Override Transaction Behavior` field

on the Advanced tab. This specifies that the activity is outside of the transaction and the SQL statement is committed when the activity completes, even if the activity is in a transaction group.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JDBC Connection	Yes	Path to the shared configuration resource containing the JDBC connection information. For more information about JDBC resources, see JDBC Connection .
SQL Statement	Yes	The SQL SELECT statement to perform in the database. You can use a wizard to build the query for the desired database. For more information about using the Query Designer wizard, see Query Designer .
Timeout (sec)	Yes	Time (in seconds) to wait for the query to complete. If the query does not complete in the given time limit, an error is returned.
Maximum Rows	Yes	The maximum number of rows to retrieve. To retrieve all rows, specify 0.
Prepared Parameters	No	For a description of this field, see Prepared Parameters .

Fetch Button

The **Fetch** button on this activity allows you to synchronize the activity with the contents of the database.

When you first configure a **JDBC** activity, you must click the **Fetch** button to retrieve the schema for the output result set before attempting to apply your changes with the **Apply** button or by saving your project. If you do not click the **Fetch** button before applying your changes, ActiveMatrix BusinessWorks displays an error dialog prompting you to first fetch the output schema.

After you have configured your activity, the **Fetch** button is useful when you make a change to the database while editing a process definition containing this activity in ActiveMatrix BusinessWorks. The **Fetch** button synchronizes with the database and changes the output schema, if necessary.

Prepared Parameters

If you want to replace static names in the SQL statement with process variables, you must edit the SQL Statement field and replace the variable items with a question mark (?). For example, if you want to specify a variable for the first column returned, you would specify the following SQL statement:

```
SELECT DEMO.ITEM.ITEM_ID FROM DEMO.ITEM  
WHERE DEMO.ITEM.ITEM_ID = ?
```

The ? replaces the value in the WHERE clause, and you can then provide input to the SQL statement from process data on the Input tab.

 **Note:** Editing the SQL statement disables the Design tab of the Query Designer wizard. Make sure you have designed the query you need before editing it and inserting question marks for process variables. If your question marks are in the WHERE clause, you can put those conditions into the Design tab in the Where fields of each column. This allows you to keep the Design tab available and still have process variables in your query. Keep in mind, ? is not valid SQL syntax, so checking the syntax of a query that contains question marks can return an invalid character error in the Query Designer wizard.

Note: With MySQL database, Query Designer wizard does not populate the table structure in the JDBC Query palette.

When you add or remove question marks in SQL statements, a warning appears to the right of the Prepared Parameters field ("SQL/ Prepared Parameter Count Mismatch") to alert you to add or remove corresponding prepared parameters. Each prepared parameter corresponds to the question mark in the same position in the SQL statement. That is, the first prepared parameter in the list corresponds to the first question mark, the second prepared parameter in the list corresponds to the second question mark, and so on. The warning is for informational purposes only, you must make sure the parameters in this field correctly correspond to the SQL statement.

You can optionally specify names for each prepared parameter. By default, the prepared parameters are named Prepared_Param_1, Prepared_Param_2, and so on.

You must supply the datatype of each parameter to the SQL statement, and this datatype is used in the input schema for the statement. You can use the up or down arrow keys to move a specified datatype to the previous or next parameter. You can also select a contiguous set of parameters to move them up or down, or delete them with the **delete** button.

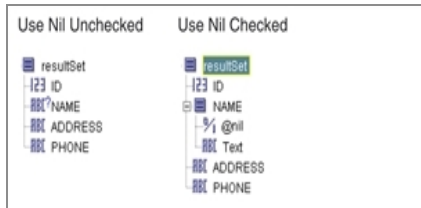
At run time, any parameters you place in the SQL statement are replaced with the mapped values for those parameters on the Input tab.

Note: Any mappings or expressions you create on the Input tab are not updated when you move or delete prepared parameters. You must manually update the Input tab after changing the prepared parameter field.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Override Transaction	Yes	Overrides the default behavior of a transaction group. If this activity is in a transaction group, the activity is normally

Field	Global Var?	Description
Behavior		committed or rolled back with the other transactional activities. If this check box is selected, this activity is not part of the transaction group and is committed when it completes. Checking this option uses a separate database connection to perform the activity and commit the SQL statement.
Use Nil	No	Specifies whether NULLs are represented as optional schema elements or whether each item that can contain a NULL has sub-items. For example, in the following schema, the NAME column can contain NULLs. If Use Nil is unchecked, the NAME element appears with a ? indicating it is optional (and if the column is NULL, the schema element is not included for that row). If Use Nil is checked, the NAME element is not optional, and it has two sub-elements, @nil and text(). The @nil element indicates whether the column value for the row is NULL, the text() element contains the column value when the column is not NULL.  The diagram shows two XML tree structures. On the left, 'Use Nil Unchecked', a 'resultSet' node contains an 'ID' node, a 'NAME' node with a trailing question mark, and 'ADDRESS' and 'PHONE' nodes. On the right, 'Use Nil Checked', a 'resultSet' node contains an 'ID' node, a 'NAME' node which itself contains an '@nil' node and a 'Text' node, and 'ADDRESS' and 'PHONE' nodes.
Interpret Empty String as Null	No	Specifies how empty strings in the activity's input elements should be handled. When the field is checked, nulls are sent to the database where empty strings are supplied. When the field is unchecked, empty strings are treated as zero-length strings. Whether you use this field or not, you can still use XPath to set input elements explicitly to null.

Field	Global Var?	Description
Note: Many databases treat empty strings and nulls as the same, so this field does not affect how the database interprets empty strings.		
Process In Subsets	No	Checking this field specifies that you would like to process the result set in smaller batches rather than processing the entire result set at once. When this field is checked, the <code>subsetSize</code> input element appears to allow you to specify the size of each batch of records you want to process. Also, the <code>lastSubset</code> output element appears and is set to true when the last batch of records is being processed. When this field is unchecked, the entire result set is returned. For more information, see Fetching Subsets of the Result Set.

Input

The input for the activity is the following.

Input Item	Datatype	Description
ServerTimeZone	string	Specifies the time zone for the database server. For more information about how time zones are constructed using this field, see Time Zone Construction. You can use the Constants tab in the XPath Formula Builder to supply a valid time zone name. The list of time zone names is generated from <code>java.util.TimeZone.getAvailableIDs()</code>. For the list of valid time zone names, see Time Zone Names.
Prepared_Param_*	varies	The list of prepared parameters supplied in the SQL statement. Prepared parameters allow you to

Input Item	Datatype	Description
(name corresponds to name specified in the Prepared Parameters field on the Configuration tab)		substitute process data for items in the SQL statement. See the description of the Prepared parameters field on the Configuration tab.
subsetSize	integer	This element is only available when the Process In Subsets field on the Advanced tab is checked. When that field is checked, records are processed in batches and this element specifies the number of records to process in each batch. This allows you to process smaller batches of rows instead of retrieving one large result set. This element must be used in conjunction with a Repeat Until True loop group for processing the result sets. For more information, see Fetching Subsets of the Result Set.
timeout	number	Time (in seconds) to wait for the query to complete. If the query does not complete in the given time limit, an error is returned.
maxRows	number	The maximum number of rows to retrieve. To retrieve all rows, specify 0.

Fetching Subsets of the Result Set

When the result set of a query is very large, you can retrieve subsets of the result set and iterate until the entire result set is processed. To retrieve subsets, you must use a Repeat Until True loop group to iterate until the entire result set is processed. Use the following procedure as a guideline for creating your Repeat Until True loop group.

Procedure

1. Create a **JDBC Query** activity.

2. Specify the fields on the Configuration tab and create a query that returns multiple rows.
3. Check the Process In Subsets field on the Advanced tab.
4. Set the subsetSize input item for the **JDBC Query** activity to the number of records you want to process for each execution of the loop. You could specify a constant, or you could use a global variable to store the value. Using a global variable has the advantage that you can change the value of the variable at deployment time. For example, create a global variable named BATCH_SIZE, then specify the chunkSize input element as `$_globalVariables/pfx:GlobalVariables/BATCH_SIZE`.
5. Select the **JDBC Query** activity and click the group icon on the toolbar to create a group containing the activity. For more information about working with groups, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.
6. Select Repeat Until True as the Group action, and specify an index name (for example, i).
7. The loop should exit when the entire result set has been consumed. The output element `lastSubset` is a boolean that is set to true when the last subset of records is being processed. For example, the condition for the loop could be set to the following:

```
$JDBCQuery/resultSet/lastSubset = "true"
```

The procedure above is a general guideline for creating a loop group for processing a large set of records. You may want to modify the procedure to include additional processing of the records, or you may want to change the XPath expressions to suit your business process.

For an example of processing subsets of a larger result set, see the examples included in the ActiveMatrix BusinessWorks installation directory .

Output

The output for the activity is the following.

Output Item	Datatype	Description
resultSet	complex	The root class for the output of the JDBC Query activity. This class contains all output items for the activity.
Record	complex	The result of the database query. The contents of the Record element depends upon the columns returned by the query.
lastSubset	boolean	This element only appears when the Process In Subsets field on the Advanced tab is checked. This field is set to true when the current subset of records being processed is the last subset of records. You can use this element to create an XPath expression to determine when a Repeat Until True loop that is processing records in batches should stop processing. For example: <pre>\$JDBCQuery/resultSet/lastSubset = "true"</pre>
Elapsed Time	long	This field displays the time taken for the execution of the activity in milliseconds. This is an optional element that appears only when the property <code>java.property.bw.activity.output.stats.elapsedTime.JDBCQueryActivity</code> is set to true in <code>designer.tra</code>. For run-time, set the property to true in the <code>bwengine.tra</code> file. By default, the property is set to false. If set in the design-time, this property must be carried forward to the run-time; else it can cause run-time errors. Enabling this property causes additional performance overhead. Hence, it is recommended that you use this property only when this information is to be consumed by some other ActiveMatrix BusinessWorks activity. Also, it is recommended that this property be used only in a non-production environment as a diagnostic tool and the property be turned off at run-time.

Output Item	Datatype	Description
		You can turn off the property and not calculate the elapsed time at run-time by setting the property <code>java.property.bw.activity.output.stats.elapsedTime.turnoff</code> in the <code>bwengine.tra</code> file. However, note that setting this property turns off the calculation of elapsed time at run-time for all the JDBC activities.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
<code>JDBCDriverLoadingException</code>	An error occurred when loading the specified JDBC driver.
<code>InvalidSQLException</code>	A parameter's datatype does not match the datatype of the table column.
<code>JDBCConnectionNotFoundException</code>	An error occurred when attempting to connect to the specified database.
<code>DuplicatedFieldNameException</code>	A duplicate field name was specified.
<code>InvalidTimeZoneException</code>	An invalid timezone was specified. For a list of valid timezones, see Time Zone Names .
<code>JDBCSQLException</code>	A SQL error occurred.
<code>JDBCTimedOutException</code>	The operation has timed out.
<code>LoginTimedOutException</code>	A timeout has occurred while attempting to connect to the database.

JDBC Update

Activity



The **JDBC Update** activity performs the specified SQL INSERT, UPDATE, or DELETE statement.

If this activity is not part of a transaction group, the SQL statement is committed after the activity completes. If this activity is part of a transaction group, the SQL statement is committed or rolled back with the other JDBC activities in the group at the end of the transaction. For more information about creating groups for transactions, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

If you want to override the default behavior of transaction groups for certain JDBC activities in a transaction group, you can check the Override Transaction Behavior field on the Advanced tab. This specifies that the activity is outside of the transaction and the SQL statement is committed when the activity completes, even if the activity is in a transaction group.

Configuration

The Configuration tab has the following fields:

Field	Global Var?	Description
Name	No	The name to appear as the label for this field in the process definition.
Description	No	Short description of the activity.
JDBC Connection	Yes	Path to the shared configuration resource containing the JDBC connection information. For more information, see JDBC Connection .
SQL Statement	Yes	The SQL INSERT, UPDATE, or DELETE statement to perform.
Timeout (sec)	Yes	The time to wait for the statement to complete.

Field	Global Var?	Description
Prepared Parameters	No	For a description of this field, see Prepared Parameters .

Prepared Parameters

If you want to replace static names in the SQL statement with process variables, you must edit the SQL statement field and replace the variable items with a question mark (?). For example, you might specify the following statement:

```
UPDATE emp SET ename = ?, phone = ?  
WHERE id = ?
```

The ? replaces the value, and you can then provide input to the SQL statement from process data on the input tab.

When you add or remove question marks in SQL statements, a warning appears to the right of the Prepared Parameters field ("SQL/ Prepared Parameter Count Mismatch") to alert you to add or remove corresponding prepared parameters. Each prepared parameter corresponds to the question mark in the same position in the SQL statement. That is, the first prepared parameter in the list corresponds to the first question mark, the second prepared parameter in the list corresponds to the second question mark, and so on. The warning is for informational purposes only, you must make sure the parameters in this field correctly correspond to the SQL statement.

You can optionally specify names for each prepared parameter. By default, the prepared parameters are named Prepared_Param_1, Prepared_Param_2, and so on.

You must supply the datatype of each parameter to the SQL statement, and this datatype is used in the input schema for the statement. You can use the up or down arrow keys to move a specified datatype to the previous or next parameter. You can also select a contiguous set of parameters to move them up or down, or delete them with the delete button.

At run time, any parameters you place in the SQL statement are replaced with the mapped values for those parameters.

i Note: Any mappings or expressions you create on the Input tab are not updated when you move or delete prepared parameters. You must manually update the Input tab after changing the prepared parameter field.

For INSERT and DELETE statements, all prepared parameters appear as required items in the Input tab. For UPDATE statements, all prepared parameters appear as optional in the Input tab, but you must supply a value for at least one column in the statement or a runtime error occurs.

Advanced

The Advanced tab has the following fields:

Field	Global Var?	Description
Override Transaction Behavior	Yes	Overrides the default behavior of a transaction group. If this activity is in a transaction group, the activity is normally committed or rolled back with the other transactional activities. If this check box is selected, this activity is not part of the transaction group and is committed when it completes. Checking this option uses a separate database connection to perform the activity and commit the SQL statement.
Interpret Empty String as Null	No	Specifies how empty strings in the activity's input elements should be handled. When the field is checked, nulls are sent to the database where empty strings are supplied. When the field is unchecked, empty strings are treated as zero-length strings. Whether you use this field or not, you can still use XPath to set input elements explicitly to null. Note: Many databases treat empty strings and nulls as the same, so this field does not affect how the database interprets empty strings.

Field	Global Var?	Description
Batch Update	No	Checking this field signifies you want to perform multiple statements by supplying an array of records as input to the activity where each record matches the prepared parameters for the statement. The statements are performed in one batch at the end of the activity's execution. This field is only meaningful if there are prepared parameters in the SQL statement (see Prepared Parameters). If this field is not checked, the expected input is the list of prepared parameters for the statement (no array of records). Some database drivers may not support batch updates. A JDBCSQLException is thrown if the database driver does not support batch updates.

Input

The input for the activity is the following:

Input Item	Datatype	Description
Record	repeating	When batch updates are performed, the list of prepared parameters are contained in a repeating element. When only one update is performed, the list of prepared parameters appears as individual input elements. For more information about batch updates, see the description of the Batch Update field on the Advanced tab.
Prepared_Param_* (name corresponds to name specified in the Prepared Parameters field on	varies	The list of the prepared parameters supplied in the SQL statement. Prepared parameters allow you to substitute process data for items in the SQL statement. See the description of the Prepared Parameters field on the Configuration tab.

Input Item	Datatype	Description
the Configuration tab)		
ServerTimeZone	string	Specifies the time zone for the database server. For more information about how time zones are constructed using this field, see Time Zone Construction. You can use the Constants tab in the XPath Formula Builder to supply a valid time zone name. The list of time zone names is generated from <code>java.util.TimeZone.getAvailableIDs()</code>. For the list of valid time zone names, see Time Zone Names .
timeout	number	Time (in seconds) to wait for the statement to complete.

Output

The output for the activity is the following:

Output Item	Datatype	Description
noOfUpdates	integer	The number of updates performed by the statement specified on the Configuration tab. This element can also return -2 that indicates, in the event of a batch update, that the updates were successful, but it is unknown how many updates were performed.
Elapsed Time	long	This field displays the time taken for the execution of the activity in milliseconds. This is an optional element that appears only when the property <code>java.property.bw.activity.output.stats.elapsedTime.JDBCUpdateActivity</code> is set to true in <code>designer.tra</code>. For run-time, set the property to

Output Item	Datatype	Description
		true in the <code>bwengine.tra</code> file. By default, the property is set to false. If set in the design-time, this property must be carried forward to the run-time; else it can cause run-time errors. Enabling this property causes additional performance overhead. Hence, it is recommended that you use this property only when this information is to be consumed by some other ActiveMatrix BusinessWorks activity. Also, it is recommended that this property be used only in a non-production environment as a diagnostic tool and the property be turned off at run-time. You can turn off the property and not calculate the elapsed time at run-time by setting the property <code>java.property.bw.activity.output.stats.elapsedTime.turnoff</code> in the <code>bwengine.tra</code> file. However, note that setting this property turns off the calculation of elapsed time at run-time for all the JDBC activities.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
<code>JDBCDriverLoadingException</code>	An error occurred when loading the specified JDBC driver.
<code>InvalidSQLException</code>	A parameter's datatype does not match the datatype of the table column.
<code>JDBCConnectionNotFoundException</code>	An error occurred when attempting to connect to the specified database.

Exception	Thrown When...
InvalidTimeZoneException	An invalid timezone was specified. For a list of valid timezones, see Time Zone Names .
JDBCSQLException	A SQL error occurred.
JDBCTimedOutException	The operation has timed out.
LoginTimedOutException	A timeout has occurred while attempting to connect to the database.

Query Designer

Wizard

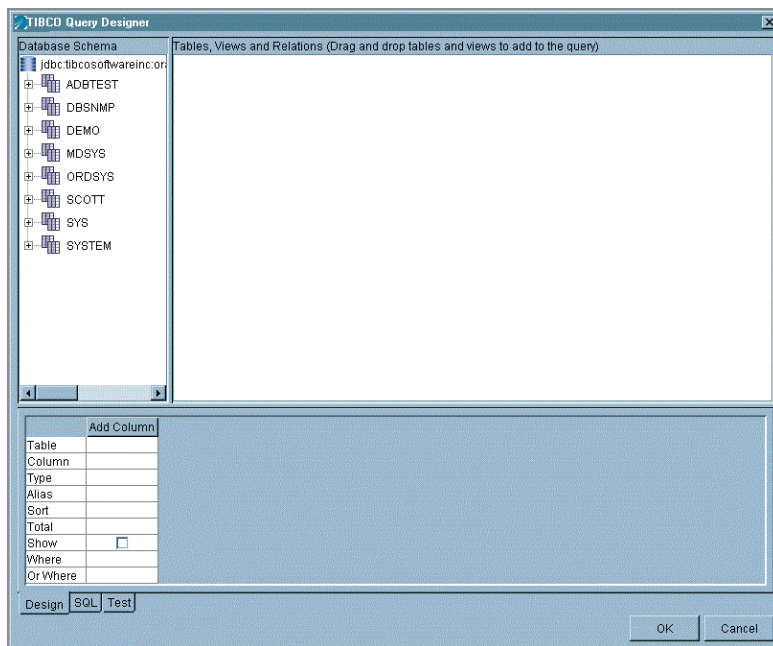
Note: Query Designer wizard supports Oracle, Microsoft SQL Server, and DB2. It does not work with Postgres, MariaDB, MySQL databases.

The Query Designer is a GUI tool for building SQL queries, testing syntax, and previewing results before running the process. You can use the Query Designer in one of the following ways:

- to automatically generate SQL queries
- to manually edit and check the syntax of existing queries

In the [JDBC Query](#) activity, you must first specify a [JDBC Connection](#) shared configuration resource to connect to a database. Once the database connection is specified in the **JDBC Query** activity, you can click the **Build Using Wizard** button to create a query using the Query Designer wizard. [The Query Designer wizard](#) illustrates the Query Designer.

Figure 18: The Query Designer wizard



The Query Designer window has the following elements:

- a Database Schema panel, where schema information for the specified database and user is displayed in a tree structure.
- a Table Diagram panel, where you can drag and drop tables from the Schema panel to build a SQL query.
- a SQL panel, where you can design, refine, view, and test the query. This panel has three tabs for designing, editing, and testing your queries.

i Note: Prepared parameters in ActiveMatrix BusinessWorks are inserted into a query by using the question mark (?) character. This is not valid SQL syntax, but ActiveMatrix BusinessWorks uses this syntax for inserting process data into the query at runtime. Because this is not valid SQL syntax, an error is returned if you attempt to check the syntax of a query that has prepared parameters. You should design and test the syntax of your query before inserting prepared parameters.

You can use the Design tab of the Query Designer to automatically generate queries. You can then click on the SQL tab of the Query Designer to manually edit the query, if you want. To manually edit the query, select the **Customize** check box on the SQL tab. On the SQL tab, you can drag and drop columns into the statement.



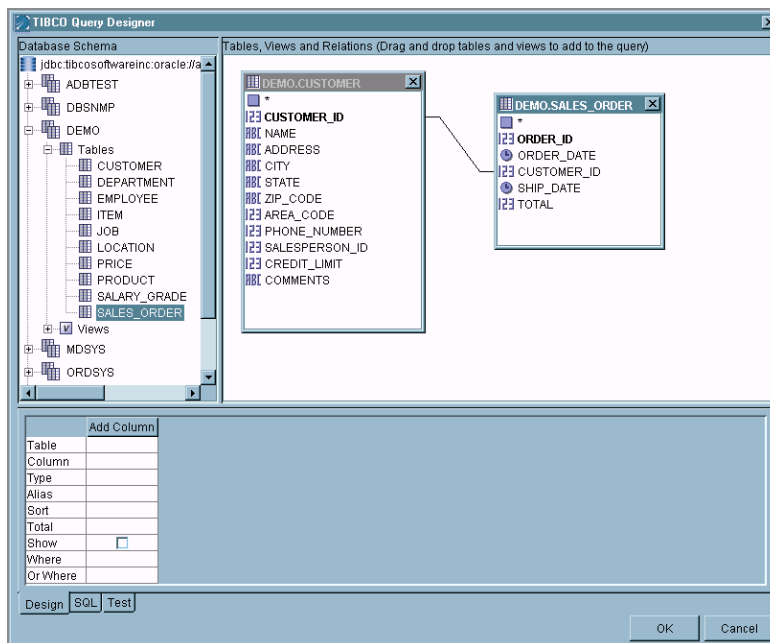
Warning: After you customize a query on the SQL tab, returning to the design tab without checking the **Customize** checkbox eliminates your changes. First, create a query by using the Design, then select the **Customize** checkbox on the SQL tab to further customize it as needed.

To use the Query Designer to generate a query, perform this procedure:

Procedure

1. Drag tables to include in the query from the Database Schema panel to the Table Diagram panel. Any joins defined between primary keys in the database are displayed.

For example:



This example demonstrates building a query that retrieves names of customers that also have placed orders. After dragging the CUSTOMER and SALES_ORDER tables from the Database Schema panel to the Table Diagram panel, a join is automatically displayed between the CUSTOMER_ID column in each table. You can also manually add joins by dragging the cursor from one join column to the other.

Clicking the SQL tab in the SQL panel shows the automatically generated syntax:

```
SELECT *
FROM DEMO.CUSTOMER, DEMO.SALES_ORDER
WHERE (DEMO.CUSTOMER.CUSTOMER_ID = DEMO.SALES_ORDER.CUSTOMER_ID)
```

By default, all table columns are included in the query. To constrain query syntax, click the Design tab in the SQL panel. You can drag individual columns from the Table Diagram panel to the Add Column area of this tab to refine the query. The Add Column entry is a placeholder only, and does not affect the output. It is meant to be the area where you can drag columns from the tables to add to the query.

The following example shows the results of dragging the CUSTOMER_ID, NAME, TOTAL, and SALESPERSON_ID columns to the SQL tab:

	CUSTOMER_ID	NAME	SALESPERSON_ID	TOTAL	Add Column
Table	DEMO.CUSTOMER	DEMO.CUSTOMER	DEMO.CUSTOMER	DEMO.SALES_ORDER	
Column	CUSTOMER_ID	NAME	SALESPERSON_ID	TOTAL	
Type	NUMBER	VARCHAR2	NUMBER	NUMBER	Drag and drop col
Alias					
Sort	None	None	None	None	
Total	None	None	None	None	
Show	☒	☒	☒	☒	☐
Where					
Or Where					

Clicking the SQL tab to view the modified syntax shows the query now selects only from the set of columns.

```
SELECT DEMO.CUSTOMER.CUSTOMER_ID, DEMO.CUSTOMER.NAME,
       DEMO.CUSTOMER.SALESPERSON_ID, DEMO.SALES_ORDER.TOTAL
FROM DEMO.CUSTOMER, DEMO.SALES_ORDER
WHERE (DEMO.CUSTOMER.CUSTOMER_ID =
DEMO.SALES_ORDER.CUSTOMER_ID)
```

2. To group sales by customer and show sales totals, perform the following:
 - For the TOTAL column, click in the Total field and select **Sum** from the list. The other columns are automatically updated to select the Group By value required for this function.
 - For the CUSTOMER_ID column, click the **Show** check box to deselect it. This column is necessary for the query, but can be hidden in the output.

Clicking the SQL tab to view the syntax shows the SUM function and GROUP BY clause has been added to the SQL:

```
SELECT DEMO.CUSTOMER.NAME, DEMO.CUSTOMER.SALESPERSON_ID,
SUM(DEMO.SALES_ORDER.TOTAL)
FROM DEMO.CUSTOMER, DEMO.SALES_ORDER
```

```
WHERE (DEMO.CUSTOMER.CUSTOMER_ID =
GROUP BY DEMO.CUSTOMER.CUSTOMER_ID, DEMO.CUSTOMER.NAME,
DEMO.CUSTOMER.SALESPERSON_ID
```

Other criteria can be applied using the Sort, Criteria, and Or fields.

3. To check the SQL syntax, click the **Check Syntax** button.

A dialog displays either a success message, or a database error code and message.

4. After checking the syntax, click the **Execute** button on the Test tab to preview the query results:

NAME	ADDRESS	SALESPERSON_ID	SUM(DEMO.SALES_ORDER.TOTAL)
JOCKSPORTS	345 VIEWRIDGE	7844	5285.9
TK&B SPORT SHOP	490 BOLI RD.	7521	101.4
VOLLYRITE	9722 HAMILTON	7654	27348
JUST TENNIS	HILLVIEW MALL	7521	806
EVERY MOUNTAIN	574 SURRY RD.	7499	7160.8
K + T SPORTS	3476 EL PASEO	7844	46370
SHAPE UP	908 SEQUOIA	7521	9076.8
WOMENS SPORTS	VALCO VILLAGE	7499	710
NORTH WOODS HEALTH AND FITNESS ...	98 LONE PINE WAY	7844	6400
STADIUM SPORTS	47 IRVING PL.	7557	16333.2

A representative number of rows, specified by the Maximum Rows value, is returned.

5. When the query returns the desired result set, click **OK**.

The SQL syntax is automatically entered in the SQL Statement field of the Database task.

The following usage restrictions apply to the Query Designer:

- You can build, modify, and test SQL queries only, not UPDATE statements or stored procedures.
- Only queries that are automatically generated can be manipulated and regenerated using the Design tab features. Query syntax that was manually typed in the SQL Statement field can be modified by typing or dropping column names on the SQL tab. The syntax checking and preview results features can also be used.
- After manually editing an automatically generated query on the SQL tab, your changes are lost if you uncheck the Customize box on the SQL tab and return to the Design tab.
- If you want to select columns with the same name from different tables, you must specify column aliases for the columns that have the same name. In SQL, you may

select columns with the same name from different tables in the same SQL statement, however, ActiveMatrix BusinessWorks requires that all names in an input or output schema are unique. Therefore, you must specify column aliases to ensure that all items in the output schema for a query are unique.

SQL Direct

Activity



The **SQL Direct** activity executes a SQL statement that you provide. This activity allows you to build a SQL statement dynamically (using other activities), then pass the SQL statement into this activity's input. This activity also allows you to execute SQL statements that are not supported by other activities in the **JDBC** palette. For example, DDL commands (for example, CREATE TABLE) are not available using any other activity.

If this activity is not part of a transaction group, the SQL statement is committed after the activity completes. If this activity is part of a transaction group, the SQL statement is committed or rolled back with the other JDBC activities in the group at the end of the transaction. For more information about creating groups for transactions, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

If you want to override the default behavior of transaction groups for certain JDBC activities in a transaction group, you can check the Override Transaction Behavior field on the Advanced tab. This specifies that the activity is outside of the transaction and the SQL statement is committed when the activity completes, even if the activity is in a transaction group.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.

Field	Global Var?	Description
Description	No	Short description of the activity.
JDBC Connection	Yes	Path to the shared configuration resource containing the JDBC connection information. For more information about JDBC resources, see JDBC Connection .
Timeout (sec)	Yes	Time (in seconds) to wait for the statement to complete. If the statement does not complete in the given time limit, an error is returned.
Maximum Rows	Yes	The maximum number of rows to retrieve. To retrieve all rows, specify 0.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Override Transaction Behavior	Yes	Overrides the default behavior of a transaction group. If this activity is in a transaction group, the activity is normally committed or rolled back with the other transactional activities. If this check box is selected, this activity is not part of the transaction group and is committed when it completes. Checking this option uses a separate database connection to perform the activity and commit the SQL statement.

Input

The input for the activity is the following.

Input Item	Datatype	Description
ServerTimeZone	string	Specifies the time zone for the database server. For more information about how time zones are constructed using this, see Time Zone Construction. You can use the Constants tab in the XPath Formula Builder to supply a valid time zone name. The list of time zone names is generated from <code>java.util.TimeZone.getAvailableIDs()</code>. For the list of valid time zone names, see Time Zone Names.
statement	string	The SQL statement you want to execute. You can use any process variables to build the statement dynamically, if desired.
timeout	number	Time (in seconds) to wait for the statement to complete. If the statement does not complete in the given time limit, an error is returned.
maxRows	number	The maximum number of rows to retrieve. To retrieve all rows, specify 0.

Output

The output for the activity is the following.

Output Item	Datatype	Description
noOfUpdates	integer	The number of updates performed by the statement. There could be more than one statement performed (for example, if a stored procedure is executed). Because of this, this item is a list of integers. Each item in the list corresponds to the number of updates performed by each statement. The first integer in the list is the number of updates performed by the first statement executed by this activity. The second integer is the

Output Item	Datatype	Description
		number of updates performed by the second statement, and so on.
unknownResultSet	complex	The result of the database statement. The result element depends upon what is returned by the statement. The result set is a list of zero or more rows with zero or more columns. Each column contains a name and a value element. You can use XPath expressions to retrieve the desired row and column from the result set. There could be multiple result sets returned by the statements executed by this activity. Because of this, this output item is a repeatable list of result sets. The first item in the list of result sets corresponds to the result set returned by the first statement executed by this activity. The second item in the list of result sets corresponds to the result set returned by the second statement executed by this activity, and so on.
Elapsed Time	long	This field displays the time taken for the execution of the activity in milliseconds. This is an optional element that appears only when the property <code>java.property.bw.activity.output.stats.elapsedTime.JDBCGeneralActivity</code> is set to true in <code>designer.tra</code>. For run-time, set the property to true in the <code>bwengine.tra</code> file. By default, the property is set to false. If set in the design-time, this property must be carried forward to the run-time; else it can cause run-time errors. Enabling this property causes additional performance overhead. Hence, it is recommended that you use this property only when this information is to be consumed by some other ActiveMatrix BusinessWorks activity.

Output Item	Datatype	Description
		Also, it is recommended that this property be used only in a non-production environment as a diagnostic tool and the property be turned off at run-time. You can turn off the property and not calculate the elapsed time at run-time by setting the property <code>java.property.bw.activity.output.stats.elapsedTime.turnoff</code> in the <code>bwengine.tra</code> file. However, note that setting this property turns off the calculation of elapsed time at run-time for all the JDBC activities.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
<code>JDBCDriverLoadingException</code>	An error occurred when loading the specified JDBC driver.
<code>JDBCConnectionNotFoundException</code>	An error occurred when attempting to connect to the specified database.
<code>InvalidTimeZoneException</code>	An invalid timezone was specified. For a list of valid timezones, see Time Zone Names .
<code>JDBCSQLException</code>	A SQL error occurred.
<code>JDBCTimedOutException</code>	The operation has timed out.
<code>LoginTimedOutException</code>	A timeout has occurred while attempting to connect to the database.

Time Zone Construction

ActiveMatrix BusinessWorks constructs any date and time data returned from the database with the time zone specified in the `ServerTimeZone` input element. Constructed `DateTime` data is in a UTC-normalized form, and daylight savings adjustments are also automatically applied.

If an invalid time zone name is specified, an `InvalidTimeZoneException` is thrown. If no value is specified in this input item, it is assumed that the database server is in the same time zone as the local machine.

All activities that retrieve and modify data to be later updated in a database should specify the same time zone for the database server or potentially incorrect values can get stored in the database.

Examples

Date in Database	Constructed DateTime
Dec/5/1972	no value
Jun/2/2003	no value
Dec/5/1972	
Jun/2/2003	
Dec/5/1972	
Jun/2/2003	

Time Zone Names

The `ServerTimeZone` input item in JDBC activities allows you to specify the name of the time zone where the database server is located. [Time zone names for the `ServerTimeZone` input item](#) lists the names of the time zones that you can specify. You

can also retrieve these time zone names on the Constants tab of the XPath Formula Builder.

Time zone names for the ServerTimeZone input item

Time Zone Name	Time Zone Name	Time Zone Name
MIT	Africa/Casablanca	Asia/Tehran
Pacific/Apia	Africa/Conakry	MET
Pacific/Niue	Africa/Dakar	Asia/Aqtau
Pacific/Pago_Pago	Africa/Freetown	Asia/Baku
America/Adak	Africa/Lome	Asia/Dubai
HST	Africa/Monrovia	Asia/Muscat
Pacific/Fakaofu	Africa/Nouakchott	Asia/Tbilisi
Pacific/Honolulu	Africa/Ouagadougou	Asia/Yerevan
Pacific/Rarotonga	Africa/Sao_Tome	Europe/Samara
Pacific/Tahiti	Africa/Timbuktu	Indian/Mahe
Pacific/Marquesas	Atlantic/Canary	Indian/Mauritius
AST	Atlantic/Faeroe	Indian/Reunion
America/Anchorage	Atlantic/Reykjavik	NET
Pacific/Gambier	Atlantic/St_Helena	Asia/Kabul
America/Los_Angeles	Europe/Dublin	Asia/Aqtobe
America/Tijuana	Europe/Lisbon	Asia/Ashgabat
America/Vancouver	Europe/London	Asia/Ashkhabad

Time Zone Name	Time Zone Name	Time Zone Name
PST	GMT	Asia/Bishkek
Pacific/Pitcairn	UTC	Asia/Dushanbe
America/Dawson_Creek	WET	Asia/Karachi
America/Denver	Africa/Algiers	Asia/Tashkent
America/Edmonton	Africa/Bangui	Asia/Yekaterinburg
America/Mazatlan	Africa/Douala	Indian/Chagos
America/Phoenix	Africa/Kinshasa	Indian/Kerguelen
MST	Africa/Lagos	Indian/Maldives
PNT	Africa/Libreville	PLT
America/Belize	Africa/Luanda	Asia/Calcutta
America/Chicago	Africa/Malabo	IST
America/Costa_Rica	Africa/Ndjamena	Asia/Katmandu
America/El_Salvador	Africa/Niamey	Antarctica/Mawson
America/Guatemala	Africa/Porto-Novo	Asia/Almaty
America/Managua	Africa/Tunis	Asia/Colombo
America/Mexico_City	Africa/Windhoek	Asia/Dacca
America/Regina	ECT	Asia/Dhaka
America/Tegucigalpa	Europe/Amsterdam	Asia/Novosibirsk
America/Winnipeg	Europe/Andorra	Asia/Thimbu

Time Zone Name	Time Zone Name	Time Zone Name
CST	Europe/Belgrade	Asia/Thimphu
Pacific/Easter	Europe/Berlin	BST
Pacific/Galapagos	Europe/Brussels	Asia/Rangoon
America/Bogota	Europe/Budapest	Indian/Cocos
America/Cayman	Europe/Copenhagen	Asia/Bangkok
America/Grand_Turk	Europe/Gibraltar	Asia/Jakarta
America/Guayaquil	Europe/Luxembourg	Asia/Krasnoyarsk
America/Havana	Europe/Madrid	Asia/Phnom_Penh
America/Indianapolis	Europe/Malta	Asia/Saigon
America/Jamaica	Europe/Monaco	Asia/Vientiane
America/Lima	Europe/Oslo	Indian/Christmas
America/Montreal	Europe/Paris	VST
America/Nassau	Europe/Prague	Antarctica/Casey
America/New_York	Europe/Rome	Asia/Brunei
America/Panama	Europe/Stockholm	Asia/Hong_Kong
America/Port-au-Prince	Europe/Tirane	Asia/Irkutsk
America/Porto_Acre	Europe/Vaduz	Asia/Kuala_Lumpur
America/Rio_Branco	Europe/Vienna	Asia/Macao
EST	Europe/Warsaw	Asia/Manila

Time Zone Name	Time Zone Name	Time Zone Name
IET	Europe/Zurich	Asia/Shanghai
America/Anguilla	ART	Asia/Singapore
America/Antigua	Africa/Blantyre	Asia/Taipei
America/Aruba	Africa/Bujumbura	Asia/Ujung_Pandang
America/Asuncion	Africa/Cairo	Asia/Ulaanbaatar
America/Barbados	Africa/Gaborone	Asia/Ulan_Bator
America/Caracas	Africa/Harare	Australia/Perth
America/Cuiaba	Africa/Johannesburg	CTT
America/Curacao	Africa/Kigali	Asia/Jayapura
America/Dominica	Africa/Lubumbashi	Asia/Pyongyang
America/Grenada	Africa/Lusaka	Asia/Seoul
America/Guadeloupe	Africa/Maputo	Asia/Tokyo
America/Guyana	Africa/Maseru	Asia/Yakutsk
America/Halifax	Africa/Mbabane	JST
America/La_Paz	Africa/Tripoli	Pacific/Palau
America/Manaus	Asia/Amman	ACT
America/Martinique	Asia/Beirut	Australia/Adelaide
America/Montserrat	Asia/Damascus	Australia/Broken_Hill
America/Port_of_Spain	Asia/Jerusalem	Australia/Darwin

Time Zone Name	Time Zone Name	Time Zone Name
America/Puerto_Rico	Asia/Nicosia	AET
America/Santiago	CAT	Antarctica/DumontDUrville
America/Santo_Domingo	EET	Asia/Vladivostok
America/St_Kitts	Europe/Athens	Australia/Brisbane
America/St_Lucia	Europe/Bucharest	Australia/Hobart
America/St_Thomas	Europe/Chisinau	Australia/Sydney
America/St_Vincent	Europe/Helsinki	Pacific/Guam
America/Thule	Europe/Istanbul	Pacific/Port_Moresby
America/Tortola	Europe/Kaliningrad	Pacific/Saipan
Antarctica/Palmer	Europe/Kiev	Pacific/Truk
Atlantic/Bermuda	Europe/Minsk	Australia/Lord_Howe
Atlantic/Stanley	Europe/Riga	Asia/Magadan
PRT	Europe/Simferopol	Pacific/Efate
America/St_Johns	Europe/Sofia	Pacific/Guadalcanal
CNT	Europe/Tallinn	Pacific/Kosrae
AGT	Europe/Vilnius	Pacific/Noumea
America/Buenos_Aires	Africa/Addis_Ababa	Pacific/Ponape
America/Cayenne	Africa/Asmera	SST
America/Fortaleza	Africa/Dar_es_Salaam	Pacific/Norfolk

Time Zone Name	Time Zone Name	Time Zone Name
America/Godthab	Africa/Djibouti	Antarctica/McMurdo
America/Miquelon	Africa/Kampala	Asia/Anadyr
America/Montevideo	Africa/Khartoum	Asia/Kamchatka
America/Paramaribo	Africa/Mogadishu	NST
America/Sao_Paulo	Africa/Nairobi	Pacific/Auckland
BET	Asia/Aden	Pacific/Fiji
America/Noronha	Asia/Baghdad	Pacific/Funafuti
Atlantic/South_Georgia	Asia/Bahrain	Pacific/Majuro
America/Scoresbysund	Asia/Kuwait	Pacific/Nauru
Atlantic/Azores	Asia/Qatar	Pacific/Tarawa
Atlantic/Cape_Verde	Asia/Riyadh	Pacific/Wake
Atlantic/Jan_Mayen	EAT	Pacific/Wallis
Africa/Abidjan	Europe/Moscow	Pacific/Chatham
Africa/Accra	Indian/Antananarivo	Pacific/Enderbury
Africa/Banjul	Indian/Comoro	Pacific/Tongatapu
Africa/Bissau	Indian/Mayotte	Pacific/Kiritimati

JMS Palette

Java Message Service (JMS) is a specification for how messages are sent and received between applications in a Java environment.

The **JMS** palette is used to send and receive JMS messages in a process definition. Both the JMS point-to-point (queues) and publish/subscribe (topics) models are supported.

Refer to the documentation of your JMS provider or the JMS specification for more information about JMS and its message models.

For more information about specific support for JMS providers, see *TIBCO ActiveMatrix BusinessWorks™ Release Notes*.

Enabling JMS in ActiveMatrix BusinessWorks

ActiveMatrix BusinessWorks acts as a JMS client when sending messages to or receiving messages from a JMS provider. Because of this, you must place the appropriate JMS client files supplied by the JMS provider in the proper location. Typically, this includes placing JAR files into a directory that is included on the client's CLASSPATH.

For more information about which files are needed and the location to place them to enable client access to the JMS server, see the documentation for your JMS provider.

Once you have located the appropriate files, you can place them in the `<TIBCO_HOME>/bw/<version>/lib` directory. This directory is in the ActiveMatrix BusinessWorks CLASSPATH, and placing the JMS client files in that directory allows you to design and test ActiveMatrix BusinessWorks process definitions that include **JMS** Palette activities.

When deploying your project, the machine where the process engine is deployed must also have the appropriate JMS client files. You should also place the JMS client files in the `<TIBCO_HOME>/bw/<version>/lib` directory of the machines where your process engines are deployed.

Get JMS Queue Message

Activity



The **Get JMS Queue Message** activity retrieves a message from the specified queue. This activity allows you to perform a receive operation on the queue as opposed to waiting for a queue message to be delivered to the [Wait for JMS Queue Message](#) activity or the [JMS Queue Receiver](#) process starter.

You can use the Message Selector field on the Advanced tab to retrieve a specific queue message from the queue.

The **Get JMS Queue Message** activity is different from the **Wait for JMS Queue Message** activity in the following ways:

- Unlike the Wait for activity, which starts listening for messages from the time the ActiveMatrix BusinessWorks engine starts, this activity starts listening for incoming messages on the specified queue from the time the activity is triggered.
- This activity can receive only one message from the specified queue at a time, when the Message Selector is not used.
- Once triggered, this activity can either gets a message from the specified destination queue name before timeout and proceeds or it throws a timeout error and exits.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JMS Connection	No	The JMS connection. For more information about creating a JMS connection, see JMS Connection .

Field	Global Var?	Description
Destination Queue	Yes	Name of the queue from which to retrieve the message. The syntax of the queue name is specific to the JMS provider you are using. For more information about queue names, see your JMS provider documentation. Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Message Type	No	The type of the message. This can be one of the following: • Simple – A message with no body portion. • Bytes – A stream of bytes. • Map – A set of name or value pairs. The names are strings, and the values are simple data types (JAVA primitives), an array of bytes (use the Binary datatype when mapping this data), or a string. Each item can be accessed sequentially or by its name. • Object – A serializable Java object. • Object Ref – An object reference to a Java object. • Stream – A stream of Java primitives, strings, or arrays of bytes. Each value must be read sequentially. • Text – The message is a java.lang.String. • XML Text – The message is XML text.
Acknowledge Mode	Yes	The acknowledge mode for incoming messages. It can be one of the following: • Auto – the message is automatically acknowledged when it is received.

Field	Global Var?	Description
		Client – the message get acknowledged at a later point by using the Confirm activity. If the message is not confirmed before the process instance ends, the message is redelivered and a new process instance is created to handle the new incoming message. Ensure that your process definition confirms the message when using this acknowledge mode.
Acknowledge Mode		- TIBCO EMS Explicit – (only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, can be redelivered instead of all messages in the session. The session is not blocked and one session handles all incoming messages for each process instance. Note: In the ActiveMatrix BusinessWorks 5.10 release, there is a change in the behavior of this feature. Now only the messages that are not confirmed, would be redelivered. - TIBCO EMS Explicit Dups OK – (only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, can be redelivered instead of all messages in the session. The session is not blocked and one session handles all the incoming messages for each process instance. However, the messages would be lazily acknowledged. - TIBCO EMS No Acknowledge – the message is not acknowledged when it is received. - Dups OK – the message is acknowledged automatically when it is received. JMS provides this mode for lazy acknowledgement, but ActiveMatrix BusinessWorks acknowledges messages upon

Field	Global Var?	Description
		receipt. Transactional – this mode is used when a transaction that can process JMS messages is included in the process definition. The message is acknowledged when the transaction commits. For more information about creating transactions that JMS activities can participate in, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Message Selector	Yes	A string to determine whether a message should be received. The syntax of the message selector is determined by the JMS provider, but it is usually a subset of SQL92 (where message properties are used instead of table column names). For more information and syntax for a message selector string, see your JMS provider documentation .
JMS Application Properties	No	Any application-specific message properties that is the part of the message. This is specified by the JMS Application Properties shared configuration resource. Once specified, these properties appear on the Output tab as "OtherProperties".

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
destinationQueue	string	The queue to which to send the request. This input item overrides the Destination Queue field on the Configuration tab.
timeout	integer	Specifies how long the message can remain active (in seconds). If set to 0, the message does not expire.
selector	string	A string to determine whether a message should be received. The value of this element overrides any value specified in the Message Selector field of the Advanced tab. The syntax of the message selector is determined by the JMS provider, but it is usually a subset of SQL92 (where message properties are used instead of table column names). For more information and syntax for a message selector string, see your JMS provider documentation.

Output Editor

The Output Editor tab defines the schema to use for messages of type Map, Stream, or XML Text. Map messages are name/value pairs, and the schema allows you to define the structure of the retrieved queue message. The schema defined on the Output Editor tab becomes the body of the message on the Output tab.

For a complete description of using the Output Editor tab, see [Specifying Data Schema](#).

Output

The output for the activity is the following.

Output Item	Datatype	Description
JMSHeaders	complex	The message header fields for the message. For more information about message header fields, see JMS Properties. Only properties applicable to this type of message are displayed.
JMSProperties	complex	The message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration resource.
Body	depends upon message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements. Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance. Value – Required. Value of the property with one

Output Item	Datatype	Description
		instance. Type – Optional. Type of the property. If not provided, it is considered as string. Following Types are supported: • string • boolean • short • integer • long • float • double • byte Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. The DynamicProperties is also added to the outgoing message.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When
JMSInvalidInputException	Input to the activity is not valid.

Exception	Thrown When
JMSSessionCreateException	The JMS session could not be created.
JMSReceiveException	The JMS receive operation failed.
ActivityTimeoutException	The specified timeout has been reached and the activity has not completed its execution.

JMS Application Properties

Shared Configuration



The JMS Application Properties resource describes any JMS message properties that a JMS application expects. These properties can then be added to a JMS message on the Advanced tab of a **JMS** activity. These properties appear in the OtherProperties element on the Input or Output tab of the activity.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.
Properties	No	A table listing any application-specific properties. Use the + and x buttons to the right of the table to add and delete properties. Use the up and down arrow buttons to move selected properties to the desired location in the table.

Field	Global Var?	Description
		Specify the name of the property by typing the name in the PropertyName column.
		Specify the datatype of the property by double clicking the cell of the Type column for the desired property. This causes a drop down list of valid JMS datatypes to appear.
		Specify whether the property is optional or required by double clicking on the cell of the Cardinality column for the desired property. Select either optional or required in the drop down list that appears.

JMS Connection

Shared Configuration



The JMS Connection resource describes a JMS connection. This resource is used to specify activities on the **JMS** palette.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	This name appears as the label for the resource.
Description	No	Short description of the shared resource.
User Name	Yes	User name to use when logging into the JMS server. If the JMS provider does not require access control, this

Field	Global Var?	Description
		field can be empty. Not all JMS servers require user names and passwords. Refer to your JMS provider documentation and consult your system administrator to determine if your JMS server requires a user name and password.
Password	Yes	Password to use when logging into the JMS server. If the JMS provider does not require access control, this field can be empty.
Auto-Generate Client ID	Yes	Checking this field specifies that you want to automatically generate the client ID if no client ID is specified or if the specified ID is already in use. When this field is checked, if a value is specified in the Client ID field, an attempt is made to use the specified value. However, if the specified value is already in use, a new client ID is generated. When an automatically generated client ID is required, the following format is used for the client ID: <i>BW-<code><projectName></code>-<code><topic or queue></code>-<code><engineName></code>-<code><timestamp></code></i> If this field is not checked, then the value specified in the Client ID field is used. If no value is specified in the Client ID field, then no client ID is set. If the value specified in the Client ID field is already in use, an exception is thrown.
Client ID	Yes	Client ID for the connection. Typically JMS providers have a provider-specific format for client IDs. For more information about client IDs, see your JMS provider's documentation. Each connection must use a unique Client ID. If you do not specify a value in this field, ActiveMatrix BusinessWorks ensures the uniqueness of the client ID for each connection.
SSL	No	Specifies whether you want to use SSL for the connection

Field	Global Var?	Description
		to the JMS server. SSL is used when the Use SSL? check box is selected. Click the Configure SSL button to configure the SSL connection parameters. Note: SSL is supported only when using TIBCO Enterprise Message Service. For more information, see SSL Configuration.
Use JNDI for Connection Factory	Yes	Specifies whether JNDI should be used to look up the ConnectionFactory object. If this field is unchecked, the Provider URL and Use XA Connection Factory fields appear. If this field is checked, JNDI configuration fields appear.
Provider URL	Yes	This field is only available when the Use JNDI for Connection Factory field is unchecked. This is the URL to use to connect to the JMS server.
Use XA Connection Factory	No	When checked, this field specifies that an XA connection factory is to be used. This is useful if you want activities that use this connection to participate in an XA transaction. Note: This field appears only when the Use JNDI for Connection Factory field is unchecked. For more information about transactions, see <i>TIBCO ActiveMatrix BusinessWorks Process Design</i>. Note: If this field is selected, the "Use UFO Connection Factory" field cannot be used.
Use UFO Connection Factory	Yes	Use this check box when using EMS Unshared State Failover factories. For more information about Unshared State Failover, see <i>TIBCO Enterprise Message Service</i>

Field	Global Var?	Description
		<i>User's Guide</i> . This field appears only when the Use JNDI for Connection Factory field is unchecked. Note: If this field is selected, the Use XA Connection Factory field cannot be used. When using Unshared Failover setup, if a connection loss is detected to server (A), ActiveMatrix BusinessWorks will try to connect to server (B) as defined in the connection factory configuration. For more information about Unshared Failover, see <i>TIBCO Enterprise Message Service User's Guide</i>. After enabling this check box you can pass the provider URL in the form of UFO. For example: "tcp://serverA:7222+tcp://serverB:7222" where serverA and serverB are two EMS servers configured for the Unshared State Failover. Unsupported Features and Activities • JMS Palette ◦ JMS Queue Requestor ◦ JMS Topic Requestor ◦ Any other Requestor activity • SOAP Palette ◦ All activities • Service Palette ◦ All activities • AE Adaptor ◦ All activities

Field	Global Var?	Description
		Looking up factories using JNDI is not supported because: • UFO factories cannot be retrieved through JNDI directly.
Use Shared JNDI Configuration	No	When this field is checked, the JNDI Configuration field appears that allows you to choose a JNDI Configuration shared configuration resource. When this field is unchecked, the JNDI Context Factory, JNDI Context URL, JNDI User Name, and JNDI Password fields appear.
JNDI Configuration	No	This field only appears when the Use Shared JNDI Configuration field is checked. This field allows you to choose a JNDI Configuration shared configuration object that specifies the JNDI connection information. For more information, see JNDI Configuration.
JNDI Context Factory	Yes	This field appears only when the Use Shared JNDI Configuration field is unchecked. It is the initial context factory class for accessing JNDI. (javax.naming.Context.INITIAL_CONTEXT_FACTORY). You can choose from the drop-down list of supported classes, or you can type in a different InitialContextFactory class name. When using IBM WebSphere, you should use the JNDI Configuration resource instead of specifying the JNDI configuration on this resource. For more information, see IBM WebSphere and JNDI Context Caching. Note: ActiveMatrix BusinessWorks attempts to find the class. However, you may need to add the Java file supplied by your JNDI service provider to the CLASSPATH environment variable to use JNDI.

Field	Global Var?	Description
JNDI Context URL	Yes	This field only appears when the Use Shared JNDI Configuration field is unchecked. This is the URL to the JNDI service provider (<code>javax.naming.Context.PROVIDER_URL</code>). An example URL is provided when one of the supported JNDI context factory classes is selected. For the syntax of the URL, see your JNDI provider documentation.
JNDI User Name	Yes	This field only appears when the Use Shared JNDI Configuration field is unchecked. User name to use when logging into the JNDI server (<code>javax.naming.Context.SECURITY_PRINCIPAL</code>). If the JNDI provider does not require access control, this field can be empty.
JNDI Password	Yes	This field only appears when the Use Shared JNDI Configuration field is unchecked. Password to use when logging into the JNDI server (<code>javax.naming.Context.SECURITY_CREDENTIALS</code>). If the JNDI provider does not require access control, this field can be empty.
Connection Factory SSL Password	Yes	This field is only available when the SSL and the Use JNDI for Connection Factory fields are checked and the Use Shared JNDI Configuration field is unchecked. The SSL configuration is specified in the <code>ConnectionFactory</code> object, except for the client SSL password. You can specify your client SSL password in this field, or you can leave this field empty. If your password is not specified, the private key password is used.
Note: Support for values given below is added for TIBCO ActiveMatrix BusinessWorks™ 5.16.1 HF-002 and above.		
OAuth Client ID	Yes	The Client ID issued to the client during the client

Field	Global Var?	Description
		registration process.
Client Secret	Yes	The Client Secret issued to the client during the client registration process.
Access Token URL	Yes	The endpoint for the authentication server. This is used to exchange the authorization code for the access token.
Grant Type	No	Client Credentials. Currently, only Client Credentials as a grant type is supported.
Authentication Type	No	Two types of authentication are supported: OAuth 2.0 and Basic Authentication. • If you select Basic Authentication, then the User Name and Password fields appear. • If you select OAuth 2.0, then the User Name, OAuth Client ID, Client Secret, Access Token URL and OAuth TLS fields appear.
OAuth TLS	No	Selecting this checkbox enables the support for generation access token from TLS based OAuth server when encrypting or decrypting Messages.
OAuth Disable Verify Hostname	No	Indicates whether the name on the server's certificate must be verified against the server's host name. If the server's host name is different from the name on the certificate, the TLS/SSL connection fails. The name on the certificate can be verified against another name by specifying OAuth Expected Hostname. Selecting this checkbox disables verification of OAuth Hostname. Deselecting this check box displays the OAuth Expected Hostname field. Default: This check box is selected.

Field	Global Var?	Description
Expected Hostname	Yes	Optional. The expected name of the remote host. The default is None.
Server Trust File	Yes	The location of the certificate. Following certificate extension supported in this File picker: pem crt cer cert

Test Connection Button

Allows you to test the connection specified in the configuration of this resource. This button is only enabled if JNDI is not used (that is, the Use JNDI for Connection Factory field is unchecked) or if the required JNDI fields on the Configuration and Advanced tab are supplied when JNDI is used.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Topic Connection Factory	Yes	This field is only available when the Use JNDI for Connection Factory field on the Configuration tab is checked. The TopicConnectionFactory object stored in JNDI. This object is used to create a topic connection with a JMS application. For more information about creating and storing TopicConnectionFactory objects, see your JNDI provider documentation.
Queue	Yes	This field is only available when the Use JNDI for Connection

Field	Global Var?	Description
Connection Factory		Factory field on the Configuration tab is checked. The QueueConnectionFactory object stored in JNDI. This object is used to create a queue connection with a JMS application. For more information about creating and storing QueueConnectionFactory objects, see your JNDI provider documentation.
JNDI Properties	No	Any additional properties to supply for the connection. You specify a name, datatype, and value for each property. These properties are typically vendor-specific. For more information about the available properties, see your JNDI provider documentation.

SSL Configuration

The **SSL Configuration** button allows you to configure the SSL connection parameters. The SSL Configuration dialog specifies the basic and advanced SSL parameters you can set for the JMS server you are connecting to.



Note: When using JNDI to lookup the JMS Connection factory, the parameters `ssl_identity` and `ssl_verify_host` must be specified in the `factories.conf` file of the Enterprise Message Service server.

The following sections describe the SSL Configuration dialog.

Basic

The Basic tab of the SSL Configuration dialog allows you to specify the simplest required SSL configuration parameters for the connection.

Field	Description
Trusted Certificates Folder	Location of the trusted certificates on this machine. The trusted certificates are a collection of certificates from servers to whom you establish connections. If the server you want to establish a connection presents a certificate that does not match one of your trusted certificates, the connection is refused. This prevents connections to unauthorized servers. Trusted certificates must be imported into a folder, and then you can select the folder in this field.
Identity	The location of the client certificate. This is a resource contained in the General palette. Specify the client certificate and Identity field when the JMS server requires client authentication. For more information, see <i>TIBCO Designer™ Palette Reference</i>.

Advanced

The Advanced tab of the SSL Configuration dialog allows you to specify more advanced SSL configuration parameters for the connection.

Field	Description
Trace	Specifies whether SSL tracing should be enabled during the connection. If checked, the SSL connection messages are logged and sent to the console.
Debug Trace	Specifies whether SSL debug tracing should be enabled during the connection. Debug tracing provides more detailed messages than standard tracing.
Verify Host Name	Specifies whether you want to verify that the host you are connecting to is the expected host. The host name in the host's digital certificate is compared against the value you supply in the Expected Host Name field. If the host name does not match the expected host name, the connection is refused.

Field	Description
	Note: The default context factories for TIBCO Enterprise Message Service automatically determine if host name verification is necessary. If you are using a custom implementation of the context factories, your custom implementation must explicitly set the verify host property to the correct value. For example: <pre>com.tibco.tibjms.TibjmsSSL.setVerifyHost(false)</pre>
Expected Host Name	Specifies the name of the host you are expecting to connect to. This field is only relevant if the Verify Host Name field is also checked. If the name of the host in the host's digital certificate does not match the value specified in this field, the connection is refused. This prevents hosts from attempting to impersonate the host you are expecting to connect to.
Strong Cipher Suites Only	When checked, this field specifies that the minimum strength of the cipher suites used can be specified with the <code>bw.plugin.security.strongcipher.minstrength</code> custom engine property. For more information about this property, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i>. The default value of the property disables cipher suites with an effective key length below 128 bits. When this field is unchecked, only cipher suites with an effective key length of up to 128 bits can be used.

JMS Properties

[Properties for JMS Messages](#) table describes message headers and message properties used in JMS messages. MessageHeader properties are set by the JMS application sending the message and are available to view once the message is received. MessageProperties can be set on outgoing messages using the Input tab of the activity that sends the message.

Properties for JMS Messages

Property	Datatype	Description
MessageHeaders		
JMSDestination	string	Destination where a message is sent.
JMSReplyTo	string	The JMS application where the reply message should be sent.
JMSDeliveryMode	string	The delivery mode of the message. Can be one of the following: • Persistent signifies the messages are stored and forwarded. • Non-Persistent messages are not stored and may be lost due to failures in transmission. • When using TIBCO Enterprise Message Service, the JMSDeliveryMode header can be TIBCO EMS Reliable. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
JMSMessageID	string	The unique ID of the message.
JMSTimestamp	long	The time a message was handed off to a JMS provider to be sent. The time is expressed as the amount of time, in milliseconds, since midnight, January 1, 1970, UTC.
JMSExpiration	long	The expiration time of the message. The time is expressed as the amount of time, in milliseconds since midnight, January 1, 1970, UTC. If set to 0, the message does not expire. For outgoing messages, this is set by specifying

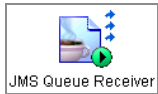
Property	Datatype	Description
		the JMSExpiration field on the activity's Advanced tab.
JMSRedelivered	boolean	Typically this item is set when the JMS provider has delivered the message at least once before.
JMSPriority	integer	The priority of the message. Priority is a value from 0-9. Higher numbers signify a higher priority (that is, 9 is a higher priority than 8). For outgoing messages, the default priority is 4. This is set on the activity's Advanced tab.
JMSCorrelationID	string	This ID is used to link a response message with its related request message. This is usually the message ID of a request message when this field is found in a reply message. For more information about JMSCorrelationID, see the following topics: • JMS Queue Requestor • JMS Queue Sender • JMS Topic Publisher • JMS Topic Requestor • Reply To JMS Message
JMSType	string	The JMS provider-supplied string to describe the type of the message. Some JMS providers use this property to define messages in the provider's repository. For more information about the use of this field, see your JMS provider documentation .
MessageProperties		
JMSXUserID	string	The ID of the user sending the message.

Property	Datatype	Description
		This property is optionally set on incoming messages by the JMS application. This property is not set on outgoing messages.
JMSXAppID	string	The ID of the application sending the message. This property is optionally set on incoming messages by the JMS application. This property is not set on outgoing messages.
JMSXProducerTXID	string	The transaction identifier of the transaction where this message was produced. This property is optionally set on incoming messages by the JMS application. This property is not set on outgoing messages.
JMSXConsumerTXID	string	The transaction identifier of the transaction where this message was consumed. This property is optionally set on incoming messages by the JMS application. This property is not set on outgoing messages.
JMSXRcvTimestamp	integer	The time the JMS server delivered the message to the consumer. The time is expressed as the amount of time, in milliseconds since midnight, January 1, 1970, UTC. This property is optionally set on incoming messages by the JMS server. This property is not set on outgoing messages.
JMSXDeliveryCount	integer	The number of delivery attempts for this message.
JMSXGroupID	string	The identity of the message group this message is a part of.

Property	Datatype	Description
		You can set this property on outgoing messages to group messages into a numbered sequence.
JMSXGroupSeq	integer	The sequence number of this message in its group.
		You can set this property on outgoing messages to group messages into a numbered sequence.

JMS Queue Receiver

Process Starter



Starts a process based on the receipt of a message for the specified JMS queue.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JMS Connection	No	The JMS connection. For more information about creating a JMS connection, see JMS Connection .
Destination Queue	Yes	Name of the queue of the incoming message. The syntax of the queue name is specific to the JMS provider you are using. For more information about queue

Field	Global Var?	Description
		names, see your JMS provider documentation.
		Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Message Type	No	The type of the message. This can be one of the following: • Simple – A message with no body portion. • Bytes – A stream of bytes. • Map – A set of name/value pairs. The names are strings, and the values are simple data types (Java primitives), an array of bytes (use the Binary datatype when mapping this data), or a string. Each item can be accessed sequentially or by its name. • Object – A serializable Java object. • Object Ref – An object reference to a Java object. • Stream – A stream of Java primitives, strings, or arrays of bytes. Each value must be read sequentially. • Text – The message is a java.lang.String. • XML Text – The message is XML text.
Acknowledge Mode	Yes	The acknowledge mode for incoming messages. It can be one of the following: • Auto – the message is automatically acknowledged when it is received. • Client – the message gets acknowledged at a later point by using the Confirm activity. If the message is

Field	Global Var?	Description
		not confirmed before the process instance ends, the message is redelivered and a new process instance is created to handle the new incoming message. Ensure that your process definition confirms the message when using this acknowledge mode.
		- TIBCO EMS Explicit – (only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, can be redelivered instead of all messages in the session. The session is not blocked and one session handles all incoming messages for each process instance. Note: In the ActiveMatrix BusinessWorks 5.10 release, there is a change in the behavior of this feature. Now only the messages that are not confirmed, can be redelivered. - TIBCO EMS Explicit Dups OK – (only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, can be redelivered instead of all messages in the session. The session is not blocked and one session handles all the incoming messages for each process instance. The messages however, would be lazily acknowledged. - TIBCO EMS No Acknowledge – the message is not acknowledged when it is received. - Dups OK – the message is acknowledged automatically when it is received. JMS provides this mode for lazy acknowledgement, but ActiveMatrix BusinessWorks acknowledges messages upon receipt.

Field	Global Var?	Description
		• Transactional – this mode is used when a transaction that can process JMS messages is included in the process definition. The message is acknowledged when the transaction commits. For more information about creating transactions that JMS activities can participate in, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>. • Local Transactional – this mode is used when a transaction involves only JMS messages. For more information about creating transactions that JMS activities can participate in, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Max Sessions	Yes	When the Acknowledge Mode field is set to Client, Transactional and Local Transactional, this field appears for specifying the maximum number of JMS sessions to create for incoming messages. When a JMS queue message is received, the session is blocked until the message is acknowledged. Because the acknowledgement can come at a later time with the Confirm activity, this field allows you to specify a maximum number of new sessions to create to handle incoming messages. After reaching a maximum number of sessions, no new incoming messages can be processed. The total number of active sessions is decreased and another incoming message can be processed after confirming an incoming message.

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Load-Balancing of Incoming Messages

One common application of a JMS queue is to distribute queue messages across multiple receivers, thus balancing the processing of the messages on the queue. To achieve this goal, both the JMS server and ActiveMatrix BusinessWorks must be properly configured. The JMS server must allow the queue messages to be distributed across multiple receivers. For example, in TIBCO Enterprise Message Service, the exclusive property on the queue controls whether messages can be delivered across receivers or not. In ActiveMatrix BusinessWorks, the process definition containing the JMS Queue Receiver must be deployed across multiple process engines. This creates multiple queue receivers for the same queue.

When balancing incoming messages across ActiveMatrix BusinessWorks engines, you should ensure that one engine does not attempt to accept and confirm a large number of incoming messages before other engines can receive the messages. In general, most JMS servers balance the load by distributing messages in a round-robin fashion to all queue receivers. However, there are situations that can cause an uneven distribution of messages across queue receivers. If you set the Acknowledge Mode field to "Auto" on the Configuration tab of the JMS Queue Receiver, the process starter confirms messages as it receives them. When process engines are started at different times, this can lead to one process engine receiving all queue messages and paging them to disk, depending upon how the engine's Max Jobs and Activation Limit properties are set when the engine is deployed.

If you are using TIBCO Enterprise Messaging Service, you can avoid this problem by setting the acknowledge mode to TIBCO EMS Explicit and then use the Flow Limit property in the deployment configuration to control the number of process instances created by the process starter.

If you are not using TIBCO Enterprise Messaging Service, set the Acknowledge Mode field to "Client". In this mode, a process engine can only receive as many messages as it has sessions specified in the Max Sessions field. Once a process engine reaches the maximum number of sessions, another process engine can begin to accept incoming messages. A process engine cannot receive more messages until the messages have been acknowledged by using the [Confirm](#) activity. Once the message is acknowledged, the session is released and the process engine can accept a new message.

Note: If the Sequencing Key field is set to preserve the order of incoming messages, to confirm the messages sequentially you must either set the Acknowledge mode to TIBCO EMS Explicit Client Acknowledge mode or set the Acknowledge mode to Client and Max Sessions to 1. Setting Max Sessions to 1 can limit the system's throughput, so using TIBCO Enterprise Message Service and TIBCO EMS Explicit Client Acknowledge is a better choice.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Message Selector	Yes	A string to determine whether a message should be received. The syntax of the message selector is determined by the JMS provider, but it is usually a subset of SQL92 (where message properties are used instead of table column names). For more information and syntax for a message selector string, see your JMS provider documentation.
JMS	Yes	Any application-specific message properties that is part of

Field	Global Var?	Description
Application Properties		the message. This is specified by the JMS Application Properties shared configuration object. Once specified, these properties appear on the Output tab as "OtherProperties".
Receiver Timeout (seconds)	No	Specifies the polling interval to check for new messages. This default interval is set by the custom engine property <code>bw.plugin.jms.receiverTimeout</code>. If no value is specified for the property, the default polling interval is one second. Set a value in this field to specify the polling interval (in seconds) for this activity. The value in this field overrides the default polling interval.
Precreate Transactions	No	When selected, this field specifies that the transaction is created before a message arrives at the JMS receiver. This field is available when Acknowledge Mode is set to Transactional. Note: The Precreate Transactions feature should be used only with IBM MQ Websphere.
XA Transaction Manager Reference	No	Specifies the XA Transaction Manager Configuration used to create the XA transaction. This field is available when Precreate Transactions field is checked. This configuration overrides the Transaction Manager associated with the group the Event Source is tied to.

Output Editor

The Output Editor tab defines the schema to use for incoming messages whose message type is Map, Stream, or XML Text. Map messages are name/value pairs, and the schema allows you to define the structure of the incoming message. Once defined, the schema on the Output Editor tab becomes the structure used for the body of the message displayed on the Output tab.

For a complete description of using the Output Editor tab, see [Specifying Data Schema](#).

Output

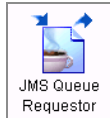
The output for the activity is the following.

Output Item	Datatype	Description
JMSHeaders	complex	The message header fields for the message. For more information about message header fields, see JMS Properties. Only properties applicable to this type of message are displayed.
JMSProperties	complex	The message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration resource.
Body	depends upon the message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements.

Output Item	Datatype	Description
		Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance. Value – Required. Value of the property with one instance. Type – Optional. Type of the property. If not provided, it is considered as string. Following Types are supported: • string • boolean • short • integer • long • float • double • byte Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. The DynamicProperties are also added to the outgoing message.

JMS Queue Requestor

Activity



The **JMS Queue Requestor** activity is used to send a request to a JMS queue name and receive a response back from the JMS client.



Note: This activity uses temporary destinations to ensure that reply messages are received only by the process that sent the request.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JMS Connection	No	The JMS connection. For more information about creating a JMS connection, see JMS Connection .
Destination Queue	Yes	Name of the queue of the outgoing message. The syntax of the queue name is specific to the JMS provider you are using. For more information about queue names, see your JMS provider documentation. Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Message Type	No	The type of the message. This can be one of the following: Simple – A message with no body portion.

Field	Global Var?	Description
		• Bytes – A stream of bytes. • Map – A set of name/value pairs. The names are strings, and the values are simple data types (Java primitives), an array of bytes (use the Binary datatype when mapping this data), or a string. Each item can be accessed sequentially or by its name. • Object – A serializable Java object. • Object Ref – An object reference to a Java object. • Stream – A stream of Java primitives, strings, or arrays of bytes. Each value must be read sequentially. • Text – The message is a <code>java.lang.String</code>. • XML Text – The message is XML text.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Delivery mode	Yes	The delivery mode of the message. Can be one of the following: • PERSISTENT: signifies the messages are stored and forwarded. • NON_PERSISTENT: messages are not stored and may be lost due to failures in transmission. • TIBCO EMS Reliable: this mode is only available when using TIBCO Enterprise Message Service. For more information about this mode, see the TIBCO Enterprise Message Service documentation.

Field	Global Var?	Description
JMSExpiration (secs)	Yes	Corresponds to JMSExpiration property that specifies how long the message can remain active (in seconds). If set to 0, the message does not expire. This field is set in seconds, but the JMSExpiration property is displayed in milliseconds.
Delivery Delay (msecs)	Yes	Delivery Delay feature, which is supported in EMS, is now supported in ActiveMatrix BusinessWorks. It is the minimum length of time in milliseconds that must elapse after a message is sent before the JMS provider may deliver the message to a consumer.
Priority	Yes	Priority of the message. You may set the priority to a value from 0-9. The default value is 4.
Type	Yes	The value to supply to the JMSType header property.
JMS Application Properties	No	Any application-specific message properties that is part of the message. This is specified by the JMS Application Properties shared configuration resource. Once specified, these properties appear on the Output tab as OtherProperties.

Input Editor/Output Editor

The Input Editor/Output Editor tabs define the schema to use for messages of type Map, Stream, or XML Text. Map messages are name/value pairs, and the schema allows you to define the structure of the outgoing request and the incoming reply. The schema defined on the Input Editor tab becomes the body of the message on the Input tab. The schema defined on the Output Editor tab becomes the body of the message on the Output tab.

For a complete description of using the Input/Output Editor tabs, see [Specifying Data Schema](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
destinationQueue	string	The queue to which to send the request. This input item overrides the Destination Queue field on the Configuration tab.
JMSExpiration	integer	Specifies how long the message can remain active (in seconds). If set to 0, the message does not expire.
JMSPriority	string	The priority of the message. This item overrides the priority set on the Advanced tab.
JMSCorrelationID	string	This ID is used to link a response message with its related request message. This is usually the message ID of a request message when this field is found in a reply message, but you can set this element to any value.
JMSType	string	The type of the message. This item overrides the value specified on the Advanced tab.
requestTimeout	integer	The time to wait (in seconds) for the synchronous request-reply operation to complete. An error is returned if the operation does not complete in this time limit.
replyToDestination	string	The destination to use for replies for this activity.

Input Item	Datatype	Description
		Note: If more than one job has the same replyToDestination (for example, you specify a constant in this field), then the correct reply may not be received by each job. Make certain that you specify an expression in this field that assigns a different replyToDestination to each process instance.
JMSProperties	complex	The message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration resource.
Body	depends upon message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements. Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance.

Input Item	Datatype	Description
		Value – Required. Value of the property with one instance. Type – Optional. Type of the property. If not provided, it is considered as string. Following Types are supported: • string • boolean • short • integer • long • float • double • byte Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. Also, DynamicProperties are added to the outgoing message.

Output

The output for the activity is the following.

Output Item	Datatype	Description
JMSHeaders	complex	The message header fields for the message. For more information about message header fields, see JMS Properties .

Output Item	Datatype	Description
		Only properties applicable to this type of message are displayed.
JMSProperties	complex	The message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration resource.
Body	depends upon message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements. Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance. Value – Required. Value of the property with one instance. Type – Optional. Type of the property. If not provided, it is considered as string.

Output Item	Datatype	Description
		Following Types are supported: • string • boolean • short • integer • long • float • double • byte Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. The DynamicProperties are also added to the outgoing message.

Error Output

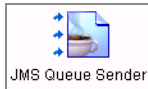
The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When
JMSInvalidInputException	Input to the activity is not valid.
JMSMessageCreateException	The JMS message could not be created.
JMSSessionCreateException	The JMS session could not be created.

Exception	Thrown When
JMSSendException	The JMS send operation failed.
JMSReceiveException	The JMS receive operation failed.
JMSActivityException	This activity fails.
ActivityTimeoutException	This activity has reached the specified time out and the response has not been received.

JMS Queue Sender

Activity



The **JMS Queue Sender** activity sends a message to the specified JMS queue.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JMS Connection	No	The JMS connection. For more information about creating a JMS connection, see JMS Connection .
Destination Queue	Yes	Name of the queue of the outgoing message. The syntax of the queue name is specific to the JMS provider you are using. For more information about queue names, see

Field	Global Var?	Description
		your JMS provider documentation . Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Message Type	No	The type of the message. This can be one of the following: • Simple – A message with no body portion. • Bytes – A stream of bytes. • Map – A set of name/value pairs. The names are strings, and the values are simple data types (Java primitives), an array of bytes (use the Binary datatype when mapping this data), or a string. Each item can be accessed sequentially or by its name. • Object – A serializable Java object. • Object Ref – An object reference to a Java object. • Stream – A stream of Java primitives, strings, or arrays of bytes. Each value must be read sequentially. • Text – The message is a <code>java.lang.String</code> • XML Text – The message is XML text.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Reply-to Queue	Yes	The name of the queue where replies should be sent. This field is optional, and replies are sent to the queue specified on the Configuration tab by default. Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Delivery mode	Yes	The delivery mode of the message. It can be one of the following: • Persistent: signifies the messages are stored and forwarded. • Non-Persistent: messages are not stored and may be lost due to failures in transmission. • TIBCO EMS Reliable: is mode is only available when using TIBCO Enterprise Message Service. For more information about this mode, see the TIBCO Enterprise Message Service documentation .
JMSExpiration (secs)	Yes	Corresponds to JMSExpiration property that specifies how long the message can remain active (in seconds). If set to 0, the message does not expire. This field is set in seconds, but the JMSExpiration property is displayed in milliseconds.
DeliveryDelay (msec)	Yes	Delivery Delay feature, which is supported in EMS, is now supported in ActiveMatrix BusinessWorks. It is the minimum length of time in milliseconds that must elapse after a message is sent before the JMS provider may deliver the message to a consumer.

Field	Global Var?	Description
Priority	Yes	Priority of the message. You may set the priority to a value from 0-9. The default value is 4.
Type	Yes	The value to supply to the JMSType header property.
JMS Application Properties	No	Any application-specific message properties that is part of the message. This is specified by the JMS Application Properties shared configuration resource. Once specified, these properties appear on the Output tab as OtherProperties.
Override Transaction Behavior	Yes	Overrides the default behavior of a transaction group. If this activity is in a transaction group, the JMS message is normally committed or rolled back with the other participants in the transaction. If this check box is selected, the JMS message does not participate in the transaction.

Input Editor

The Input Editor tab defines the schema to use for outgoing messages whose message type is Map, Stream, or XML Text. Map messages are name/value pairs, and the schema allows you to define the structure of the outgoing message. Once defined, the schema on the Input Editor tab becomes the structure used for the body of the message displayed on the Input tab.

For a complete description of using the Input Editor tab, see [Specifying Data Schema](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
destinationQueue	string	The queue to which to send the request. This input item overrides the Destination Queue field on the Configuration tab.
replyToQueue	string	The queue where replies should be sent for this request. This input item overrides the reply-to destination specified on the Advanced tab.
JMSExpiration	integer	Specifies how long the message can remain active (in seconds). If set to 0, the message does not expire.
JMSPriority	string	The priority of the message. This item overrides the priority set on the Advanced tab.
JMSDeliveryMode	string	The delivery mode of the message. Can be one of the following: • Persistent: signifies the messages are stored and forwarded. • Non-Persistent: messages are not stored and may be lost due to failures in transmission. • When using TIBCO Enterprise Message Service, the JMSDeliveryMode header can be TIBCO EMS Reliable. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
JMSCorrelationID	string	This ID is used to link a response message with its related request message. This is usually the message ID of a request message when this field exists in a reply message.
JMSType	string	The type of the message. This item overrides the value specified on the Advanced tab.

Input Item	Datatype	Description
JMSProperties	complex	The message properties. For more information about message properties, see JMS Properties . Only properties applicable to this type of message are displayed.
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration resource.
Body	depends upon message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements. Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance. Value – Required. Value of the property with one instance. Type – Optional. Type of the property. If not provided, it is considered as string. Following Types are supported: • string • boolean

Input Item	Datatype	Description
		• short • integer • long • float • double • byte Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. Also, DynamicProperties are added to the outgoing message.

Output

The output for the activity is the following.

Output Item	Datatype	Description
MessageID	string	The unique identifier of the message.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
JMSInvalidInputException	Input to the activity is not valid.
JMSMessageCreateException	The JMS message could not be created.
JMSSessionCreateException	The JMS session could not be created.
JMSSendException	The JMS send operation failed.
JMSActivityException	This activity fails.

JMS Topic Publisher

Activity



The JMS Topic Publisher sends a message to the specified JMS topic.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JMS Connection	No	The JMS connection. For more information about creating a JMS connection, see JMS Connection .
Destination Topic	Yes	Name of the topic of the outgoing message. The syntax of the topic name is specific to the JMS provider you are using. For more information about topic names, see

Field	Global Var?	Description
		your JMS provider documentation . Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Message Type	No	The type of the message. This can be one of the following: • Simple – A message with no body portion. • Bytes – A stream of bytes. • Map – A set of name/value pairs. The names are strings, and the values are simple data types (Java primitives), an array of bytes (use the Binary datatype to map this data), or a string. Each item can be accessed sequentially or by its name. • Object – A serializable Java object. • Object Ref – An object reference to a Java object. • Stream – A stream of Java primitives, strings, or arrays of bytes. Each value must be read sequentially. • Text – The message is a java.lang.String. • XML Text – The message is XML text.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Reply-to Topic	Yes	The name of the topic to which replies should be sent. This field is optional, and replies are sent to the topic specified on the Configuration tab by default. Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Delivery Mode	Yes	The delivery mode of the message. Can be one of the following: • Persistent : signifies the messages are stored and forwarded. • Non-Persistent: messages are not stored and may be lost due to failures in transmission. • TIBCO EMS Reliable: this mode is only available when using TIBCO Enterprise Message Service. For more information about this mode, see the TIBCO Enterprise Message Service documentation .
JMSExpiration (secs)	Yes	Corresponds to JMSExpiration property that specifies how long the message can remain active (in seconds). If set to 0, the message does not expire. This field is set in seconds, but the JMSExpiration property is displayed in milliseconds.
Delivery Delay (msecs)	Yes	Delivery Delay feature, which is supported in EMS, is now supported in ActiveMatrix BusinessWorks. It is the minimum length of time in milliseconds that must elapse after a message is sent before the JMS provider may deliver the message to a consumer.

Field	Global Var?	Description
Priority	Yes	Priority of the message. You may set the priority to a value from 0-9. The default value is 4.
Type	Yes	The value to supply to the JMSType header property.
JMS Application Properties	No	Any application-specific message properties to include in the message. This is specified by the JMS Application Properties shared configuration resource. Once specified, these properties appear on the Output tab as OtherProperties.
Override Transaction Behavior	Yes	Overrides the default behavior of a transaction group. If this activity is in a transaction group, the JMS message is normally committed or rolled back with the other participants in the transaction. If this check box is selected, the JMS message does not participate in the transaction.

Input Editor

The Input Editor tab defines the schema to use for outgoing messages whose message type is Map, Stream, or XML Text. Map messages are name/value pairs, and the schema allows you to define the structure of the outgoing message. Once defined, the schema on the Input Editor tab becomes the structure used for the body of the message displayed on the Input tab.

For a complete description of using the Input Editor tab, see [Specifying Data Schema](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
destinationTopic	string	The topic to which to send the request. This input item overrides the Destination Topic field on the Configuration tab.
replyToTopic	string	The topic to which replies should be sent for this request. This input item overrides the Reply-to destination on the Advanced tab.
JMSExpiration	integer	Specifies how long the message can remain active (in seconds). If set to 0, the message does not expire.
JMSPriority	string	The priority of the message. This item overrides the priority set on the Advanced tab.
JMSDeliveryMode	string	The delivery mode of the message. Can be one of the following: • Persistent: signifies the messages are stored and forwarded. • Non-Persistent: messages are not stored and may be lost due to failures in transmission. • When using TIBCO Enterprise Message Service, the JMSDeliveryMode header can be TIBCO EMS Reliable. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
JMSCorrelationID	string	This ID is used to link a response message with its related request message. This is usually the message ID of a request message when this field is found in a reply message.
JMSType	string	The type of the message. This item overrides the value specified on the Advanced tab.

Input Item	Datatype	Description
JMSProperties	complex	The message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration resource.
Body	depends upon message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements. Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance. Value – Required. Value of the property with one instance. Type – Optional. Type of the property. If not provided, it is considered as string. Following Types are supported: • string

Input Item	Datatype	Description
		• boolean • short • integer • long • float • double • byte Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. Also, DynamicProperties are added to the outgoing message.

Output

The output for the activity is the following.

Output Item	Datatype	Description
MessageID	string	The unique identifier of the message.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
JMSInvalidInputException	Input to the activity is not valid.
JMSMessageCreateException	The JMS message could not be created.
JMSSessionCreateException	The JMS session could not be created.
JMSSendException	The JMS send operation failed.

JMS Topic Requestor

Activity



The **JMS Topic Requestor** activity is used to communicate with a JMS application's request-response service. This service invokes an operation with input and output. The request is sent to a JMS topic and the JMS application returns the response to the request.



Note: This activity uses temporary destinations to ensure that reply messages are received only by the process that sent the request.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JMS	No	The JMS connection. For more information about creating a

Field	Global Var?	Description
Connection		JMS connection, see JMS Connection .
Destination Topic	Yes	Name of the topic of the outgoing message. The syntax of the topic name is specific to the JMS provider you are using. For more information about topic names, see your JMS provider documentation. Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Message Type	No	The type of the message. This can be one of the following: • Simple – A message with no body portion. • Bytes – A stream of bytes. • Map – A set of name/value pairs. The names are strings, and the values are simple data types (Java primitives), an array of bytes (use the Binary datatype to map this data), or a string. Each item can be accessed sequentially or by its name. • Object – A serializable Java object. • Object Ref – An object reference to a Java object. • Stream – A stream of Java primitives, strings, or arrays of bytes. Each value must be read sequentially. • Text – The message is a <code>java.lang.String</code>. • XML Text – The message is XML text.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Delivery Mode	Yes	The delivery mode of the message. Can be one of the following: • Persistent: signifies the messages are stored and forwarded. • Non-Persistent: messages are not stored and may be lost due to failures in transmission. • When using TIBCO Enterprise Message Service, the JMSDeliveryMode header can be TIBCO EMS Reliable. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
JMSExpiration (secs)	Yes	Corresponds to JMSExpiration property that specifies how long the message can remain active (in seconds). If set to 0, the message does not expire. This field is set in seconds, but the JMSExpiration property is displayed in milliseconds.
Delivery Delay (msecs)	Yes	Delivery Delay feature, which is supported in EMS, is now supported in ActiveMatrix BusinessWorks. It is the minimum length of time in milliseconds that must elapse after a message is sent before the JMS provider may deliver the message to a consumer.
Priority	Yes	Priority of the message. You may set the priority to a value from 0-9. The default value is 4.
Type	Yes	The value to supply to the JMSType header property.
JMS Application Properties	No	Any application-specific message properties to include in the message. This is specified by the JMS Application Properties shared configuration resource.

Field	Global Var?	Description
Once specified, these properties appear on the Output tab as OtherProperties.		

Input Editor/Output Editor

The Input Editor/Output Editor tabs define the schema to use for messages of type Map, Stream, or XML Text. Map messages are name/value pairs, and the schema allows you to define the structure of the outgoing request and the incoming reply. The schema defined on the Input Editor tab becomes the body of the message on the Input tab. The schema defined on the Output Editor tab becomes the body of the message on the Output tab.

For a complete description of using the Input/Output Editor tab, see [Specifying Data Schema](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
destinationTopic	string	The topic to which to send the request. This input item overrides the Destination Topic field on the Configuration tab.
JMSExpiration	integer	Specifies how long the message can remain active (in seconds). If set to 0, the message does not expire.
JMSPriority	string	The priority of the message. This item overrides the priority set on the Advanced tab.

Input Item	Datatype	Description
JMSDeliveryMode	string	The delivery mode of the message. Can be one of the following: • Persistent: signifies the messages are stored and forwarded. • Non-Persistent: messages are not stored and may be lost due to failures in transmission. • When using TIBCO Enterprise Message Service, the JMSDeliveryMode header can be TIBCO EMS Reliable. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
JMSCorrelationID	string	This ID is used to link a response message with its related request message. This is usually the message ID of a request message when this field exists in a reply message, but you can set this element to any value.
JMSType	string	The type of the message. This item overrides the value specified on the Advanced tab.
requestTimeout	integer	The time to wait (in seconds) for the synchronous request-reply operation to complete. An error is returned if the operation does not complete in this time limit.
replyToTopic	string	The destination to use for replies for this activity. Note: If more than one job has the same replyToDestination (for example, you specify a constant in this field), then the correct reply may not be received by each job. Make certain that you specify an expression in this field that assigns a different replyToDestination to each process instance.

Input Item	Datatype	Description
JMSProperties	complex	The message properties. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration resource.
Body	depends upon message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements. Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance. Value – Required. Value of the property with one instance. Type – Optional. Type of the property. If not provided, it is considered as string. Following Types are supported: • string • boolean

Input Item	Datatype	Description
		• short • integer • long • float • double • byte Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. Also, DynamicProperties are added to the outgoing message.

Output

The output for the activity is the following.

Output Item	Datatype	Description
JMSHeaders	complex	The message header fields for the message. For more information about message header fields, see JMS Properties. Only properties applicable to this type of message are displayed.
JMSProperties	complex	The message properties for the message. For more information about message properties, see JMS Properties. Only properties applicable to this type of message are displayed.

Output Item	Datatype	Description
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration resource.
Body	depends upon message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements. Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance. Value – Required. Value of the property with one instance. Type – Optional. Type of the property. If not provided, it is considered as string. Following Types are supported: • string • boolean • short • integer • long

Output Item	Datatype	Description
		• float • double • byte Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. The DynamicProperties are also added to the outgoing message.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
JMSInvalidInputException	Input to the activity is not valid.
JMSMessageCreateException	The JMS message could not be created.
JMSSessionCreateException	The JMS session could not be created.
JMSSendException	The JMS send operation failed.
JMSReceiveException	The JMS receive operation failed.
ActivityTimedOutException	This activity has reached the specified time out and the response has not been received.

JMS Topic Subscriber

Process Starter



Starts a process based on the receipt of a message for the specified JMS topic.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JMS Connection	No	The JMS connection. For more information about creating a JMS connection, see JMS Connection .
Destination Topic	Yes	Name of the topic of the incoming message. The syntax of the topic name is specific to the JMS provider you are using. For more information about topic names, see your JMS provider documentation. Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Message Type		The type of the message. This can be one of the following: - Simple – A message with no body portion. - Bytes – A stream of bytes.

Field	Global Var?	Description
		• Map – A set of name/value pairs. The names are strings, and the values are simple data types (Java primitives), an array of bytes (use the Binary datatype to map this data), or a string. Each item can be accessed sequentially or by its name. • Object – A serializable Java object. • Object Ref – An object reference to a Java object. • Stream – A stream of Java primitives, strings, or arrays of bytes. Each value must be read sequentially. • Text – The message is a <code>java.lang.String</code>. • XML Text – The message is XML text.
Durable Subscription	Yes	Specifies a durable subscription. You must specify a name in the Subscription name field that is registered with the JMS application as the durable subscription name.
Subscription Name	Yes	The subscription name registered with the JMS application for durable subscriptions. This field is only available when the Durable Subscription field is checked. Once a durable subscription is created, it can only be removed by accessing the administration tool of the JMS provider. For more information about managing durable subscriptions, see the documentation of your JMS provider.
Suppress Local Messages	Yes	Specifies to not receive messages on the specified topic name when the message origin is the JMS application on the same connection as the process engine. If your process definition publishes and subscribes to messages with the same topic name, this option is useful if you want to specify whether to receive messages sent by the same JMS application that published the message.

Field	Global Var?	Description
		Checking this field prevents the process from receiving messages sent by the same connection. Unchecking this option specifies messages sent by the same connection should be received.
Acknowledge Mode	Yes	This field is available when a JMS transport is selected. This field specifies the acknowledgement mode to be used for incoming messages. It can be one of the following: • Auto : the message is automatically acknowledged when it is received. • Client: the message is acknowledged when the process implementing the operation ends successfully. • Dups OK : the message is acknowledged automatically when it is received. JMS provides this mode for lazy acknowledgement, but ActiveMatrix BusinessWorks acknowledges messages upon receipt. • TIBCO EMS Explicit : (only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, are redelivered instead of all messages in the session. The session is not blocked and one session handles all incoming messages for each process instance. Note: In the ActiveMatrix BusinessWorks 5.10 release, a change occurs in the behavior of this feature. Now only the messages that are not confirmed, are redelivered. • TIBCO EMS Explicit Dups OK : (only available for TIBCO Enterprise Message Service) a message

Field	Global Var?	Description
		which is not acknowledged using the Confirm activity before the process instance ends, are redelivered instead of all messages in the session. The session is not blocked and one session handles all the incoming messages for each process instance. The messages however, would be lazily acknowledged.
Acknowledge Mode	Yes	• Dups OK – the message is acknowledged automatically when it is received. JMS provides this mode for lazy acknowledgement, but ActiveMatrix BusinessWorks acknowledges messages upon receipt. • Transactional – this mode is used when a transaction that can process JMS messages is included in the process definition. The message is acknowledged when the transaction commits. For more information about creating transactions that JMS activities can participate in, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>. • Local Transactional – this mode is used when a transaction involves only JMS messages. For more information about creating transactions that JMS activities can participate in, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Max Sessions (Read Only)	No	When the Acknowledge Mode field is set to "Client", "Transactional" or "Local Transactional", this read-only field appears to notify you that only one session can receive incoming topic messages until the message is confirmed. When a JMS topic message is received, the session is blocked until the message is acknowledged. Because all sessions subscribed to JMS topics receive the same messages, only one session can actively listen for new messages until each message is confirmed.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Message Selector	Yes	A string to determine whether a message should be received. The syntax of the message selector is determined by the JMS provider, but it is usually a subset of SQL92 (where message properties are used instead of table column names). For more information and syntax for a message selector string, see your JMS provider documentation.
JMS Application Properties	No	Any application-specific message properties to include in the message. This is specified by the JMS Application Properties shared configuration resource. Once specified, these properties appear on the Output tab as OtherProperties.
Receiver Timeout (seconds)	No	Specifies the polling interval to check for new messages. This default interval is set by the custom engine property <code>bw.plugin.jms.receiverTimeout</code>. If no value is specified for the property, the default polling interval is one second. Set a value in this field to specify the polling interval (in seconds) for this activity. The value in this field overrides the default polling interval.
Precreate Transactions	No	When checked, this field specifies that the transaction is created before a message arrives at the JMS receiver. Note: The Precreate Transactions feature should be used only with IBM MQ Websphere.
XA Transaction Manager Reference	No	Specifies the XA Transaction Manager Configuration used to create the XA transaction. This field is available when Precreate Transactions field is checked.

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value are executed sequentially in the order the process instances were created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output Editor

The Output Editor tab defines the schema to use for incoming messages whose message type is Map, Stream, or XML Text. Map messages are name/value pairs, and the schema allows you to define the structure of the incoming message. Once defined, the schema on the Output Editor tab becomes the structure used for the body of the message displayed on the Output tab.

For a complete description of using the Output Editor tab, see [Specifying Data Schema](#).

Output

The output for the activity is the following.

Output Item	Datatype	Description
JMSHeaders	complex	The message header fields for the message. For more information about message header fields, see JMS Properties .
JMSProperties	complex	The message properties for the message. For more information about message properties, see JMS Properties .
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration resource.
Body	depends upon the message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements. Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance. Value – Required. Value of the property with one instance. Type – Optional. Type of the property. If not provided, it is considered as string. Following Types are supported:

Output Item	Datatype	Description
		• string • boolean • short • integer • long • float • double • byte
		Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. The DynamicProperties are also added to the outgoing message.

Reply To JMS Message

Activity



The **Reply To JMS Message** activity sends a reply to a previously received JMS queue or topic message. Activities that can receive JMS messages are listed on the Configuration tab. The activity that you select determines which message the reply message is in response to.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Reply To	No	A drop-down list of activities that can receive JMS queue or topic messages. The activity you select determines the message this activity replies to. The reply-to queue or topic name from the message in the selected activity is used to send the reply message.
Message Type	No	The type of the message. This can be one of the following: • Simple – A message with no body portion. • Bytes – A stream of bytes. • Map – A set of name/value pairs. The names are strings, and the values are simple data types (Java primitives), an array of bytes (use the Binary datatype to map this data), or a string. Each item can be accessed sequentially or by its name. • Object – A serializable Java object. • Object Ref – An object reference to a Java object. • Stream – A stream of Java primitives, strings, or arrays of bytes. Each value must be read sequentially. • Text – The message is a <code>java.lang.String</code>. • XML Text – The message is XML text.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Delivery Mode	Yes	The delivery mode of the message. It can be one of the following: • Persistent: signifies the messages are stored and forwarded. • Non-Persistent messages are not stored and may be lost due to failures in transmission. • TIBCO EMS Reliable: this mode is only available when using TIBCO Enterprise Message Service. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
JMSExpiration (secs)	Yes	Corresponds to JMSExpiration property that specifies how long the message can remain active (in seconds). If set to 0, the message does not expire. This field is set in seconds, but the JMSExpiration property is displayed in milliseconds.
Delivery Delay (msecs)	Yes	Delivery Delay feature, which is supported in EMS, is now supported in TIBCO ActiveMatrix BusinessWorks. It is the minimum length of time in milliseconds that must elapse after a message is sent before the JMS provider may deliver the message to a consumer.
Priority	Yes	Priority of the message. You may set the priority to a value from 0-9. The default value is 4.
Type	Yes	The value to supply to the JMSType header property.
JMS Application Properties	No	Any application-specific message properties that is part of the message. This is specified by the JMS Application Properties shared configuration resource. Once specified, these properties appear on the Output tab as "OtherProperties".

Input Editor

The Input Editor tab defines the schema to use for outgoing replies where the message type is Map, Stream, or XML Text. Map messages are name/value pairs, and the schema allows you to define the structure of the outgoing message. Once defined, the schema on the Input Editor tab becomes the structure used for the body of the message displayed on the Input tab.

For a complete description of using the Input Editor tab, see [Specifying Data Schema](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Delivery mode	Long	The delivery mode of the message. Can be one of the following: • Persistent : signifies the messages are stored and forwarded. • Non-Persistent : messages are not stored and may be lost due to failures in transmission. • TIBCO EMS Reliable: this mode is only available when using TIBCO Enterprise Message Service. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
JMSExpiration (secs)	String	Corresponds to JMSExpiration property that specifies how long the message can remain active (in seconds). If set to 0, the message does not expire. This field is set in seconds, but the JMSExpiration

Input Item	Datatype	Description
		property is displayed in milliseconds.
JMSPriority	string	Priority of the message. You may set the priority to a value from 0-9. This item overrides the value specified on the Configuration tab.
JMSCorrelationID	string	This ID is used to link a response message with its related request message. This property is usually set to the message ID of the message you are replying to, but any value can be used. For example, you may use another field in the body of the message (such as orderID) to correlate request and reply messages. The JMSCorrelationID of the reply message is set in this input element. If this element is not set, the correlation ID is set as follows: • If the JMSCorrelationID input element is not set, the value of the JMSCorrelationID property in the message you are replying to is used. • If neither this input element nor the JMSCorrelationID of the message you are replying to are set, the message ID of the message you are replying to is used. • If none of the above values are set, the JMSCorrelationID of the reply message is set to null.
JMSType	string	The type of the message. This item overrides the value specified on the Advanced tab.
JMSProperties	complex	The message properties for the message. For more information about message properties, see JMS Properties .

Input Item	Datatype	Description
		Only properties applicable to this type of message are displayed.
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration object.
Body	depends upon message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements. Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance. Value – Required. Value of the property with one instance. Type – Optional. Type of the property. If not provided, it is considered as string. Following Types are supported: • string • boolean • shortx

Input Item	Datatype	Description
		• integer • long • float • double • byte Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. Also, DynamicProperties are added to the outgoing message.

Output

The activity has no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
JMSInvalidInputException	Input to the activity is not valid.
JMSMessageCreateException	The JMS message could not be created.
JMSSessionCreateException	The JMS session could not be created.
JMSSendException	The JMS send operation failed.

Wait for JMS Queue Message

Activity



The Wait for JMS Queue Message waits for the receipt of a message for the specified JMS queue.

For information on running the "Wait for" activities in a multi-engine mode, see [Scalability With Incoming Events](#) in *TIBCO ActiveMatrix*

BusinessWorks™ Process Design. For more details on using the activity in a multi-engine design, see [Using Wait for JMS Queue Message Activity in Multi-Engine Design](#).

Configuration

The Configuration tab has the following fields

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JMS Connection	No	The JMS connection. For more information about creating a JMS connection, see JMS Connection .
Destination Queue	Yes	Name of the queue of the incoming message. The syntax of the queue name is specific to the JMS provider you are using. For more information about queue names, see your JMS provider documentation. Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.

Field	Global Var?	Description
Message Type		The type of the message. This can be one of the following: • Simple – A message with no body portion. • Bytes – A stream of bytes. • Map – A set of name/value pairs. The names are strings, and the values are simple data types (Java primitives), an array of bytes (use the Binary datatype to map this data), or a string. Each item can be accessed sequentially or by its name. • Object – A serializable Java object. • Object Ref – An object reference to a Java object. • Stream – A stream of Java primitives, strings, or arrays of bytes. Each value must be read sequentially. • Text – The message is a <code>java.lang.String</code>. • XML Text – The message is XML text.
Acknowledge Mode	Yes	The acknowledge mode for the incoming messages. It can be one of the following: • Auto : the message is automatically acknowledged when it is received. • Client : the message gets acknowledged at a later point by using the Confirm activity. If the message is not confirmed before the process instance ends, the message is redelivered. Ensure that your process definition confirms the message when using this acknowledge mode. • TIBCO EMS Explicit :(only available for TIBCO

Field	Global Var?	Description
		Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, is redelivered instead of all messages in the session. The session is not blocked and one session handles all incoming messages for each process instance. Note: In the ActiveMatrix BusinessWorks 5.10 release, there is a change in the behavior of this feature. Now only the messages that are not confirmed, would be redelivered. • TIBCO EMS Explicit Dups OK :(only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, is redelivered instead of all messages in the session. The session is not blocked and one session handles all the incoming messages for each process instance. The messages however, would be lazily acknowledged. • TIBCO EMS No Acknowledge: the message is not acknowledged when it is received. • Dups OK : the message is acknowledged automatically when it is received. JMS provides this mode for lazy acknowledgement, but ActiveMatrix BusinessWorks acknowledges messages upon receipt. • Transactional : this mode is used when a transaction that can process JMS messages is included in the process definition. The message is acknowledged when the transaction commits. For more information about creating transactions that JMS activities can participate in, see <i>TIBCO</i>

Field	Global Var?	Description
		<i>ActiveMatrix BusinessWorks™ Process Design.</i>

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Message Selector	Yes	A string to determine whether a message should be received. The syntax of the message selector is determined by the JMS provider, but it is usually a subset of SQL92 (where message properties are used instead of table column names). For more information and syntax for a message selector string, see your JMS provider documentation .
JMS Application Properties	No	Any application-specific message properties that is part of the message. This is specified by the JMS Application Properties shared configuration resource. Once specified, these properties appear on the Output tab as "OtherProperties".
Receiver Timeout (seconds)	No	Specifies the polling interval to check for new messages. This default interval is set by the custom engine property <code>bw.plugin.jms.receiverTimeout</code>. If no value is specified for the property, the default polling interval is one second. Set a value in this field to specify the polling interval (in seconds) for this activity. The value in this field overrides the default polling interval.

Message Event

The Event tab has the following fields.

Field	Description
Candidate Event Key	Expression built from the data of the incoming file change event. This expression must evaluate to a string and it is compared to the "key" field of the activity's input. If the Candidate Event Key and the activity key match, then the process accepts the incoming message. For example, you may have JMS Queue Sender activity earlier in the process diagram and you are expecting a message in response to the sent message. Place the JMSCorrelationID from the header of the incoming message into the Candidate Event Key field, then place the JMSMessageID from the header of the sent message in the "key" field on the input tab. This expression is specified in XPath, and only data from the incoming event is available for use in this XPath expression. For more information about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Event Timeout (msec)	A message may arrive before this activity is executed. This field specifies the amount of time (in milliseconds) a message waits if it is received before this activity is executed. If the event timeout expires, an error is logged and the event is discarded. If no value is specified in this field, the message waits indefinitely. If zero is specified, the event is discarded immediately, unless this activity has already been executed.

i Note: When building an expression in the Candidate Event Key field, only data from the incoming event is available. This is because you want to place an expression containing incoming event data in the Candidate Event Key field. When the results of this expression match the results of the expression in the "key" item on the Input tab, the **Wait For JMS Queue Message** activity proceeds.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
key	string	The value that is compared to the Candidate Event Key field of the Event tab to determine if this message applies to this process. For an example of using the Candidate Event Key field with the key input item, see Message Event .
processTimeout	integer	The time (in milliseconds) to wait for the incoming message. An error is returned if the message is not received in this time limit.

Output Editor

The Output Editor tab defines the schema to use for incoming messages whose message type is Map, Stream, or XML Text. Map messages are name/value pairs, and the schema allows you to define the structure of the incoming message. Once defined, the schema on the Output Editor tab becomes the structure used for the body of the message displayed on the Output tab.

For a complete description of using the Output Editor tab, see [Specifying Data Schema](#).

Output

The output for the activity is the following.

Output Item	Datatype	Description
JMSHeaders	complex	The message header fields for the message. For more information about message header fields, see JMS Properties .
JMSProperties	complex	The message properties for the message. For more information about message properties, see JMS

Output Item	Datatype	Description
		Properties.
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration resource.
Body	depends upon the message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements. Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance. Value – Required. Value of the property with one instance. Type – Optional. Type of the property. If not provided, it is considered as string. Following Types are supported: • string • boolean • short • integer

Output Item	Datatype	Description
		• long • float • double • byte
		Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. The DynamicProperties are also added to the outgoing message.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ActivityTimeoutException	The activity has reached the event or process timeout.

Wait for JMS Topic Message

Activity



The Wait for JMS Topic Message waits for the receipt of a message for the specified JMS topic.

For information on running the "Wait for" activities in a multi-engine mode, see [Scalability With Incoming Events](#) in *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
JMS Connection	No	The JMS connection. For more information about creating a JMS connection, see JMS Connection .
Destination Topic	Yes	Name of the topic of the outgoing message. The syntax of the topic name is specific to the JMS provider you are using. for more information about topic names, see your JMS provider documentation. Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Message Type	No	The type of the message. This can be one of the following: • Simple – A message with no body portion. • Bytes – A stream of bytes. • Map – A set of name/value pairs. The names are strings, and the values are simple data types (Java primitives), an array of bytes (use the Binary datatype to map this data), or a string. Each item can be accessed sequentially or by its name. • Object – A serializable Java object. • Object Ref – An object reference to a Java object.

Field	Global Var?	Description
		• Stream – A stream of Java primitives, strings, or arrays of bytes. Each value must be read sequentially. • Text – The message is a <code>java.lang.String</code>. • XML Text – The message is XML text.
Durable Subscription	Yes	Specifies a durable subscription. You must specify a name in the Subscription name field that is registered with the JMS application as the durable subscription name.
Subscription Name	Yes	The subscription name registered with the JMS application for durable subscriptions. This field is only available when the Durable subscription field is checked. Once a durable subscription is created, it can only be removed by accessing the administration tool of the JMS provider. See the documentation of your JMS provider for more information about managing durable subscriptions.
Suppress Local Messages	No	Specifies to not receive messages on the specified topic name when the message origin is the JMS application on the same connection as the process engine. If your process definition publishes and subscribes to messages with the same topic name, this option is useful if you want to specify whether to receive messages sent by the same JMS application that published the message. Checking this field prevents the process from receiving messages sent by the same connection. Unchecking this option specifies messages sent by the same connection should be received.
Acknowledge Mode	Yes	The acknowledge mode for the incoming messages. It can be one of the following:

Field	Global Var?	Description
		• Auto : the message is automatically acknowledged when it is received. • TIBCO EMS Explicit : (only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, can be redelivered instead of all messages in the session. The session is not blocked and one session handles all incoming messages for each process instance. Note: In the ActiveMatrix BusinessWorks 5.10 release, there is a change in the behavior of this feature. Now only the messages that are not confirmed, would be redelivered. • TIBCO EMS Explicit Dups OK :(only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, can be redelivered instead of all messages in the session. The session is not blocked and one session handles all the incoming messages for each process instance. The messages however, would be lazily acknowledged. • Dups OK :the message is acknowledged automatically when it is received. JMS provides this mode for lazy acknowledgement, but ActiveMatrix BusinessWorks acknowledges messages upon receipt.

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Message Selector	Yes	A string to determine whether a message should be received. The syntax of the message selector is determined by the JMS provider, but it is usually a subset of SQL92 (where message properties are used instead of table column names). For more information and syntax for a message selector string, see your JMS provider documentation.
JMS Application Properties	No	Any application-specific message properties to include in the message. This is specified by the JMS Application Properties shared configuration resource. Once specified, these properties appear on the Output tab as OtherProperties.
Receiver Timeout (seconds)	No	Specifies the polling interval to check for new messages. This default interval is set by the custom engine property <code>bw.plugin.jms.receiverTimeout</code>. If no value is specified for the property, the default polling interval is one second. Set a value in this field to specify the polling interval (in seconds) for this activity. The value in this field overrides the default polling interval.

Message Event

The Event tab has the following fields.

Field	Description
Candidate Event Key	Expression built from the data of the incoming file change event. This expression should evaluate to a string and it is compared to the "key" field of the activity's input. If the Candidate Event Key and the activity's key match, then the process accepts the incoming message. For example, you may have JMS Topic Publisher activity earlier in the

Field	Description
	process diagram and you are expecting a message in response to the sent message. You would place the JMSCorrelationID from the header of the incoming message into the Candidate Event Key field. You would then place the JMSMessageID from the header of the outgoing message earlier in the process in the "key" field on the input tab. This expression is specified in XPath, and only data from the incoming event is available for use in this XPath expression. For more information about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Event Timeout (msec)	A message may arrive before this activity is executed. This field specifies the amount of time (in milliseconds) a message waits if it is received before this activity is executed. If the event timeout expires, an error is logged and the event is discarded. If no value is specified in this field, the message waits indefinitely. If zero is specified, the event is discarded immediately, unless this activity has already been reached.

Note: When building an expression in the Candidate Event Key field, only data from the incoming event is available. This is because you want to place an expression containing incoming event data in the Candidate Event Key field. When the results of this expression match the results of the expression in the "key" item on the Input tab, the **Wait For JMS Topic Message** activity proceeds.

Using Wait for JMS Queue Message Activity in Multi-Engine Design

All "Wait for" activities start listening from the time the bwengine loads and can be configured to filter static data. These activities cannot filter dynamic data.

The activity pulls all the messages that are in the queue upon startup and processes them according to the Event key. Any messages that were picked up but do not match the key are discarded. A Wait for JMS Queue Message Activity cannot be used with the Candidate Event key as the key is applied after receiving the message from the queue.

In a multi-engine environment, when new messages arrive on the queue, the messages are dispatched among all active listeners in a round-robin fashion. Thus an engine may

potentially pick up the message which was intended for some other engine, and later discard the message because of a mismatch in the Event key.

It is recommended that you use the Wait for JMS Queue Message activity with the Message Selector in a multi-engine mode. The Message Selector is applied while browsing the queue before picking up the messages.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
key	string	The value that is compared to the Candidate Event Key field of the Event tab to determine if this message applies to this process. For an example of using the Candidate Event Key field with the key input item, see Message Event .
processTimeout	integer	The time (in milliseconds) to wait for the incoming message. An error is returned if the message is not received in this time limit.

Output Editor

The Output Editor tab defines the schema to use for incoming messages whose message type is Map, Stream, or XML Text. Map messages are name/value pairs, and the schema allows you to define the structure of the incoming message. Once defined, the schema on the Output Editor tab becomes the structure used for the body of the message displayed on the Output tab.

For a complete description of using the Output Editor tab, see [Specifying Data Schema](#).

Output

The output for the activity is the following.

Output Item	Datatype	Description
JMSHeaders	complex	The message header fields for the message. For more information about message header fields, see JMS Properties .
JMSProperties	complex	The message properties for the message. For more information about message properties, see JMS Properties .
OtherProperties	complex	If the JMS Application Properties field is specified on the Advanced tab, the OtherProperties group lists the application-specific properties in the selected JMS Application Properties shared configuration resource.
Body	depends upon the message type	The body of the message.
DynamicProperties	complex	The dynamic properties is an additional parameter to add runtime property to the Outgoing JMS Messages which is specified in the Input tab. This is an optional element with only one instance. Dynamic property can have multiple property elements. Each property element denotes a single dynamic property and can contain the following elements: Name – Required. Name of the property with one instance. Value – Required. Value of the property with one instance. Type – Optional. Type of the property. If not

Output Item	Datatype	Description
		provided, it is considered as string. Following Types are supported: • string • boolean • short • integer • long • float • double • byte Note: The DynamicProperty overwrites the value of a property (with the same name) added using the JMS Application Property. The DynamicProperties are also added to the outgoing message.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ActivityTimedOutException	The activity has reached the event or process timeout.

Reconnecting to JMS Servers

ActiveMatrix BusinessWorks supports some exceptions, where the ActiveMatrix BusinessWorks engine tries to reconnect to the JMS Server if the connection is interrupted.

The property "bw.plugin.jms.exceptionListForReconnection" provides a configurable way to specify a comma separated list of exceptions. The JMS Client tries to reconnect to these servers if there are any interruptions in the connections and receives the exception listed against the property. Provide the fully qualified name of the exception.

The value for the property is a comma separated list of the exceptions.

This is applicable to all the JMS Clients, namely,

- [JMS Queue Requestor](#)
- [JMS Queue Sender](#)
- [JMS Topic Publisher](#)
- [JMS Topic Requestor](#)
- [SOAP Request Reply](#)

Mail Palette

The Mail palette is used to receive incoming email or send outgoing email.

Receive Mail

Process Starter



The Receive Mail process starter polls a POP3 mail server for new mail. When new mail is detected and retrieved, the Receive Mail process starter starts a new process for the process definition it resides in and passes the mail data to the next activity in the process flow.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Host	Yes	Host name or IP address for the mail server. The default port (110) is used to connect to POP3 the server. If your server is configured to use a different port, you can specify it in this field using the syntax: <code><hostName>:<portNumber></code> For example, myMailServer.myCo.com:76

Field	Global Var?	Description
Protocol	No	Protocol used by mail server. The drop-down list has two options: POP3 and IMAP.
Authentication	No	This drop-down list is displayed only when you select the IMAP option in the Protocol drop-down list. It has two options: Basic Authentication and OAuth 2.0. If you select Basic Authentication, then the User Name and Password fields appear. If you select OAuth 2.0, then the User Name and Access Token Value fields appear. Note: You can obtain the field information for Access Token Value, User Name, and Password from a Service Provider.
User Name	Yes	User name to use when logging into the POP3 server.
Password	Yes	Password to use when logging into the POP3 server.
Client ID	Yes	The Client ID issued to the client during the client registration process.
Client Secret	Yes	The Client Secret issued to the client during the client registration process.
Access Token URL	No	The endpoint for the authentication server. This is used to exchange the authorization code for the access token.
Scope	No	The scope of the access request.
Grant Type	Yes	Client Credentials. Currently Client Credentials as a grant type is supported.
Polling Interval	Yes	Polling interval (in seconds) to check for new mail. The

Field	Global Var?	Description
(sec)		default, if no polling interval is specified, is 5 seconds.
Delete Mail	Yes	If checked, mail gets deleted from the POP3 server once the process starter has retrieved it. Note: If you choose to delete mail, you should place a Checkpoint activity immediately after the Receive Mail process starter to ensure the message is not lost in the event of machine failure. Alternatively, you can choose to leave this field unchecked, and then later in the process definition perform a checkpoint then use the Confirm activity to confirm the message. The Confirm activity deletes the message from the mail server.
SSL	Yes	The Use SSL check box specifies that a Secure Sockets Layer (SSL) should be used to specify client SSL configuration. When this field is checked, the Configure SSL button is enabled.
Upgrade Configuration	No	This field only appears in Mail palette resources placed into process definitions prior to release 5.2.0. In previous releases, attachments were lists in the input of the activity. In release 5.2.0 or later, a new schema is used to handle MIME attachments. If you have a Mail palette resource created prior to release 5.2.0, this field should be unchecked to maintain backward compatibility. This field does not appear for new resources created in release 5.2.0 or later. Unchecking this field disables the Advanced and Output Headers tab and changes the output schema.

Configure SSL Button

The Configure SSL button allows you to configure the SSL parameters when the transport supports SSL. Check SSL to display Configure SSL dialog. Provide values in the following fields to configure SSL.

Field	Description
Trusted Certificates Folder	Specifies a folder in the project containing one or more certificates from the trusted certificate authorities. This folder is checked when a client connects to ensure that the server's certificate is from a trusted certificate authority. This prevents connections from rogue servers.
Identity	An Identity resource that contains the HTTP server's digital certificate and private key. For more information, see <i>TIBCO Designer™ Palette Reference</i> .
Verify Host Name	Specifies whether to verify that the host you are connecting to is the expected host. The host name in the host's digital certificate is compared against the value you specify in the Expected Host Name field. If the host name does not match the expected host name, the connection is refused.
Strong Cipher Suites Only	Specifies that only cipher suites with strong encryption should be used, if they are available on the host you are connecting to.

Test Connection Button

The **Test Connection** button tests the connection to the specified mail server. You can use this button to ensure that your **Receive Mail** activity is properly configured to receive mail from the specified mail server. This button is only enabled when the Host, Username, and Password fields on the Configuration tab are populated.

Advanced

The Advanced tab contains the following fields:

Field	Description
Provide Raw Message	When checked, this field specifies that the whole message is to be provided in binary form. The output schema changes to include an element named <code>rawMessage</code> that contains the message instead of <code>bodyText</code> and <code>mimeEnvelopeElements</code>. The Receive Mail process starter uses the <code>javax.mail</code> API. This API assumes that the content type of the message is <code>multipart/mixed</code>. When the multipart subtype is not mixed (such as <code>multipart/signed</code> or <code>multipart/alternative</code>) or when another content type is used (for example <code>application/*</code>), the <code>javax.mail</code> API may not correctly parse the message. Checking this field allows you to send the raw message to a Java Code activity that uses the Java Activation framework to implement the appropriate handlers to process the message. For more information on how the Java Activation framework can be used for this purpose, please consult the API documentation for <code>javax.activation</code> and <code>javax.mail</code>. You can also feed the binary message to the <code>javax.mail.MimeMessage</code> constructor in the form of a <code>java.io.InputStream</code> to construct a <code>MimeMessage</code> object. In that case, the default <code>javax.mail</code> API handler is used.
Write to File	Checking this field specifies that incoming messages whose body and attachments (or raw message size if the Provide Raw Message field is checked) exceed the specified threshold size should be written to a file instead of stored in memory. This allows you to accept large incoming messages without consuming a great deal of memory. When this field is checked, the Directory and Threshold Size fields appear. Leaving this field unchecked specifies that incoming messages should be kept in memory. Note: Once written, the files created by using this option are not deleted automatically. You must manage the storage used by these files and delete them when they are no longer used.
Directory	The directory to write messages that are above the specified threshold. The process engine does not attempt to create the directory if the specified directory does not exist. Therefore, create the directory before starting the process engine.

Field	Description
Create Non-Existing Directories	When checked, all directories in the path specified in the Directory field are created, if they do not already exist. If this field is unchecked and there are one or more directories in the specified path in the Directory field that do not exist, an exception is raised.
Threshold Message Data Size	The maximum size (in bytes) of an incoming message that can be kept in memory. Messages larger than the specified size are written to a file in the specified directory. The file's name is output so that subsequent activities in the process definition can access the file and read its contents. Specifying zero (0) in this field causes all incoming messages to be saved to a file.

Output Headers

Incoming messages may have custom headers. The Output Headers tab allows you to define a custom schema for the headers of the incoming mail messages. You can specify any standard header supported by the `javax.mail` package. For example, X-Mailer or X-Priority. Values of headers in the incoming message populates the corresponding output headers of the same name that you have defined.

For a complete description of using the Output Headers tab, see [Specifying Data Schema](#).

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created.

Field	Description
	For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i> .
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output

The output for the activity is the following.

Output Item	Datatype	Description
from	string	The email address of the sender of the email.
to	string	The recipient list of the email.
cc	string	The cc (carbon copy) list of the email.
replyTo	string	The reply to list of the email.
subject	string	The subject of the email.
sentdate	string	The date the email was sent.

Fields available when the Upgrade Configuration field is checked

Headers	complex	This element contains the schema you defined on the Output Headers tab. You can specify custom headers that can be included in incoming mail messages. You can specify any standard header supported by the <code>javax.mail</code> package. For example, X-Mailer

Output Item	Datatype	Description
		or X-Priority. Values of headers in the incoming message populates the corresponding output headers of the same name that you have defined.
rawMessage	binary	The content of the message in binary form. This element is only available when the Provide Raw Message field on the Advanced tab is checked.
bodyText	string	The text of the email message. The Provide Raw Message field on the Advanced tab must be unchecked for this field to appear.
mimeEnvelopeElement	complex	This element contains the message attachments. This element contains a repeating element named mimePart that contains each mime attachment. The Provide Raw Message field on the Advanced tab must be unchecked for this field to appear.
mimeHeaders	complex	This element contains the mime header for each mimePart. Mime headers can contain the following information: • content-disposition • content-type • content-transfer-encoding • content-id • other mime header information For more information about MIME headers and their syntax, see http://www.faqs.org/rfcs/rfc2045.html.

Output Item	Datatype	Description
Note: When the content type is specified as "text/*" (for example, "text/xml"), the attachment content is expected to be in either the textContent output element or the file name storing the attachment is expected to be in the fileName output element. When the content type is anything other than "text/*", the attachment content is expected to be in either the binaryContent output element or the file name storing the attachment is expected to be in the fileName output element.		
binaryContent textContent fileName	choice	This element contains the mime attachment. The element can be one of the following: • binaryContent – content of the attachment when the attachment is binary data. • textContent – content of the attachment when the attachment is text data. • fileName – the file name of the attachment when the attachment is stored in the file system (see Advanced tab for more information).

Fields available when the Upgrade Configuration field is unchecked

bodyText	string	The text of the email message.
attachmentNames	string	A list of names for each attachment to the message. These names are typically the file names of the attachments.

Output Item	Datatype	Description
Note: The attachmentNames and attachmentData elements are corresponding lists. That is, for each item in the attachmentName element, there must be a corresponding item in the same position in the attachmentData list. For example, if there are three attachments, both elements must contain three items and item in the attachmentNames element corresponds to the element in the same position in the attachmentData list.		
attachmentData	binary	A list of attachments to the message.
contentType	string	The MIME content type specified in the message.

Send Mail

Activity



The **Send Mail** activity sends an email by way of a SMTP server.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.

Field	Global Var?	Description
Host	Yes	Host name for the SMTP server. The default port (25) is used to connect to the SMTP server. If your server is configured to use a different port, you can specify it in this field using the syntax: <pre><hostName>:<portNumber></pre> For example, myMailServer.myCo.com:37
SSL	No	The Use SSL check box specifies that Secure Sockets Layer (SSL) should be used to specify a client SSL configuration. When this field is checked, the Configure SSL button is enabled.
Authenticate	No	Specifies whether you want to authenticate to the mail server. When selected, the Authentication Type field appears. Note: The mail server must be configured to require authentication for sending mail. If the mail server is not configured to require authentication, the Authentication Type field is ignored.
Authentication Type	No	Two types of authentication are supported: OAuth 2.0 and Basic Authentication. If you select Basic Authentication, then the User Name and Password fields appear. If you select OAuth 2.0, then the User Name and Access Token Value fields appear. Note: You can obtain the field information for Access Token Value, User Name, and Password from a Service Provider.

Field	Global Var?	Description
User Name	Yes	User name to use when authenticating to the mail server.
Password	Yes	Password to use when authenticating to the mail server. Passwords entered in this field are encrypted when stored in the project.
Access Token Value	Yes	The value authorized by the authorization server's token endpoint.
Upgrade Configuration	No	This field only appears in Mail palette resources placed into process definitions prior to release 5.2.0. In previous releases, attachments were lists in the input of the activity. In release 5.2.0 or later, a new schema is used to handle MIME attachments. If you have a Mail palette resource created prior to release 5.2.0, this field should be unchecked to maintain backward compatibility. This field does not appear for new resources created in release 5.2.0 or later. Unchecking this field disables the Input Headers tab and changes the input schema.

Configure SSL Button

The **Configure SSL** button allows you to configure the SSL parameters when the transport supports SSL. Check SSL to display Configure SSL dialog. Provide values in the following fields to configure SSL.

Field	Global Var?	Description
Trusted Certificates Folder	No	Folder in the project containing one or more certificates from trusted certificate authorities. This folder is checked when an invocation is made to the partner service to ensure that the server is trusted. This prevents connections to rogue servers.

Field	Global Var?	Description
Identity	No	This is an identity resource that contains the client's digital certificate and private key. For more information, see <i>TIBCO Designer™ Palette Reference</i> .
Verify Host Name	No	Specifies whether to verify that the host you are connecting to is the expected host. The host name in the host's digital certificate is compared against the value you specify in the Expected Host Name field. If the host name does not match the expected host name, the connection is refused.
Strong Cipher Suites Only	No	Specifies that only cipher suites with strong encryption should be used, if they are available on the host you are connecting to.

Support for STARTTLS

Some Mail servers (such as Microsoft Exchange 2010) support SMTP over SSL using the STARTTLS feature. ActiveMatrix BusinessWorks supports this using the property, "bw.plugin.mail.enableStartTLS," which is set to default as "false". Select the 'Use SSL' check box when using such servers and provide the Trusted Certificates folder.

When not using such servers, provide this engine property for SMTP with value as "false".

Advanced

The Advanced tab has the following fields.

Field	Global Var?	Description
Allow	No	By default, email IDs specified in the to, from, cc, bcc, or replyTo input

Field	Global Var?	Description
Non-Standard Email IDs		elements must conform to the RFC 822 standard for email addresses. Checking this box allows you to specify email IDs that do not conform to the RFC 822 standard. This is useful if you want to send mail to a server that has extensions for sending email to devices such as a fax machine. For more information about email address syntax, see http://www.faqs.org/rfcs/rfc822.html. Note: When this field is unchecked, not all of the RFC 822 syntax rules are enforced. For example, addresses composed of simple names (with no "@domain" part) are allowed. For more information on enforcing RFC 822, see the description of the strict flag of the parse method of <code>javax.mail.internet.InternetAddress</code> at http://java.sun.com/products/javamail/javadocs/javax/mail/internet/InternetAddress.html.

Input Headers

You may want to add custom headers to outgoing mail messages. The Input Headers tab allows you to define a custom schema for the headers of the outgoing mail message. You can specify any standard header supported by the `javax.mail` package. For example, X-Mailer or X-Priority.

For a complete description of using the Input Headers tabs, see [Specifying Data Schema](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
username	string	Username to use when authenticating to the mail server. Note: The mail server must be configured to require authentication for sending mail. If the mail server is not configured to require authentication, the username and password elements are ignored.
password	string	Password to use when authenticating to the mail server.
from	string	The email address of the sender.
to	string	The recipient list for the email. This is a repeatable element, so that you can send mail to more than one recipient. You can also supply a list of recipients in a single string by using either a comma or a semicolon (but not both in the same string) to separate the recipients.
cc	string	The cc (carbon copy) list for the email. This is a repeatable element, so that you can place more than one recipient on the CC list. You can also supply a list of CC recipients in a single string by using either a comma or a semicolon (but not both in the same string) to separate the CC recipients.
bcc	string	The bcc (blind carbon copy) list for the email. This is a repeatable element, so that you can place more than one recipient on the BCC list. You can also supply a list of BCC recipients in a single string by using either a comma or a semicolon

Input Item	Datatype	Description
		(but not both in the same string) to separate the BCC recipients.
replyTo	string	The reply to list for the email. This is a repeatable element, so that you can place more than one recipient on the reply to list. You can also supply a reply to list in a single string by using either a comma or a semicolon (but not both in the same string) to separate the addresses in the list.
subject	string	The subject of the email.
sentdate	string	The date stamp for the email. Note: The email server supplies the actual date stamp for the email, so this input item is ignored. This item is deprecated and may be removed in a future release. Do not attempt to place a value in this input item.

Fields available when the Upgrade Configuration field is checked

bodyElement	complex	This element contains a choice element. You can either supply the bodyText element or the fileName element. The bodyText element is a string containing the body of the message. The fileName element is the path and name of a file containing the body of the message.
Headers	complex	This element contains the schema you defined on the Input Headers tab. You can specify custom headers for outgoing mail messages. You can specify any standard header supported by

Input Item	Datatype	Description
		the javax.mail package. For example, X-Mailer or X-Priority. If you want to use attachments, make sure the Content-Type header is specified as multipart/* (where * is a valid subtype of multipart).
mimeEnvelope Element	complex	This element contains the message attachments. This element contains a repeating element named mimePart that contains each mime attachment.
mimeHeaders	complex	This element contains the mime header for each mimePart. Mime headers can contain the following information: • content-disposition – To suggest a filename for an attachment, use ";filename=<filename>" in this element. Note: E-mail servers may alter or choose to ignore the suggested name. • content-type • content-transfer-encoding • content-id other mime header information For more information about MIME headers and their syntax, see http://www.faqs.org/rfcs/rfc2045.html.

Input Item	Datatype	Description
Note: When the content type is specified as "text/*" (for example, "text/xml"), the attachment content is expected to be in either the textContent input element or the file name storing the attachment is expected to be in the fileName input element. When the content type is anything other than "text/*", the attachment content is expected to be in either the binaryContent input element or the file name storing the attachment is expected to be in the fileName input element.		
binaryContent textContent fileName	choice	This element contains the mime attachment. The element can be one of the following: • binaryContent – content of the attachment when the attachment is binary data. • textConetnt – content of the attachment when the attachment is text data. • fileName – the file name of the attachment written to the disk.
Fields available when the Upgrade Configuration field is unchecked		
bodyText	string	The text of the email message.
attachmentNames	string	A list of names for each attachment to the message. These names are typically the file names of the attachments.

Input Item	Datatype	Description
Note: The attachmentNames and attachmentData elements are corresponding lists. That is, for each item in the attachmentName element, there must be a corresponding item in the same position in the attachmentData list. For example, if there are three attachments, both elements must contain three items and item in the attachmentNames element corresponds to the element in the same position in the attachmentData list.		
attachmentData	binary	A list of attachments to the message.
contentType	string	The MIME content type for the message. You can specify a character encoding in this element for the encoding of the body of the message.

Output

This activity produces no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When
InvalidInputException	Input is not valid.
UnknownHostException	Mail server host name could not be resolved. The exception contains the mail server host name.
SendFailedException	The send operation failed for any reason.

i **Note:** When you want to create a new session instead of using previously created session for the Mail palette activities, set the following property to false.

`bw.plugin.mail.reuseSession=false.`

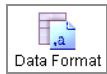
Parse Palette

The **Parse** palette provides shared configuration resources and activities for parsing and rendering formatted text. This is useful if you want to transform formatted lines of text into a data schema. This is also useful if you want to transform a data schema into a formatted text string.

The text lines can be formatted either by delimiters separating each field or offsets can be specified to determine where each field begins and ends.

Data Format

Shared Configuration

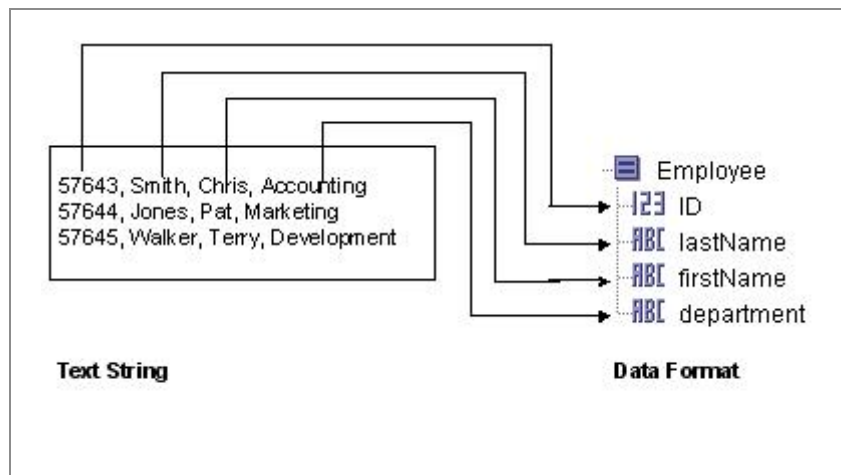


The Data Format resource contains the specification for parsing or rendering a text string using the [Parse Data](#) and [Render Data](#) activities. This shared configuration resource specifies the type of formatting for the text (delimited columns or fixed-width columns), the column separator for delimited columns, the line separator, and the fill character and field offsets for fixed-width columns. You must also specify the data schema to use for parsing or rendering the text.

When parsing text, each column of an input line is transformed into the corresponding item in the specified data schema. The first column of the text line is turned into the first item in the data schema, the second column is transformed into the second item, and so on. Each line is treated as a record, and multiple lines result in a repeating data schema containing the lines of the input text string.

[Parsing a text string into a data schema](#) illustrates how an input text string is parsed into a specified data schema.

Figure 19: Parsing a text string into a data schema



When rendering text, each record in the input data schema is transformed into a line of output text. The first item of the data schema is transformed into the first column of the text line, the second item is transformed into the second column, and so on. Each record in a repeating data schema is transformed into a separate line in the output text string. Rendering a data schema into a text string is exactly the opposite process of parsing a text string into a data schema. Rendering is the reverse of the process illustrated in [Parsing a text string into a data schema](#).

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the resource.
Description	Short description of the shared resource.
Format Type	The type of formatting for the text. The text can be either "Delimiter separated" or "Fixed format". In delimiter-separated text, each column is separated by a delimiter character, specified in the Col Separator field. Each line is separated by the character specified in the Line Separator field. For more information,

Field	Description
	see Delimiter Separated Fields. In Fixed format text, each column occupies a fixed position on the line. For fixed format text, you must specify the Fill Character, the line length, and the column offsets. For more information, see Field Offsets.
Col Separator	This field specifies one or more separator characters between columns when "Delimiter separated" is specified in the Format Type field. When rendering text, each element in the input data schema is separated by the column separator in the output text string. If more than one character is specified in this field, the Render Data activity places the entire string specified in this field between each column. For example, if ";;" is specified in this field, then ";;" appears between each column in the rendered string. When parsing text, each column becomes an element in the output data schema. If more than one character is specified in this field, the Parse Data activity uses the rule specified in the Col Separator Parse Rule field to determine how to parse the data.
Col Separator Parse Rule	Specifies the rule to use for multiple column separator characters when parsing data. The choices are the following: • Treat all characters as entered as a single column separator string The characters entered into the Col Separator field are treated as a single string that acts as a separator. For example, if the specified Col Separator is ";;", then Apple;;Orange;;Pear is treated as three columns. • Treat each character entered as a potential column separator Any of the characters acts as a column separator. For example, if the specified Col Separator is ";;", then Apple;Orange;Pear is treated as three columns.
Line Separator	Specifies the character(s) that determine the end of each line. This field allows you to define a custom line separator. By default, new line, carriage return and line feed are defined and can be selected from the drop down

Field	Description
	list. When parsing text, each line is treated as a new record in the output data schema. When rendering text, each data record is separated by the line separator character in the output text string. The last line in your input file must be terminated by the specified line separator. Note: ActiveMatrix BusinessWorks does not support specifying the line separator as part of the data.
Fill Character	When processing fixed format columns, this is the type of character that is used to fill the empty space in a column and between columns. This is only available when "Fixed format" is specified in the Format Type field. This fill character is only used by the Render Data activity. You can select one of the following for this field: • Space – fills with a space • Dash – fills with a dash • Others – allows you to specify your own custom fill character in the Fill With field. For example, you have a column that holds an integer and the specified width is 10. One row has the value "588" for that column. Because the width of 588 is three and the column width is 10, the remaining 7 characters are filled with the specified fill character.
Fill With	This field is only available when Others is selected in the Fill Character field. This field specifies the fill character to use to pad unused characters in fixed-width columns. Only one character can be specified. The first character is the fill character and any other characters specified in this field are ignored.

Data Format

The Data Format tab allows you to define a custom schema for the text.

You can define your own datatype on this tab, and you can reference XML schema or ActiveEnterprise classes stored in the project. Once defined, the data specified on the Data Format tab is used to parse a text string into the specified schema or render the specified schema as a text string.



Warning: Data Format does not support nested schema. You can only specify basic plain schema with no nesting.

For a description of how to define a schema, see [Specifying Data Schema](#).

Delimiter Separated Fields

When processing delimiter-separated text, each field in the input line is separated by the delimiter specified by the Column Separator field. Leading and trailing spaces are stripped from each field and the specified Line Separator determines when a new record starts. [Data Format](#) illustrates an series of input lines containing comma-separated fields, each record on one line.

In some situations, you may not be able to choose a column separator character that does not appear in any column data. For example, if you choose a comma as the column separator, there may be commas in some of the column values. To process data that contains column separator characters in a column, you can surround the column with double quotes (" "). Double quotes also allow you to include leading and trailing spaces as well as line breaks in a field. If you want to have a double quote appear in a field, escape the double quote by using two consecutive double quotes. That is, use "" to represent a double quote in a field.

The following data illustrates input lines with each field separated by commas. Some fields, however, contain commas, leading or trailing spaces, double quotes, and line breaks.

```
57643, Smith, "Chris", Accounting , "Statement: Be prepared!"
57644, Jones, "Pat ", Marketing , "Statement: To paraphrase JFK, ""Ask not what your company
can do for you,
ask what you can do for your company.""
57645, Walker, "Terry", Development , "Statement: My goal is to be CEO someday."
```

Notice that Pat Jones' statement spans two lines and contains double quotes as well as a comma. The entire field is surrounded by double quotes, so it is still treated as part of the same record.

Field Offsets

When processing fixed format text, you must specify the line length and the column offsets. This allows a **Parse Data** or **Render Data** activity to determine where columns and lines begin and end. The Field Offsets tab allows you to specify the format of fixed-width text.

The line length is the total length of each input line, including the line separator character (s). Include the appropriate number of characters for the selected line separator on the Configuration tab to the total length of each line.

The column offset is the starting and ending character position on each line for the column. Each line starts at 0 (zero). For each column of the line, you must specify the name of the data item associated with this column (this is the same name you specified for the corresponding element in the data schema), the starting offset for the column, and the ending offset for the column.

 **Tip:** It is a good idea to have each column offset begin where the last column offset ended. Many fixed format data files are used by databases (for example, ISAM files) or are generated by automated processes. These types of files have rigid file record formats and may not have additional padding space between columns.

When you define each column offset to begin where the last column offset ends, the data can be read more quickly by ActiveMatrix BusinessWorks because the bytes of the input records can be read in sequentially.

Consider the following text file. The first two lines of the file indicate offset numbers (each 0 indicates another 10 characters), and the fill character between columns is spaces:

```
0      12      30      45
012345678901234567890123456789012345678901234567
57643   Smith   Chris   Account
57644   Jones   Pat     Marketing
57645   Walker   Terry   Development
```

[Fixed-width text strings and field offsets](#) illustrates the Field Offset tab for the file above. Notice that the line length is specified as 60, even though the offsets end at character number 58. The line separator is specified as "Carriage Return/Line Feed (windows)", so this adds two additional characters for a total line length of 60.

Figure 20: Fixed-width text strings and field offsets

The screenshot shows the 'Field Offsets' tab in the Parse Data configuration. At the top, there's a 'Line length' field set to 60. Below it is a table with three columns: 'name', 'start', and 'end'. The table contains the following data:

name	start	end
ID	0	12
lastName	12	30
firstName	30	45
department	45	58

Parse Data

Activity



The **Parse Data** activity takes a text string or input from a file and processes it, turning it into a schema tree based on the specified [Data Format](#) shared configuration.

You can use any mechanism to obtain or create a text string for processing. For example, you can use the [Read File](#) activity to obtain text from a file, or you can retrieve a text field from an adapter message. You can also specify a text file to read using this activity.

You might use this activity in a number of situations. For example, you may have a file that consists of multiple lines of comma-separated values (as in data obtained from a spreadsheet). You may also want to insert that data into a database table. To do this, read and parse the file into a data schema with the **Parse Data** activity. Then use a [JDBC Update](#) activity to insert the data schema into a database table.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process

Field	Global Var?	Description
		definition.
Description	No	Short description of the activity.
Data Format	No	The Data Format shared configuration to use when parsing the text input. For more information, see Data Format .
Input Type	No	Specifies the type of input for this activity. Input can either be a text string or a file. If the input is a text string, supply the string to the <code>textString</code> input item. If the input is a file, supply the file name and location to the <code>fileName</code> input item.
Encoding	Yes	The encoding of the input file. This field is only available when "File" is chosen in the Input Type field. Any valid Java encoding name can be used.
Skip Blank Lines	No	Skips any empty records when parsing the text input. When this field is unchecked, parsing stops at the first blank line encountered in the input.
Manually Specify Start Record	No	Allows you to specify the record in the input where you want to start parsing. This is useful if you have a large number of records and you want to read the input in parts (to minimize memory usage). Checking this check box causes the <code>startRecord</code> input item to appear. For more information on how to read the input stream in parts, see Parsing a Large Number of Records .
Strict Validation	No	Validates every input line for the specified number of fields for the "Fixed Format" text. For example, if the format states that there are three fields per line and this field is checked, all lines in the input must contain three fields.

Field	Global Var?	Description
Continue on Error	No	Continues parsing the next record in the input after encountering an error, if any. If an error occurs, the error information is separated from the output of the successfully parsed records and is provided in the output schema of the activity. For more information on how the error information for the records that failed parsing is provided, see Output. Error information is provided only for records specified as Required. If the data item is specified as Optional, the process completes parsing the input successfully. Irrespective of whether this field is checked or not, the Parse Data activity quits when any data validation errors occur.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
text	string	The text string to parse. This input item is only available when "String" is specified in the Input Type field on the Configuration tab.
fileName	string	The location and name of the file to read. The file's contents is used as the input text string for this activity. This input item is only available when "File" is specified in the Input Type field on the Configuration tab.

Input Item	Datatype	Description
startRecord	number	The line number of the input stream where you want to begin parsing. All lines before the specified line are ignored. This input item is only available if the Manually Specify Start Record field on the Configuration tab is checked. The input stream begins with line number 1 (one). This is useful if you want to read the input stream in parts to minimize memory usage. For more information, see Parsing a Large Number of Records.
noOfRecords	number	The number of records to read from the input stream. Specify "-1" if you want to read all records in the input stream. This is useful if you want to read the input stream in parts to minimize memory usage. For more information, see Parsing a Large Number of Records.
SkipHeaderCharacters	integer	The number of characters to skip when parsing. This allows you to skip over any file headers or other unwanted information.

Output

The output for the activity is the following.

Output Item	Datatype	Description
Rows	complex	This output item contains the list of parsed lines from the input. This is useful to determine the number of records parsed by this activity.

Output Item	Datatype	Description
		The schema specified by the Data Format resource is contained in this output item.
schema	complex	The schema containing the data from the parsed input text. This output item contains zero or more parsed records.
ErrorRows		This output item is available when Continue on Error is checked, and error(s) while parsing the records in the input. <<raw input data is put in the error string>> This field contains the list of error lines for the records from the input that failed parsing. See the following example explaining successful parsing of records specified as Optional and Mandatory. During parsing, if any of the Required data items are missing, the records are displayed in the Error Row. If there are missing Optional items, the records are displayed in the Output Row.
EOF	boolean	true if no more records are available for parsing. false if there are more records available. This output item is useful for checking if there are no more records in the input stream when you are reading the input in parts to preserve memory. For more information, see Parsing a Large Number of Records.

Example

The example contains two input records containing five data items, ID, FName, LName, Address and Age. All the items are declared mandatory.

,FN1,LN1,Pune,99

2,FN2,LN2,Pune,33

Shown below is the output of the two records, after parsing them.

As the field "ID" is missing from the first record, the entire record appears in the ErrorRows element.

```

<?xml version = "1.0" encoding = "UTF-8"?>
<Output>
  <Rows>
    <root>
      <ID>2</ID>
      <FName>FN2</FName>
      <LName>LN2</LName>
      <Address>Pune</Address>
      <Age>33</Age>
    </root>
  </Rows>
  <ErrorRows>
    <ErrorString>,FN1,LN1,Pune,99</ErrorString>
  </ErrorRows>
  <EOF>true</EOF>
</Output>

```

When the ID field is declared **Optional**, and is missing, it is not an error. The records appear in the output string :

```

<?xml version = "1.0" encoding = "UTF-8"?>
<Output>
  <Rows>
    <root>
      <FName>FN1</FName>
      <LName>LN1</LName>
      <Address>Pune</Address>
      <Age>99</Age>
    </root>
    <root>
      <ID>2</ID>
      <FName>FN2</FName>
      <LName>LN2</LName>
      <Address>Pune</Address>
      <Age>33</Age>
    </root>
  </Rows>
  <ErrorRows/>
  <EOF>true</EOF>
</Output>

```

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
FileNotFoundException	The specified input file cannot be located.
BadDataFormatException	The input format is not valid.

Parsing a Large Number of Records

The input for this activity is placed in a process variable and takes up memory as it is being processed. If you are reading a large number of records from a file, the process may consume significant machine resources. To avoid using too much memory, you may want to read the input in parts, parsing and processing a small set of records before moving on to the next set.

To process a large number of records, follow this procedure:

Procedure

1. Create a **Parse Data** activity.
2. Specify the fields on the Configuration tab. Check the check box for the Manually Specify Start Record field.
3. Select the **Parse Data** activity and click the **group** icon on the tool bar to create a group containing the **Parse Data** activity. For more information about working with groups, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.
4. Specify "Repeat Until True Loop" as the Group action, and specify an index name (for example, "i").
5. The loop should exit when the EOF output item for the **Parse Data** activity is set to true. For example, the condition for the loop could be set to the following: `string($ParseData/Output/done) = string(true())`
6. Set the noOfRecords input item for the **Parse Data** activity to the number of records

you want to process for each execution of the loop.

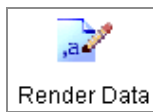
7. If you do not check the Manually Specify Start Record field on the Configuration tab of the **Parse Data** activity, the loop processes the specified `noOfRecords` with each iteration until there are no more input records to parse.

You can optionally check the Manually Specify Start Record field to specify the `startRecord` on the Input tab. If you do this, you must create an XPath expression to properly specify the starting record to read with each iteration of the loop. For example, the count of records in the input starts at zero, so the `startRecord` input item could be set to the current value of the loop index minus one. For example, `$i - 1`.

The procedure above is a general guideline for creating a loop group for parsing a large set of input records in parts. You may want to modify the procedure to include additional processing of the records, or you may want to change the XPath expressions to suit your business process.

Render Data

Activity



The **Render Data** activity takes an instance of a data schema and renders it as a text string. The schema processed is based on a specified [Data Format](#) shared configuration.

You may want to use this activity in a number of situations. For example, you may retrieve a result set from a database table. You may then want to format this result set as a formatted text string (with line breaks between each row in the result set), and then write that text string out to a file. You would use the **Render Data** activity to render the data schema as a formatted text string. Then you would use the **Write File** activity to write the string to a file.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Data Format	No	The Data Format shared configuration to use when rendering the text output. For more information, see Data Format .

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
Rows	complex	The element containing the list of items to render.
root	complex	The elementcomplex containing data schema (specified by the Data Format field on the Configuration tab) to render as a text string. This is a repeating element so that you can render more than one output record.

Output

The output for the activity is the following.

Output Item	Datatype	Description
text	string	The output text string as a result of rendering the specified data schema. Line breaks separate records of the data schema.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
BadDataFormatException	The output format is not valid.

Policy Palette

The **Policy** palette enables you to specify security policies for inbound and outbound SOAP messages. The security policies follow Web Services Security: SOAP Message Security 1.1 (WS-Security 2004) OASIS Standard Specification. You can find out more about this standard at <http://docs.oasis-open.org/wss/v1.1>.

For more information about using the resources in the **Policy** palette, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Security Policy

Shared Configuration



The Security Policy shared configuration resource specifies a security policy that can be used for inbound or outbound SOAP messages. The security policy can include any combination of the following characteristics:

- Authentication – whether messages must be authenticated. Authentication can be performed either with usernames and passwords or by way of X.509 compliant certificates.
- Integrity – whether messages must be validated with a signature to ensure the message has not been altered since its creation.
- Confidentiality –whether messages should be encrypted or decrypted.
- Timeout – whether messages should expire after a certain time.

For more information about using security policies for web services, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.
Policy Type	No	Specifies the type of communication this policy applies to. Select one of the following from the list of choices: • inbound • outbound • inbound fault • outbound fault
Authentication	No	Check this box if you want to specify that authentication should be part of this policy. Checking this box makes the fields on the Authentication tab available for editing.
Integrity	No	Check this box if you want to specify that messages should be signed or that signed messages should be verified. Checking this box makes the fields on the Integrity tab available for editing.
Confidentiality	No	Check this box if you want to specify that messages should be encrypted and decrypted. Checking this box makes the fields on the Confidentiality tab available for editing.
Timeout	No	Check this box if you want to add or verify timestamps on messages. Checking this box makes the fields on the Timeout tab available for editing.
Custom Password Lookup	No	Check the box in this field if you want to use your own custom password lookup class. By default, authentication is performed through TIBCO Administrator. This field allows you to specify a class for looking up passwords in your system, such as a database, file system, or LDAP system.

Field	Global Var?	Description
		For more information about custom password lookup, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i> .
Custom Password Callback Java Class	No	This field specifies an AliasLibrary resource containing your custom password lookup class and any referenced classes (such as third-party libraries to access an LDAP server). Use the Browse button to locate the AliasLibrary resource storing your class, then use the Show Class Browser button in the Class field to locate your class.
Security Header Layout	No	This field is available only when the policy type is outbound or outbound fault. Specifies layout rules to be applied when adding elements to the security header of a SOAP message. The four layouts are: • Lax • Strict • LaxTimeStampFirst - This value is available only if the Timeout field is selected. • LaxTimeStampLast - This value is available only if the Timeout field is selected.

Authentication

The Authentication tab specifies the characteristics for authenticating to the SOAP message server. Inbound messages can be authenticated against a list of trusted certificates. Outbound messages can specify the identity to use to authenticate to the external SOAP server.

The Authentication tab has the following fields.

Field	Global Var?	Description

Inbound or Inbound Fault Policy Type Fields

Supported Security Tokens	No	Specifies the security tokens you want to allow in inbound messages. You can select one or more of the following supported types: • X.509 Token • UsernamePassword Token
Trusted Certificates Folder	No	Specifies the folder containing the trusted certificates for binary security token authentication.

Outbound or Outbound Fault Policy Type Fields

Security Token	No	Specifies the type of security tokens you want to allow in inbound messages. You can select one of the following supported types: • X.509 Token • UsernamePassword Token
X.509 Identity	No	When X.509 Token is selected in the Security Token field, this field specifies the Identity resource containing the X.509 compliant identity file. For more information about Identity resources, see <i>TIBCO Designer™ Palette Reference</i> .
Username Password Identity	No	When UsernamePassword Token is selected in the Security Token field, this specifies the Identity resource that contains the username and password. For more information about Identity resources, see <i>TIBCO Designer™ Palette Reference</i> .
Password Type	No	Specifies whether you want to use text or digest passwords.

Field	Global Var?	Description
Certificate Alias	No	When the Identity resource specified in the X.509 Identity field is of type JKS or JCEKS, specify the certificate alias in this field to identify the private and public key pair.
Use Password for the Key	No	When the Identity resource specified in the X.509 Identity field is of type JKS or JCEKS, select the check box to configure a password for the private key. Note: If the check box is not selected, the password has to be the same for the keystore and the private keys stored inside it.
Alias Password	No	This field is enabled when the Use Password for the Key check box is selected. Specify a password for the private key.

Integrity

The Integrity tab specifies the characteristics of the signatures attached to the messages. Signatures can be used to ensure that messages are not altered after creation. The integrity of inbound messages can be checked against the trusted root certificates and public certificates. Outbound messages can specify the certificate to be used to sign the outgoing message.

The Integrity tab has the following fields.

Field	Global Var?	Description
Inbound or Inbound Fault Policy Type Fields		
Supported Signature Methods	No	The algorithm used to check the signatures of incoming messages. You can select one or more of the following:

Field	Global Var?	Description
		• SHA1 • SHA256 • SHA384 • SHA512
Supported Security Tokens	No	Is the security token to use for the signature. You can select one or more of the following: • X.509 Token • UsernameToken
Trusted Certificates Folder	No	Is the folder containing the trusted certificates for signature verification. Note: The certificates in the trusted folder are only necessary when the authenticating user is the same as the user who signed the message. In this case, the message contains the public key, the receiver must verify against the trusted certificate. However, if the authenticating user is not the same as the user who signed the message, the user must define a subject key identity that holds the public key.
Subject Key Identity	No	Specifies an Identity resource containing a keystore that holds an X.509 certificate. The inbound message must match the subject key contained in the certificate.

Outbound or Outbound Fault Policy Type Fields

Signature Method	No	The algorithm used to create signatures for outgoing messages. You can select one of the following: • SHA1 • SHA256 • SHA384

Field	Global Var?	Description
		• SHA512
Security Token	No	Is the type of security token to use for the signature. You can select one of the following: • X.509 Token • UsernameToken
Username Password Identity	No	When UsernameToken is selected in the Security Token field, this shows the Identity resource that contains the username and password. for more information about Identity resources, see <i>TIBCO Designer™ Palette Reference</i>.
Password Type	No	Specify whether you want to use text or digest passwords.
X.509 Identity	No	When you select X.509 Token in the Security Token field, this field indicates the Identity resource containing the X.509 compliant identity file. For more information about Identity resources, see <i>TIBCO Designer™ Palette Reference</i>.
Key Identifier Type	No	When you select X.509 Token in the Security Token field, this field specifies all the references to X.509 token types for signing. You can select one of the following: • Direct Reference • Subject Key Identifier • Issuer Serial Note: If the Key Identifier Type selected is of type Subject Key Identifier, ensure that the X509 Certificate used contains the Subject Key Identifier information.

Field	Global Var?	Description
Certificate Alias	No	When the Identity resource specified in the X.509 Identity field is of type JKS or JCEKS, specify the certificate alias in this field to identify the private and public key pair.
Use Password for the Key	No	When the Identity resource specified in the X.509 Identity field is of type JKS or JCEKS, select the check box to configure a password for the private key. Note: If the check box is not selected, the password has to be the same for the keystore and the private keys stored inside it.
Alias Password	No	This field is enabled when the Use Password for the Key check box is selected. Specify a password for the private key.

Confidentiality

The Confidentiality tab specifies the encryption characteristics of messages. Inbound messages can be decrypted based on algorithms specified in the message security header and the defined private key. Outbound messages can be encrypted based on algorithms stored in a public key.

The Confidentiality tab has the following fields.

Field	Global Var?	Description
Inbound or Inbound Fault Policy Type Fields		
Supported Encryption Algorithm	No	Symmetric key algorithm used to decrypt incoming messages. You can select one or more of the following: 3DES

Field	Global Var?	Description
		• AES-128 • AES-256 Note: To use Encryption Algorithm AES-256, refer to Installing Unlimited Jurisdiction Files
Private key Identifier Type	No	Specifies whether the private key is one of the following: • X.509 Token
X.509 Identity	No	When X.509 Token is selected in the Private Key Identifier Type field, this field specifies the Identity resource containing the X.509 compliant private key file to use to decrypt the message. For more information about Identity resources, see <i>TIBCO Designer™ Palette Reference</i>.
Use Password Protected Key	No	When the Identity resource specified in the X.509 Identity field is of type JKS or JCEKS, select the check box to configure a password for the private key. Note: If the check box is not selected, the password has to be the same for the keystore and the private keys stored inside it.
Certificate Alias	No	When the Identity resource specified in the X.509 Identity field is of type JKS or JCEKS, specify the certificate alias in this field to identify the private and public key pair.
Alias Password	No	Specify the password for the private key.

Outbound or Outbound Fault Policy Type Fields

Encryption Algorithm	No	Algorithm used to encrypt outgoing messages. You can select one of the following:

Field	Global Var?	Description
		• 3DES • AES-128 • AES-256 Note: To use Encryption Algorithm AES-256, refer to Installing Unlimited Jurisdiction Files
Public Key	No	Identity resource containing the X.509 compliant public key file to use to encrypt the message. For more information about Identity resources, see <i>TIBCO Designer™ Palette Reference</i> .
Key Identifier Type	No	When you select the X.509 Token in the Security Token field, this field specifies all the references to X509 token types for encryption. You can select one of the following: • Direct Reference • Subject Key Identifier • Issuer Serial Note: If the Key Identifier Type selected is of type Subject Key Identifier, ensure that the X509 Certificate used contains the Subject Key Identifier information.

Installing Unlimited Jurisdiction Files

Java vendors ship a default set of policy files that do not permit unlimited strength cryptography. In countries exempt from these restrictions, you can download and install an unlimited strength set of the policy files.

The default set of policy files restricts usage of 256-bit AES.

Follow these steps to install the unlimited strength policy files:

Procedure

1. Download the Java Cryptography Extension(JCE) Unlimited Strength Jurisdiction

Policy Files from the JRE vendor.

2. Create a backup for the files located in `TIBCO_HOME/tibcojre/jre_version/lib/security`.
3. Extract the files downloaded in Step 1 to `TIBCO_HOME/tibcojre/jre_version/lib/security`.
4. Restart all the running TIBCO applications.

Timeout

The Timeout tab specifies the characteristics of message timeout. Inbound messages can be rejected after the specified number of seconds. Outbound messages can be set to expire after the specified number of seconds.

The Timeout tab has the following fields.

Field	Global Var?	Description
Inbound or Inbound Fault Policy Type Fields		
Reject After (seconds)	No	The creation time of incoming messages is compared to the time the message was received. If the difference in time is greater than the number of seconds specified in this field, the message is rejected.
Outbound or Outbound Fault Policy Type Fields		
Expire In (seconds)	No	Outgoing and error messages expire after the specified number of seconds.

Security Policy Association

Shared Configuration



The Security Policy Association shared configuration resource associates a **Security Policy** with a particular Service operation or SOAP resource in your project. This allows you to specify security policies for inbound or outbound

SOAP messages on a per-operation or per-activity basis.

When a security policy is attached to a resource that receives SOAP messages (for example, [SOAP Event Source](#), or the reply message of a [SOAP Request Reply](#) activity), the associated security policy is used for the incoming message. For example, if authentication is specified in the security policy, then the identity of the sender of incoming messages is checked against a list of trusted identities.

When a security policy is attached to a resource that sends SOAP messages (for example, [SOAP Request Reply](#), [SOAP Send Fault](#), or [SOAP Send Reply](#)), the security the associated security policy is used for the outgoing message. For example, if encryption is specified in the security policy, the outgoing message is encrypted before it is sent.

Note: Security Policy Association shared configuration resources are not referenced by resources in process definitions. Therefore, they are not automatically included in Enterprise Archive files. You must manually add Security Policy Association resources to the Shared Archive within an Enterprise Archive for the associations to work properly in a deployed project. For more about adding resources to the Shared Archive, see *TIBCO ActiveMatrix BusinessWorks™ Administration*.

Warning: To run a project with security policy associations successfully, ensure that all the policy associations in the project are valid. Any invalid associations must be removed from the project before running the project.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.

Field	Global Var?	Description
Apply Policy To	No	The SOAP resources or Service operations to which you want to apply a Security Policy. Use the Select button to choose the desired SOAP resources or Service operations. Only the appropriate policy fields are available based on the selected resource. For example, when you select a SOAP Event Source, only the Inbound Message Policy field is available.
Inbound Message Policy	No	A Security Policy resource that specifies the security characteristics you want to apply to inbound messages.
Outbound Message Policy	No	A Security Policy resource that specifies the security characteristics you want to apply to outbound messages.
Inbound Fault Message Policy	No	A Security Policy resource that specifies the security characteristics you want to apply to inbound fault messages.
Outbound Fault Message Policy	No	A Security Policy resource that specifies the security characteristics you want to apply to outbound fault messages.

Inbound

The Inbound tab has the following fields.

Field	Global Var?	Description
Message Elements for	No	Specifies an XPath expression to select the elements within the SOAP message that are expected to have a signature. If

Field	Global Var?	Description
Signature		the incoming message does not have a signature for the selected elements, an exception is thrown.
Message Elements for Encryption	No	Specifies an XPath expression to select the elements within the SOAP message that are expected to be encrypted. If the elements selected in the incoming message are not encrypted, an exception is thrown.
Prefix Namespace Pair	No	Specify the namespace associated to the prefix used. Note: Only for elements that are not part of the schema. The table consists of two columns: • Prefix- Specify the prefix used. • Namespace- Select the namespace to be associated with the specified prefix.

Outbound

The Outbound tab has the following fields.

Field	Global Var?	Description
Message Elements for Signature	No	Specifies an XPath expression to select the elements of outgoing messages for which you want to include a signature, if a signature is specified in the selected Security Policy resource.
Message parts for Encryption	No	Specifies an XPath expression to select the elements of outgoing message parts that you want to encrypt, if confidentiality is specified in the selected Security Policy resource.

Field	Global Var?	Description
Prefix Namespace Pair	No	Specify the namespace associated to the prefix used. Note: Only for elements that are not part of the schema. The table consists of two columns: • Prefix- Specify the prefix used. • Namespace- Select the namespace to be associated with the specified prefix.

Support for Unqualified Elements

To use unqualified elements, complete one of the following tasks:

- Create a new Security Policy Association with the required fields.
- Edit the existing Security Policy Association by:
 - Removing the unqualified element.
 - Modifying the inbound/outbound field.
 - Save the project.
 - Add the unqualified element again.

Rendezvous Palette

The **Rendezvous** palette allows you to send and receive TIBCO Rendezvous messages. For more information about TIBCO Rendezvous, see the TIBCO Rendezvous documentation.

RawRVMsg2 Schema

ActiveMatrix BusinessWorks includes an XSD file that defines a generic message structure and some additional TIBCO Rendezvous message types. You can use this schema to send or receive a message with an arbitrary structure (raw TIBCO Rendezvous messages). This schema also describes the datatypes for fields that use the TIBRVMSG_XML, TIBRVMSG_IPADDR32, and TIBRVMSG_IPPORT16 TIBCO Rendezvous datatypes.

If you want to send or receive raw TIBCO Rendezvous messages, or you want to use the TIBRVMSG_XML, TIBRVMSG_IPADDR32, or TIBRVMSG_IPPORT16 datatypes in a message, you must include this schema in your project.

The RawRVMsg2 Schema

The following is the RawRVMsg2 schema:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:prefix0="http://xmlns.tibco.com/2003/5/bw/plugins/tibrv"
targetNamespace="http://xmlns.tibco.com/2003/5/bw/plugins/tibrv"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xsd:element name="msg">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="field" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="name" type="xsd:string"/></xsd:element>
              <xsd:element name="id" minOccurs="0" type="xsd:unsignedShort"/>
            </xsd:element>
          <xsd:choice>
```

```

        <xsd:element name="simple" type="xsd:anySimpleType">
        </xsd:element>
        <xsd:element ref="prefix0:msg"></xsd:element>
    </xsd:choice>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:simpleType name="ipPort">
    <xsd:restriction base="xsd:unsignedShort"/>
</xsd:simpleType>
<xsd:simpleType name="ipAddr">
    <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="tibrvXml">
    <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
</xsd:schema>

```

Adding the RawRVMMsg2 Schema to Your Project

The RawRVMMsg2.xsd file specifies the RawRVMMsg2 schema. To add this schema to your project, perform the following procedure:

Procedure

1. Locate the bw/5.12/lib/palettes/plugins.jar file in your ActiveMatrix BusinessWorks installation directory.
2. The RawRVMMsg2.xsd file is contained in the plugins.jar file. You can extract this file in a number of ways, depending upon your operating system. For example, on Windows, you can open up plugins.jar using WinZip and extract the RawRVMMsg2.xsd file. On UNIX, you can use Java's facilities to un-jar the file and copy the RawRVMMsg2.xsd file.
3. Once you have obtained the RawRVMMsg2.xsd file, copy this file into your ActiveMatrix BusinessWorks project directory. You can copy this file into the root project directory or to a subdirectory in your project.
4. Open your project, and the RawRVMMsg2 schema should be located in the folder where you placed it. If you already have your project open in TIBCO Designer, you

can choose **Resources > Refresh** from the menu to display the newly placed file in your project tree.

Datatype Conversion

When sending or receiving TIBCO Rendezvous messages, ActiveMatrix BusinessWorks converts the fields of the message to the correct datatypes. Data in incoming TIBCO Rendezvous messages is converted to ActiveMatrix BusinessWorks datatype (and represented in the activity's output). Data in an outgoing message is converted to TIBCO Rendezvous datatypes.

The following tables describe the corresponding datatypes between ActiveMatrix BusinessWorks and TIBCO Rendezvous, depending upon whether ActiveMatrix BusinessWorks is sending or receiving the message.

TIBCO Rendezvous to ActiveMatrix BusinessWorks datatype conversion

TIBCO Rendezvous Datatype	XSD Datatype
TIBRVMSG_BOOL	boolean
TIBRVMSG_I8	byte
TIBRVMSG_I16	short
TIBRVMSG_I32	int
TIBRVMSG_I64	long
TIBRVMSG_U8	unsignedByte
TIBRVMSG_U16	unsignedShort
TIBRVMSG_U32	unsignedInt
TIBRVMSG_U64	unsignedLong
TIBRVMSG_F32	float

TIBCO Rendezvous Datatype	XSD Datatype
TIBRVMSG_F64	double
TIBRVMSG_OPAQUE	hexBinary or base64Binary, depending upon what is defined in the schema.
TIBRVMSG_DATETIME	date
TIBRVMSG_STRING	string

ActiveMatrix BusinessWorks to TIBCO Rendezvous datatype conversion

XSD Datatype	TIBCO Rendezvous Datatype
boolean	TIBRVMSG_BOOL
byte	TIBRVMSG_I8
short	TIBRVMSG_I16
int	TIBRVMSG_I32
long	TIBRVMSG_I64
integer	TIBRVMSG_I64
unsignedByte	TIBRVMSG_U8
unsignedShort	TIBRVMSG_U16
unsignedInt	TIBRVMSG_U32
unsignedLong	TIBRVMSG_U64
float	TIBRVMSG_F32
double	TIBRVMSG_F64

XSD Datatype	TIBCO Rendezvous Datatype
hexBinary	TIBRVMSG_OPAQUE
base64Binary	TIBRVMSG_OPAQUE
date	TIBRVMSG_DATETIME
dateTime	TIBRVMSG_DATETIME
time	TIBRVMSG_STRING
string	TIBRVMSG_STRING

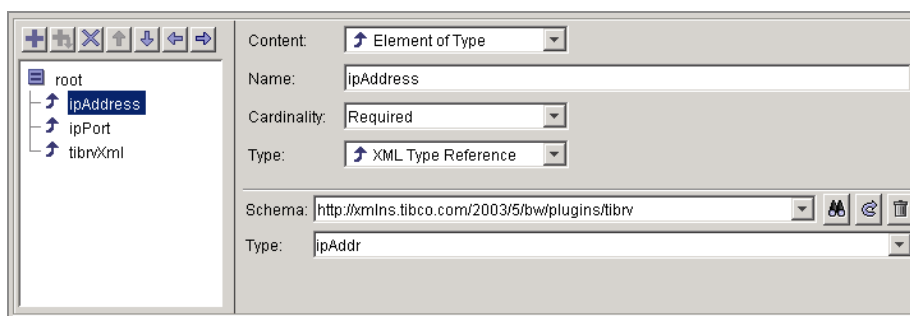
TIBRVMSG_XML, TIBRVMSG_IPADDR32, and TIBRVMSG_IPPORT16 Datatypes

If you want to use the TIBRVMSG_XML, TIBRVMSG_IPADDR32, or TIBRVMSG_IPPORT16 datatypes, you must first add the RawRVMsg2.xsd file into your project. For more information about adding this schema to your project, see [RawRVMsg2 Schema](#).

Once the RawRVMsg2 schema is in your project, you can reference the ipPort and ipAddr datatypes contained in that schema. These types correctly convert data of type TIBRVMSG_XML, TIBRVMSG_IPADDR32, or TIBRVMSG_IPPORT16 when sending or receiving TIBCO Rendezvous messages.

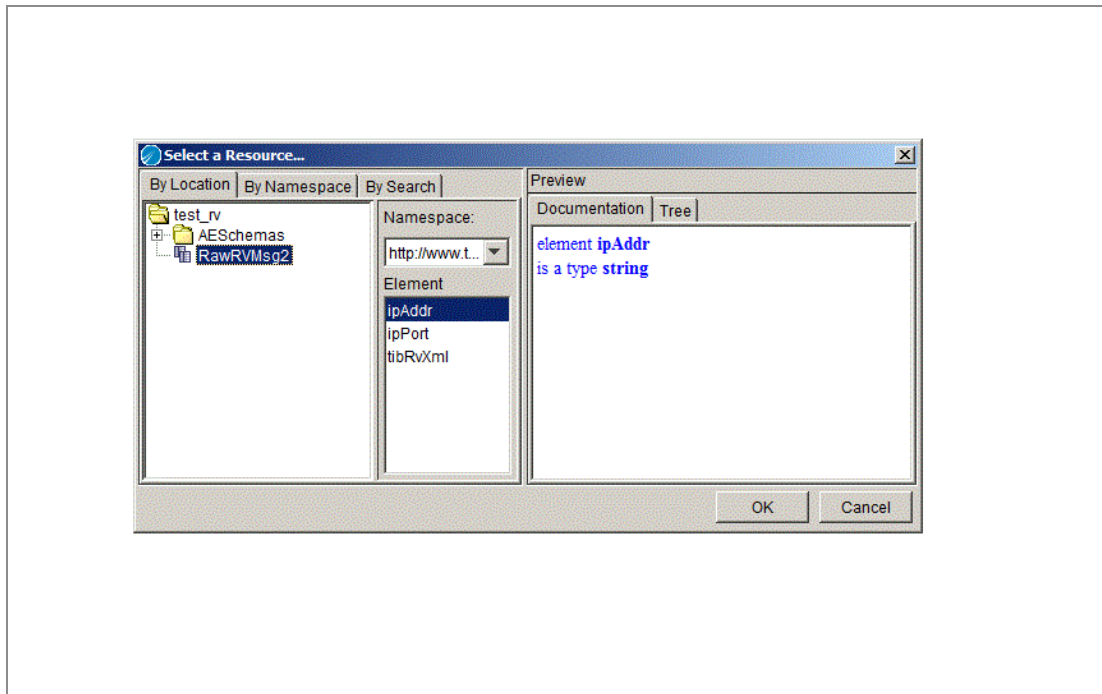
For example, you want to send a message with two fields, ipAddress, ipPort, and tibrvXml. [Using the ipPort and ipAddr datatypes](#) illustrates an Input Editor tab that defines the message structure.

Figure 21: Using the ipPort and ipAddr datatypes



In the content field, Element of Type is chosen. In the Type field, XML Type Reference is chosen to indicate you want to reference a datatype defined in a schema resource. Use the **Browse** button in the Schema field to locate the RawRVMsg2 schema and select the appropriate type for the field. [Selecting a type from the RawRVMsg2 schema](#) illustrates selecting the correct type from the RawRVMsg2 schema.

Figure 22: Selecting a type from the RawRVMsg2 schema



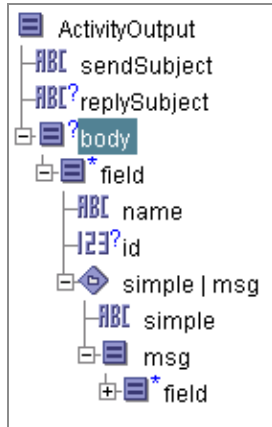
Raw TIBCO Rendezvous Message Support

In some situations, you may want to send or receive TIBCO Rendezvous messages without specifying the exact input or output schema. For example, an application may send messages with three different structures on the same subject name. You are interested in receiving all messages, but you only want to process messages when the field named "orderType" has a value of "incomingOrder". ActiveMatrix BusinessWorks supplies a generic TIBCO Rendezvous message schema for handling incoming messages without specifying their structure.

The RawRVMsg2 schema describes a generic message structure. The structure contains a root element named msg. In the msg element is a repeatable element named field. Each field contains a name, id, and a choice element. The choice element contains the data of the field. A field's data can be either a scalar value or another message.

[Schema for RawRVMsg2.xsd](#) illustrates the generic TIBCO Rendezvous message schema used in a Rendezvous Subscriber process starter.

Figure 23: Schema for RawRVMsg2.xsd



Before using the RawRVMsg2 schema, you must add the schema to your project. For more information about adding this schema to your project, see [RawRVMsg2 Schema](#).

Using the RawRVMsg2 Schema

Once you have the RawRVMsg2 schema in your project, you can reference it in any Input Editor or Output Editor tab on any activity in the TIBCO **Rendezvous** palette. To use the RawRVMsg2 schema, perform the following;

Procedure

1. Create a root element in the Input/Output Editor tab with the **Add Child** button.

Note: You can only use the msg element of the RawRVMsg2 schema on the root element in the schema you create in the Input or Output Editor tab. Do not attempt to use this schema for any element that is not the root of the schema.

2. Select XML Element Reference in the Content field on the right.
3. Use the **Browse Resources** button in the Schema field to locate and select the RawRVMsg2 schema.

Once the schema is selected, it becomes the schema for incoming or outgoing message bodies.

You can use XPath expressions to extract the desired field from the generic RawRvMsg2 schema. For example, your process begins with a **Rendezvous Subscriber** activity and you want to perform processing on incoming messages that have a field named "orderType" with a value of "incomingOrder", you would use the following XPath expression to determine which messages meet that criteria:

```
$Rendezvous-Subscriber/ActivityOutput/body/pfx:field/pfx:name  
= "orderType"  
and $Rendezvous-Subscriber/ActivityOutput/body/pfx:field/pfx:simple  
= "incomingOrder"
```

For more information about XPath, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Field Names In Messages

A TIBCO Rendezvous message can contain field names that are not compliant with XML or ActiveEnterprise naming rules. For example, XML element names cannot begin with a number, nor can they contain special characters, such as # or ^. When activities in the TIBCO **Rendezvous** palette send or receive messages, element names in the input or output schemas for the activity must comply with XML and ActiveEnterprise naming rules. Any elements in referenced schemas in the input or output schemas are automatically altered so that they comply with XML and ActiveEnterprise naming rules. The XML-Compliant Field Names field allows you to handle incoming or outgoing messages when the field names are not valid XML or ActiveEnterprise names.

When the XML-Compliant Field Names field is unchecked, any field names in incoming or outgoing messages that do not comply with XML or ActiveEnterprise naming rules are altered so that they comply with the rules. When the field is checked, field names in the message are left unaltered.

When publishing a message, unchecking the XML-Compliant Field names field ensures that field names in the message sent over the transport to the receiving application are altered when necessary to comply with XML or ActiveEnterprise naming rules. Checking the field allows you to send the message with the original field names intact; no alterations are made to the field names. This is useful in the case where you reference a schema in the Input Editor that contains invalid XML or ActiveEnterprise names (such as a TIBCO ActiveEnterprise schema that contains special characters). The receiving application receives the message with the original field names, if the XML-Compliant Field Names field is unchecked.

When receiving a message, unchecking the XML-Compliant Field Names field ensures that field names in the incoming message are altered to comply with XML and ActiveEnterprise naming rules. If the message contains fields that do not comply to XML and ActiveEnterprise naming rules, the output schema of the subscriber activity must specify the correct altered name in order to match the altered name of the field in the incoming message. For more information about how to convert a TIBCO ActiveEnterprise schema into a valid XSD, see the description of the *ae2xsd* utility in the *TIBCO Runtime Agent™ Administrator*. Using this utility allows you to see how field names get altered to comply with XML and ActiveEnterprise naming rules. Unchecking the field leaves the field names of the message intact.



Note: Unchecking the XML-Compliant Field Names field involves extra processing of incoming and outgoing messages. If you are certain that messages contain only field names that comply with XML naming rules, you can uncheck this field to improve performance.

Internationalization Support

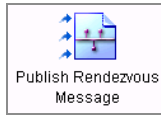
TIBCO Rendezvous activities use the TIBCO Messaging Encoding as the wire format encoding when testing processes in TIBCO Designer and when the project is running as a legacy local-file based project. The default TIBCO Messaging Encoding value is ISO8859-1, which only supports English and other western European languages that belong to ISO Latin-1 character set.

You must set the TIBCO Messaging Encoding to UTF-8 if you want to send and receive characters not in the ASCII and Latin-1 character set. When creating a new project, you can set the TIBCO Messaging Encoding in the Save Project dialog or on the Project Settings tab. For more information, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

After the project is deployed in a TIBCO Administrator domain, the messaging encoding set at design time is overridden by the TIBCO Administrator domain's encoding property. All the TIBCO components working in the same domain must always use the same encoding for intercommunication. For more information, see *TIBCO Administrator Server Configuration Guide*.

Publish Rendezvous Message

Activity



The **Publish Rendezvous Message** activity publishes a TIBCO Rendezvous message on the given subject with the given message content.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Subject	Yes	The subject on which to publish the TIBCO Rendezvous message. You can override this value by specifying a subject on the Input tab.
Transport	No	The TIBCO Rendezvous transport parameters. These are specified as a Rendezvous Transport shared configuration resource. For more information about shared configuration resources, see Rendezvous Transport .
Pre-register Listener	Yes	When using TIBCO Rendezvous Certified Messaging (RVCM), this field specifies the cmname(s) of the RVCM subscriber(s). This allows the publisher to pre-register with any expected listeners. If multiple cmnames are specified, separate each cmname with a comma. For more information about RVCM, see the TIBCO Rendezvous documentation.
XML Format	No	If checked, the body of the TIBCO Rendezvous message is placed into a single field named "xml". This field is a byte array

Field	Global Var?	Description
		containing the body of the message. The datatype of the field is TIBRVMSG_XML.
XML-Compliant Field Names	No	Specifies whether the field names of the outgoing message should be altered so that they comply with XML and ActiveEnterprise naming rules. Only field names that do not comply with XML and ActiveEnterprise naming rules are altered when this field is unchecked. When checked, field names are left unaltered.

Input Editor

The Input Editor tab allows you to define a custom schema for the body of the TIBCO Rendezvous message. For a complete description of using the Input Editor tab, see [Specifying Data Schema](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
subject	string	The subject of the TIBCO Rendezvous message.
replySubject	string	The subject to send replies for this TIBCO Rendezvous message. This is useful if you want to use the Wait for Rendezvous Message activity later in the process to receive replies to the message.
preRegisterListener	string	When using TIBCO Rendezvous Certified

Input Item	Datatype	Description
		Messaging (RVCM), this field specifies the cmname(s) of the RVCM subscriber(s). This allows the publisher to pre-register with any expected listeners. If multiple cmnames are specified, separate each cmname with a comma. Any listeners preregistered on the Transport tab are also preregistered with the listeners specified in this element. For more information about RVCM, see the TIBCO Rendezvous documentation.
body	complex	The body of the TIBCO Rendezvous message.

Output

This activity produces no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
TransportException	The transport cannot be created.
SerializationException	The message cannot be serialized.
RVPluginException	An error occurred when sending the message.

Rendezvous Subscriber

Process Starter



The Rendezvous Subscriber process starter creates a process when a TIBCO Rendezvous message on the given subject is received.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Subject	Yes	The subject on which to listen for the TIBCO Rendezvous message.
Transport	Yes	The TIBCO Rendezvous transport parameters. These are specified as a Rendezvous Transport shared configuration resource. For more information about shared configuration resources, see Rendezvous Transport .
XML Format	No	If checked, the body of the incoming TIBCO Rendezvous message is expected to be a single field named "xml". This field is a byte array containing the body of the message. ActiveMatrix BusinessWorks automatically deserializes the xml field and represents the message as a schema tree in this activity's output.
Needs Output Validation	No	Specifies that you would like the incoming message to be validated against the schema specified on Output Editor tab. An error is returned if an incoming message does not match the specified schema.

Field	Global Var?	Description
		Validation does affect system performance, and you may want to uncheck this field to handle incoming messages more quickly. If you uncheck this field, no validation is performed and any incoming message that does not match the specified schema is processed. You should only uncheck this field if you can guarantee that incoming messages conform to the expected schema.
Needs Output Filtration	No	Specifies that only fields that match the name and datatype of the schema elements specified on the Output Editor tab should be processed. Any elements that do not match the specified schema are ignored and not included in this activity's output. This is useful if you need only a subset of the fields of an incoming message. Any unnecessary fields are not included in the activity output. This can save memory, and consequently improve performance. Note: This option is only valid when the XML Format field is unchecked. Do not select this check box when the XML Format field is selected.
XML-Compliant Field Names	No	Specifies whether the field names of the incoming message should be altered so that they comply with XML and ActiveEnterprise naming rules. Only field names that do not comply with XML and ActiveEnterprise naming rules are altered when this field is unchecked. When checked, field names are left unaltered.
Raw-RV-Object Mode	No	Checking this field disables the XML Format, Needs Output Validation, Needs Output Filtration, and XML-Compliant Field Names fields. Specifies that the body of the RV message should be output as a Java object reference. A subsequent Java Code activity

Field	Global Var?	Description
		in the process definition can accept the reference as an input parameter and then construct the RV message from the object. To use the Java object, the Java code activity must cast the object reference as a byte array. For example: <pre>TibrvMsg tibrvMsg = new TibrvMsg ((byte[])in_var_1);</pre> Where in_var_1 is the input parameter for the Java Code activity that is mapped to the object reference of the object containing the RV message.

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output Editor

The Output Editor tab allows you to define a custom schema for the body of the TIBCO Rendezvous message. For a complete description of using the Output Editor tab, see [Specifying Data Schema](#).

Output

The output for the activity is the following.

Output Item	Datatype	Description
sendSubject	string	The subject of the received message.
replySubject	string	The reply subject of the TIBCO Rendezvous message.
certifiedSender	string	This element is only available when RVCM transports are used. This element contains the CM name of the message sender.
certifiedSequenceNumber	integer	This element is only available when RVCM transports are used. This element contains the sequence number of the message.
body	complex	The body of the TIBCO Rendezvous message. If the Raw-RV-Object Mode field is checked on the Configuration tab, the body is output as an object reference.

Rendezvous Transport

Shared Configuration



The Rendezvous Transport resource describes a TIBCO Rendezvous transport. This resource is used when specifying activities from the Rendezvous palette.

For more information about specifying these fields, see the TIBCO Rendezvous documentation.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.
Daemon	Yes	In the case of TIBCO Rendezvous daemon running on the same machine as ActiveMatrix BusinessWorks process engine, this is not specified. If Rendezvous is running on a different machine, then the Daemon field is specified as the remote host name followed by the socket number. For example: <pre>ssl:acct:5785</pre>
Network	Yes	This field contains the host name, IP address, network name, or interface name. For example: <pre>;224.34.103.4</pre>
Service	Yes	This is the TIBCO Rendezvous service name in one of the following formats: <pre><service name></pre> or <pre><port number></pre>

Field	Global Var?	Description
SSL	No	The Use SSL check box specifies that Secure Sockets Layer (SSL) should be used when communicating with the TIBCO Rendezvous daemon. When this field is checked, the Configure SSL button is enabled. For more information about configuring SSL parameters, see Configure SSL Button .

Configure SSL Button

The **Configure SSL** button allows you to configure the SSL parameters for communicating with the TIBCO Rendezvous daemon. For more information about how SSL is configured for TIBCO Rendezvous daemons and clients, see the TIBCO Rendezvous documentation.

The SSL Configuration for TIBCO RV dialog has the following fields:

Field	Description
Daemon Certificate	File containing one or more certificates from trusted certificate authorities. This file is checked when connecting to a daemon to ensure that the connection is to a daemon that is trusted. This prevents connections to rogue TIBCO Rendezvous daemons that attempt to impersonate trusted daemons. You can retrieve a daemon's certificate using the administration interface in TIBCO Rendezvous. For more information about obtaining certificates through the administration interface, see the TIBCO Rendezvous documentation. Once retrieved, you can select a folder in your project and import this certificate into the folder using the Tools>Trusted Certificates>Import Into PEM Format menu item. ActiveMatrix BusinessWorks supports the use of global variables to configure certificates at deployment time. You can declare a global variable and set the absolute path to a Certificate in PEM format. This eliminates the need to embed certificates in ActiveMatrix BusinessWorks projects.
Identity	This is an Identity resource used to authenticate to the TIBCO Rendezvous

Field	Description
	daemon. The Browse button allows you to select from a list of appropriately configured Identity resources. Only Identity resources whose Type field is set to Identity File or Username/Password are listed. For more information, see <i>TIBCO Designer™ Palette Reference</i>.

Advanced

The Advanced tab has the following fields.

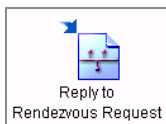
Field	Global Var?	Description
RV Type	No	The type of TIBCO Rendezvous connection to use. This can be reliable (standard RV transport), certified (RVCM), or Distributed Queue (RVCMQ). Note: Use Confirm activity in the process when using RV Certified messaging. The fields of the Advanced tab correspond to the value selected for this field.
Certified Transport		
CM Name	Yes	The name of the delivery-tracking session. This name is in the same format as TIBCO Rendezvous subject names.
Ledger File	Yes	The name and location of the persistent ledger file that tracks certified messages. If not specified, the certified message ledger is kept in process memory only.
Sync Ledger File	Yes	Specifies whether to keep the ledger file synchronous with the current messages.
Relay Agent	Yes	Name of the relay agent to use. Relay agents are useful when

Field	Global Var?	Description
		clients are disconnected from the network from time to time. The relay agents store inbound certified messages and labeled messages (and other messages related to certified delivery features) on behalf of their disconnected client programs. When a client is connected, it receives inbound messages immediately.
Require Old Message	Yes	Check this box if you want to require the retention of messages for which delivery has not been confirmed. These messages are resent.
Message Timeout (sec)	Yes	The time limit (in seconds) for certified message delivery.
Distributed Queue Transport		
CMQ Name	Yes	The name of the distributed queue. This name is in the same format as TIBCO Rendezvous subject names
Worker Weight	Yes	The weight of the worker (this pertains to the worker processing queue requests, not to ActiveMatrix BusinessWorks process engines). Relative worker weights assist the scheduler in assigning tasks. When the scheduler receives a task, it assigns the task to the available worker with the greatest worker weight.
Worker Tasks	Yes	Sets the task capacity for the worker (this pertains to the worker processing queue requests, not to ActiveMatrix BusinessWorks process engines). Task capacity is the maximum number of tasks that a worker can accept. When the number of accepted tasks reaches this maximum, the worker cannot accept additional tasks until it completes one or more of them.
Worker Complete Time	Yes	The amount of time the scheduler waits for a worker process to complete. If the worker process does not complete in the specified period, the scheduler reassigns the message to

Field	Global Var?	Description
		another worker.
Scheduler Weight	Yes	Weight represents the ability of this member to fulfill the role of scheduler, relative to other members with the same name. Cooperating distributed queue transports use relative scheduler weight values to elect one transport as the scheduler; members with higher scheduler weight take precedence. Acceptable values range from 1 to 65535.
Scheduler Heartbeat	Yes	The scheduler sends heartbeat messages at this interval (in seconds). All members with the same name must specify the same value for this parameter. The value must be strictly positive.
Scheduler Activation	Yes	When the heartbeat signal from the scheduler has been silent for this interval (in seconds), the member with the greatest scheduler weight takes its place as the new scheduler. All members with the same name must specify the same value for this parameter. The value must be positive

Reply to Rendezvous Request

Activity



The **Reply to Rendezvous Request** activity is used to send a reply to a received TIBCO Rendezvous message.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Reply For	No	The TIBCO Rendezvous activity or process starter that received the request. This is a selection list of available activities that can receive TIBCO Rendezvous messages.
Reply Subject	Yes	The reply subject of the received TIBCO Rendezvous message. You can override this value by specifying a subject on the Input tab.

Input Editor

The Input Editor tab allows you to define a custom schema for the body of the TIBCO Rendezvous message. For a complete description of using the Input Editor tab, see [Specifying Data Schema](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
replySubject	string	The reply subject of the TIBCO Rendezvous message.
body	complex	The body of the TIBCO Rendezvous message.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
RVPluginException	An error occurred when sending the reply message.

Send Rendezvous Request

Activity



The **Send Rendezvous Request** activity publishes a TIBCO Rendezvous message on the given subject with the given message content, and it expects a reply to the message on the given reply subject and with the given reply message content. This activity waits for a reply on the reply subject and outputs the content of the reply.

Note: This activity uses the INBOX mechanism to ensure that reply messages are received only by the process that sent the request.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.

Field	Global Var?	Description
Description	No	Short description of the activity.
Subject	Yes	The subject on which to publish the TIBCO Rendezvous message. You can override this value by specifying a subject on the Input tab.
Transport	No	The TIBCO Rendezvous transport parameters. These are specified as a Rendezvous Transport shared configuration resource. For more information about shared configuration resources, see Rendezvous Transport .
Request Timeout (msec)	Yes	The amount of time to wait (in milliseconds) for a reply to the TIBCO Rendezvous message. If no reply is received in the given time limit, an error is returned.
Pre-register Listener	Yes	When using TIBCO Rendezvous Certified Messaging (RVCM), this field specifies the cmname(s) of the RVCM subscriber(s). This allows the sender to pre-register with any expected listeners. If multiple cmnames are specified, separate each cmname with a comma. For more information about RVCM, see the TIBCO Rendezvous documentation.
XML Format	No	If checked, the bodies of the TIBCO Rendezvous request and reply messages are sent as a single field named "xml". This field is a byte array containing the body of the message. ActiveMatrix BusinessWorks automatically deserializes the reply message and represents the data of the message as a schema tree based on the schema specified in the Output Editor tab.
Needs Output Validation	No	Specifies that you would like the reply message to be validated against the schema specified on Output Editor tab. An error is returned if a reply message does not match the specified schema.

Field	Global Var?	Description
		Validation does affect system performance, and you may want to uncheck this field to handle reply messages more quickly. If you uncheck this field, no validation is performed and any reply message that does not match the specified schema is processed. You should only uncheck this field if you can guarantee that reply messages conform to the expected schema.
Needs Output Filtration	No	Specifies that only fields in the reply message that match the name and datatype of the schema elements specified on the Output Editor tab should be processed. Any elements that do not match the specified schema are ignored and not included in this activity's output. This is useful if you need only a subset of the fields of a reply message. Any unnecessary fields are not included in the activity output. This can save memory, and consequently improve performance. Note: This option is only valid when the XML Format field is cleared. Do not select this check box when the XML Format field is also selected.
XML-Compliant Field Names	No	Specifies whether the field names of the outgoing message should be altered so that they comply with XML and ActiveEnterprise naming rules. Only field names that do not comply with XML and ActiveEnterprise naming rules are altered when this field is unchecked. When checked, field names are left unaltered.

Input/Output Editor

The Input Editor and Output Editor tabs allow you to define custom schemas for the body of the outgoing TIBCO Rendezvous request and body of the incoming TIBCO Rendezvous response. For a complete description of using the Input/Output Editor tab, see [Specifying Data Schema](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
subject	string	The subject of the TIBCO Rendezvous message.
body	complex	The body of the TIBCO Rendezvous message.
replySubject	string	The reply subject for this activity. Note: The reply messages may not be received by the correct job if more than one job has the same reply subject (for example, if a constant is specified in this field). Use the Publish Rendezvous Message and Wait for Rendezvous message instead if you want to receive replies on a single subject.
timeout	number	The amount of time to wait (in milliseconds) for a reply to the TIBCO Rendezvous message. If no reply is received in the given time limit, an error is returned.

Output

The output for the activity is the following.

Output Item	Datatype	Description
replyMessageReplySubject	string	A reply message can optionally have a reply subject, presumably so that the two applications can communicate with further messages. This element contains the reply subject set on the reply message, if one is set. If no reply subject is set on the reply

Output Item	Datatype	Description
		message, this element is not populated.
body	complex	The body of the TIBCO Rendezvous reply message.

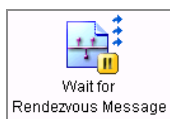
Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
TransportException	The transport cannot be created.
SerializationException	The message cannot be serialized.
RVPluginException	An error occurred when sending the message.
ActivityTimeoutException	A timeout has been reached.

Wait for Rendezvous Message

Activity



The **Wait for Rendezvous Message** activity waits to receive a TIBCO Rendezvous message with the given subject.

For information on running the "Wait for" activities in a multi-engine mode, see [Scalability With Incoming Events](#) in *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Subject	Yes	The subject on which to listen for the TIBCO Rendezvous message.
Transport	Yes	The TIBCO Rendezvous transport parameters. These are specified as a Rendezvous Transport shared configuration resource. For more information about shared configuration resources, see Rendezvous Transport .
XML Format	No	If checked, the body of the incoming TIBCO Rendezvous message is expected to be a single field named "xml". This field is a byte array containing the body of the message. ActiveMatrix BusinessWorks automatically deserializes the xml field and represents the message as a schema tree in this activity's output.
Needs Output Validation	No	Specifies that you would like the incoming message to be validated against the schema specified on Output Editor tab. An error is returned if an incoming message does not match the specified schema. Validation does affect system performance, and you may want to uncheck this field to handle incoming messages more quickly. If you uncheck this field, no validation is performed and any incoming message that does not match the specified schema is processed. You should only uncheck this field if you can guarantee that incoming messages conform to the expected schema.

Field	Global Var?	Description
Needs Output Filtration	No	Specifies that only fields that match the name and datatype of the schema elements specified on the Output Editor tab should be processed. Any elements that do not match the specified schema are ignored and not included in this activity's output. This is useful if you need only a subset of the fields of an incoming message. Any unnecessary fields are not included in the activity output. This can save memory, and consequently improve performance. Note: This option is only valid when the XML Format field is unchecked. Do not check this checkbox when the XML Format field is also checked.
XML-Compliant Field Names	No	Specifies whether the field names of the incoming message should be altered so that they comply with XML and ActiveEnterprise naming rules. Only field names that do not comply with XML and ActiveEnterprise naming rules are altered when this field is unchecked. When checked, field names are left unaltered.

Event

The Event tab has the following fields.

Field	Description
Candidate Event Key	Expression built from the data of the incoming message. This expression should evaluate to a string and it is compared to the "key" field of the activity's input. If the Candidate Event Key and the activity's key match, then the process accepts the incoming message. For example, you may have a Publish Rendezvous Message activity that sends a message with a particular ID. You are expecting a reply message

Field	Description
	that contains that same ID so that you can determine the message is a response to your sent message. You would specify the field of the incoming message that contains your ID in the Candidate Event Key. You would then use the message ID of the message you sent earlier in the process as the "key" field in the input. This expression is specified in XPath, and only data from the incoming event is available for use in this XPath expression. For more information about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Event Timeout (msec)	A message may arrive before this activity is executed. This field specifies the amount of time (in milliseconds) a message waits if it is received before this activity is executed. If the event timeout expires, an error is logged and the event is discarded. If no value is specified in this field, the message waits indefinitely. If zero is specified, the event is discarded immediately, unless this activity has already been reached.



Note: When building an expression in the Candidate Event Key field, only data from the incoming event is available. This is because you want to place an expression containing incoming event data in the Candidate Event Key field. When the results of this expression match the results of the expression in the "key" item on the Input tab, the Wait For **Rendezvous Message** activity proceeds.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
key	string	The value that is compared to the Candidate Event Key field of the Event tab to determine if this message applies to this process. For an example of using the Candidate Event Key field with the key input item, see Event .
processTimeout	integer	The time (in milliseconds) to wait for the incoming message. An error is returned if the message is not received in this time limit.

Output Editor

The Output Editor tab defines the schema to use for incoming messages whose message type is Map, Stream, or XML Text. Map messages are name/value pairs, and the schema allows you to define the structure of the incoming message. Once defined, the schema on the Output Editor tab becomes the structure used for the body of the message displayed on the Output tab.

For a complete description of using the Output Editor tab, see [Specifying Data Schema](#).

Output

The output for the activity is the following.

Output Item	Datatype	Description
sendSubject	string	The subject of the received message.
replySubject	string	The reply subject of the TIBCO Rendezvous message.
certifiedSender	string	This element is only available when RVCM transports are used. This element contains the CM name of the message sender.

Output Item	Datatype	Description
certifiedSequenceNumber	integer	This element is only available when RVCM transports are used. This element contains the sequence number of the message.
body	complex	The body of the TIBCO Rendezvous message. The elements contained in this element are specified on the Configuration tab.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ActivityTimedOutException	A timeout has been reached.

RMI Palette

Remote Method Invocation (RMI) is a protocol for calling operations on remote Java objects. ActiveMatrix BusinessWorks can act as server for incoming RMI calls, or ActiveMatrix BusinessWorks can use RMI to retrieve a remote object. For client interactions, either a remote Java object on an RMI server or an Enterprise Java Bean (EJB) can be retrieved. For server interactions, a client can make an RMI call to ActiveMatrix BusinessWorks that initiates a process instance.

This section assumes that you are familiar with the RMI protocol. For more information, see: <http://java.sun.com/docs/books/tutorial/rmi/index.html>.

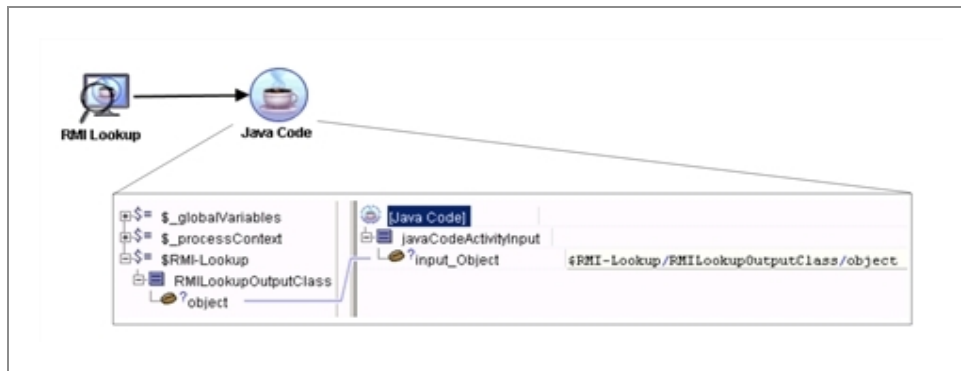
RMI Overview

You can use the resources in the **RMI** palette to either retrieve a remote Java object or to create a process definition that starts a process instance for each incoming RMI call. This section describes the uses of the resources in the **RMI** palette.

Using RMI to Call a Remote Object

The **RMI Lookup** activity allows you to use a registry server to retrieve a Java object that implements a client stub of a remote object. You can then pass the object to a **Java Code** or **Java Method** activity and invoke methods on the object. [Using the RMI Lookup activity](#) illustrates using the **RMI Lookup** activity to obtain a client stub for a remote Java object and then mapping that object to an input parameter for the **Java Code** activity.

Figure 24: Using the RMI Lookup activity



The object obtained by the lookup activity is an instance of the server's stub class for the actual server object interface. To call methods on this object with a **Java Code** activity, use the stub class interface to cast the object before making the calls. Similarly, the **Java Method** activity should specify the stub class interface in its Class field. The generated client stub's .jar file must be either in the ActiveMatrix BusinessWorks classpath or in an AliasLibrary resource before it can be used by the **Java Method** activity.

Acting as a RMI Server

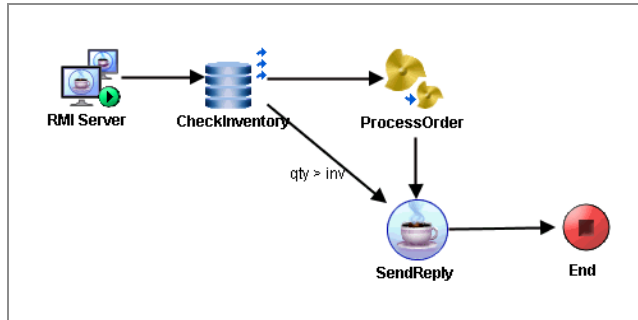
The [RMI Server](#) process starter registers the specified object name with the specified registry server. Then, requests from RMI clients for the object are directed to the RMI Server process starter. A new process instance is created to handle each incoming request.

The RMI Server process starter can output two Java objects, one for the request and one for the reply. Either the request or reply objects can be nil. Optionally, you can also specify an output schema in the Output Editor tab of the RMI Server process starter. This allows you to specify fields that are expected in the request object and the data from the request object is used to populate the desired output schema. You can then process the request based on the specified output schema, and you can also access the actual request object by passing it to a Java Code or [Java Method](#) activity.

When the client is expecting a reply, you can use the Java Code or Java Method activities to invoke the appropriate methods on the reply object.

[Using the RMI Server process starter](#) illustrates using the RMI Server process starter.

Figure 25: Using the RMI Server process starter



RMI Lookup

Activity



The **RMI Lookup** activity retrieves a reference to a remote object from the specified registry server. The object reference output by this activity can be passed to a [Java Code](#) or [Java Method](#) activity to invoke methods on the remote object.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Alias Library	No	Use this field to specify an AliasLibrary resource that is used to locate the Java class files. For more information about AliasLibrary resources, see <i>TIBCO Designer™ Palette Reference</i> .

Field	Global Var?	Description
Remote Object Name	Yes	Name of the remote object you want to retrieve.
Registry Host Name	Yes	Host name of the registry server. The registry server provides the remote object reference for making the call to the remote object.
Registry Server Port	Yes	Port number for the registry server.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
remoteObjectName	string	Name of the remote object whose client stub you want to retrieve. The value specified in this element overrides any value specified in the Remote Object Name field on the Configuration tab.

Output

The output for the activity is the following.

Output Item	Datatype	Description
object	Java Object	An object reference for the specified remote Java object.

RMI Server

Process Starter



The RMI Server process starter registers the specified remote object name with the specified registry server and then creates process instances to handle incoming requests for the object. The process definition acts as the implementation of the specified object.

You can specify an output schema on the Output editor tab to extract expected fields from the incoming request object. The data in the output schema is then available to subsequent activities in the process definition. The request and reply Java objects are also output and available to activities that can accept Java objects as input (for example, [Java Code](#) or [Java Method](#)). You can send a reply to the client by invoking the appropriate methods on the reply object, if the client requires a reply.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Alias Library	No	Use this field to specify an AliasLibrary resource that is used to locate the Java class files. For more information about AliasLibrary resources, see <i>TIBCO Designer™ Palette Reference</i> .
Remote Object Name	Yes	Name of the Java object that this process definition implements.
Registry Host Name	Yes	Host name of the registry server with which you want to register the specified object.

Field	Global Var?	Description
Registry Server Port	Yes	Port number for the registry server.

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output Editor

The Output Editor tab allows you to define a custom schema for expected fields in the request object. For a complete description of using the Output Editor tab, see [Specifying Data Schema](#).

Output

The output for the activity is the following.

Output Item	Datatype	Description
request	Java Object	An object reference to the request object sent in the RMI request. The client might not send any data in the request, and in that case, this element remains nil.
reply	Java Object	An object reference to the reply object required by the client. The client might not require a reply, and in that case, this element remains nil.
<data>	varies	The output schema specified on the Output Editor tab. Data from the request object is extracted into the elements of this output schema. This allows you to easily access data from the request object without passing the object to a Java Code or Java to XML activity.

Service Palette

The **Service** palette contains resources for implementing Services. Services implement a set of interfaces and expose one or more endpoints. Services provide an abstraction so that you can implement a more generic service-oriented architecture (SOA). This allows you to decouple a service from its underlying transport and implementation.

For more information about services, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Context Resource

Shared Configuration



The Context Resource shared configuration resource allows you to specify a schema to hold context data from an incoming request or outgoing replies to a service. This allows you to pass on data, such as the username and password sent by the requesting application, to operation implementations without requiring the implementations to be aware of which transport the request was bound to. Also, you can pass data from the process that implements an operation back to the Service for any outgoing reply message. The Context Resource can also be used to store SOAP headers or message attachments.

For example, your operation implementation may require the user ID of the requestor or the public certificate of the user making the request. Also, you may want to set outgoing reply header values based on data from the process that implements the operation.

For more information about context separation for Service resources and operation implementations, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the shared resource.

Schema

The Schema tab allows you to define a custom schema to hold the data of the context. You can define your own datatype on this tab, and you can reference an XML schema stored in the project. For more information about how to define a schema, see [Specifying Data Schema](#).

Get Context

Activity



The **Get Context** activity retrieves the value of the specified [Context Resource](#). This is useful if your process definition requires some context information from an incoming request.

For more information about context separation for Service resources and operation implementations, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.

Field	Global Var?	Description
Context Configuration	No	The Context Resource whose value you would like to retrieve.

Output

The output for the activity is the following.

Output Item	Datatype	Description
<schema>	varies	The value of the context resource specified in the Context Configuration field of the Configuration tab is the output for this activity. The schema for the output is determined by the schema specified for the context resource.

Invoke Partner

Activity



The **Invoke Partner** activity can be used in a ActiveMatrix BusinessWorks Process Definition to invoke external services over SOAP.

For more information about services, partners and invoking partner services, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields

Field	Global Var?	Description
Name	No	The name to appear as the label for the invoke partner activity in the process definition.
Description	No	Short description of the activity.
Partner	No	The partner whose service is to be invoked by the activity. You can select the partner from the drop down menu that lists all the available partners in the process.
Operation	No	Select the operation that is to be invoked from the drop down menu. The drop down menu lists all the available operations for the selected partner.

Input

The input for the invoke partner service has the following fields

Input Item	Datatype	Description
input message	complex	The input message of the operation. This element contains all the input message parts for the operation.

Output

The output for the invoke partner service has the following fields

Output Item	Datatype	Description
output message	complex	The output message for the operation. This element contains all the output message parts for the operation.

Error Output

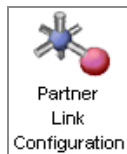
The Error Output tab lists the possible SOAP Fault messages that can be thrown by the invoked operation on the selected external service. These SOAP Fault messages are defined in the abstract WSDL. If no SOAP fault messages are defined in the abstract WSDL, there are no exceptions or fault messages listed in the error output tab.

For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

i Note: Unlike the **SOAP Request Reply** activity where you can define dynamic endpoint URL, you cannot define dynamic endpoint URL for the **invoke partner** activity.

Partner Link Configuration

Shared Resource



The Partner Link Configuration associates abstract partner portTypes with concrete port bindings. This allows you to easily link to new partner services without changing processes that invoke the partners.

For more information about partners and Partner Link Configuration resources, see [Partners](#).

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the resource.

Field	Global Var?	Description
Partner Links		Use the + button to add partner links to the list, use the X button to delete partner links from the list, use the arrow buttons to move partner links up and down in the list. Specify the following in the columns of the table for each partner: • Name : Double-click on this field to specify a name for the partner. • Timeout : Double-click on this field to specify a timeout in milliseconds (msec) for invocations of this partner. By default, the timeout is zero (0) which provides an unlimited amount of time for invocations. • Service Endpoint – The endpoint of the service. You can use concrete WSDL resources with port bindings or you can specify local service resources with Local specified in the endpoint binding. • Endpoint Type – By default the field selection is SOAP. It can be used when we check enable for AMX check box and change to AMX endpoint. This endpoint of the service populated from the concrete WSDL.
	No	
	Yes	
	No	
	No	
Attachment Style	No	This field allows you to choose the attachment style for the outbound/response messages. Two attachment styles are available: SwA and MTOM.
Operations	No	The operation that this process definition implements. The drop down list in this field is populated with the operations contained in the specified port type in the

Field	Global Var?	Description
		specified WSDL. You can choose one operation to implement per process definition. If you want to implement multiple operations, you must create a process definition for each.
SOAP Action	Yes	The soapAction that is expected from incoming SOAP requests. This field is required, and by default, the operation name is used as the soapAction. Each operation in an interface must have a unique value for SOAP Action. For more information about soapAction, see the SOAP specification. Note: When Partner Link Configuration is created, these fields get populated with the values from concrete WSDL. You can override these values in Partner Link Configuration. If the concrete WSDL is changed after configuring Partner Link Configuration, the changes are not visible in the Partner Link Configuration.

Additional Configuration for HTTP Endpoints

EndPoint URI	No	It has the endpoint of the service populated from the concrete WSDL. Note: When Partner Link Configuration is created, this field get populated with the values from concrete WSDL. You can override these values in Partner Link Configuration. If the concrete WSDL is changed after configuring Partner Link Configuration , the changes will are not visible in the Partner Link Configuration.
Use HTTP Proxy	No	Specifies to use a proxy server to gain access outside of a firewall.

Field	Global Var?	Description
Proxy Name	No	Host name or IP address of the proxy server.
HTTP Authentication	No	Specifies what authentication should be used. An authentication can be of the following three types: • NONE – is default. • BASIC – to continue with the current implementation of using username and password credentials. The Identity field appears when this field is enabled. • NTLM – Identity and Domain fields appear when NTLM is selected. Identity is a reference of the Username and Password combination.
Identity	No	When Basic and NTLM authentication (username and password) is required, provide an Identity resource in this field that contains the appropriate username and password. For more information, see <i>TIBCO Designer™ Palette Reference</i>.
Domain	No	Specifies the domain to be used for Authentication. The user name and password specified in the Identity field must exist in the domain.
SSL	No	When SSL is required for authentication to the service, click the Configure SSL button to supply the appropriate properties. For more information, see HTTP SSL Configuration Fields .

HTTP SSL Configuration Fields

Trusted Certificates Folder	No	Folder in the project containing one or more certificates from the trusted certificate authorities. This folder is checked when an invocation is made to the partner

Field	Global Var?	Description
		service to ensure that the server is trusted. This prevents connections to rogue servers.
Identity	No	This is an identity resource that contains the client's digital certificate and private key. For more information, see <i>TIBCO Designer™ Palette Reference</i>.
Strong Cipher Suites Only	No	Specifies that only cipher suites with strong encryption should be used, if they are available on the host you are connecting to.

Additional Configuration for JMS Endpoints

JNDI Tab

Context URL	Yes	The URL for the JNDI connection to the JMS server (javax.naming.Context.PROVIDER_URL).
User Name	Yes	The user name for logging into the JNDI server (javax.naming.Context.SECURITY_PRINCIPAL). If the JNDI provider does not require access control, this field can be empty.
Password	Yes	The password for logging into the JNDI server (javax.naming.Context.SECURITY_CREDENTIALS). If the JNDI provider does not require access control, this field can be empty.
Connection Factory	Yes	The QueueConnectionFactory or TopicConnectionFactory object stored in JNDI. This object is used to create a queue connection with a JMS application. The value for this field is taken from the WSDL file, if it is available in the <jms:connectionFactory> element in the service specification. You can override the

Field	Global Var?	Description
		supplied value, if you want to use a different connection factory. For more information about creating and storing QueueConnectionFactory objects, see your JNDI provider documentation.
SSL	No	When SSL is required for authentication to the service, click the Configure SSL button to supply the appropriate SSL information. For more information, see JMS SSL Configuration Fields.

JMS Tab

Use Shared JMS Configuration	No	Check this field if you want to specify a JMS Connection shared configuration resource to override the JMS transport parameters in the concrete WSDL. When checked, the JMS Connection field appears. Note: When selected, the JNDI fields get disabled.
JMS Connection	No	This field only appears when the Use Shared JMS Configuration field is selected. This field allows you to specify a JMS Connection shared configuration resource that contains the connection information you want to use.
User Name	Yes	User name to use when logging into the JMS server. If the JMS provider does not require access control, this field can be empty.
Password	Yes	Password to use when logging into the JMS server. If the JMS provider does not require access control, this field can be empty.
JMS Destination	Yes	This field is available when a JMS transport is selected.

Field	Global Var?	Description
		This field specifies the name of the destination for incoming JMS messages for this service. The syntax of the destination name is specific to the JMS provider you are using. For more information about destination names, see your JMS provider documentation. Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Delivery Mode	No	The delivery mode of the message. It can be one of the following: • Persistent : signifies the messages are stored and forwarded. • Non-Persistent : messages are not stored and may be lost due to failures in transmission. • TIBCO EMS Reliable: this mode is only available when using TIBCO Enterprise Message Service. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
Expiration	No	Corresponds to JMSExpiration property that specifies how long the message can remain active (in seconds). If set to 0, the message does not expire. This field is set in seconds, but the JMSExpiration property is displayed in milliseconds.
Priority		Priority of the message. You may set the priority to a value from 0-9. The default value is 4.

Field	Global Var?	Description
Enable W3C SOAP JMS 1.0 Compliance	No	This is a read-only field that displays whether the service is W3C SOAP Over JMS 1.0 compliant or not.
Target Service	No	This field is available only when the check box Enable W3C SOAP JMS 1.0 Compliance is selected. Specify the target service value, if any. Note: If Target Service is configured, the value is reflected in the WSDL as a query parameter.
Note: Support for values given below for the JMS tab is added for TIBCO ActiveMatrix BusinessWorks™ 5.16.1 HF-002 and above.		
OAuth Client ID	Yes	The Client ID issued to the client during the client registration process.
Client Secret	Yes	The Client Secret issued to the client during the client registration process.
Access Token URL	Yes	The endpoint for the authentication server. This is used to exchange the authorization code for the access token.
Grant Type	No	Client Credentials. Currently, only Client Credentials as a grant type is supported.
Authentication Type	No	Two types of authentication are supported: OAuth 2.0 and Basic Authentication. - If you select Basic Authentication, then the User Name and Password fields appear. - If you select OAuth 2.0, then the User Name, OAuth Client ID, Client Secret, Access Token

Field	Global Var?	Description
URL and OAuth TLS fields appear.		
OAuth TLS	No	Selecting this checkbox enables the support for generation access token from TLS based OAuth server when encrypting or decrypting Messages.
OAuth Disable Verify Hostname	No	Indicates whether the name on the server's certificate must be verified against the server's host name. If the server's host name is different from the name on the certificate, the TLS/SSL connection fails. The name on the certificate can be verified against another name by specifying OAuth Expected Hostname. Selecting this checkbox disables verification of OAuth Hostname. Deselecting this check box displays the OAuth Expected Hostname field. Default: This check box is selected.
Expected Hostname	Yes	Optional. The expected name of the remote host. The default is None.
Server Trust File	Yes	The location of the certificate. Following certificate extension supported in this File picker: pem crt cer cert

Note: When Partner Link Configuration is created, these fields get populated with the values from concrete WSDL. You can override these values in Partner Link Configuration. If the concrete WSDL is changed after configuring Partner Link Configuration, the changes are not visible in the Partner Link Configuration.

JMS SSL Configuration Fields

Basic Tab	No	Folder in the project containing one or more certificates from trusted certificate authorities. This folder is checked

Field	Global Var?	Description
Trusted Certificates Folder		when an invocation is made to the partner service to ensure that the server is trusted. This prevents connections to rogue servers.
Basic Tab Identity	No	This is an identity resource that contains the client's digital certificate and private key. For more information, see <i>TIBCO Designer™ Palette reference</i>.
Advanced Tab Trace	No	Specifies whether SSL tracing should be enabled during the connection. If checked, the SSL connection messages are logged and sent to the console.
Advanced Tab Debug Trace	No	Specifies whether SSL debug tracing should be enabled during the connection. Debug tracing provides more detailed messages than standard tracing.
Advanced Tab Verify Host Name	No	This field specifies to check the host name of the server against the host name listed in the server's digital certificate. This provides additional verification that the host name you believe you are connecting to is in fact the desired host. If the host name specified in the endpoint binding is not an exact match to the host name specified in the server's digital certificate, the connection is refused. Note: If an equivalent host name (for example, an IP address) is specified in the endpoint binding, but the name is not an exact match of the hostname in the host's digital certificate, the connection is refused.
Advanced Tab Expected Host Name	No	Specifies the name of the host you are expecting to connect to. This field is only relevant if the Verify Host Name field is also checked.

Field	Global Var?	Description
		If the name of the host in the host's digital certificate does not match the value specified in this field, the connection is refused. This prevents hosts from attempting to impersonate the host you are expecting to connect to.
Advanced Tab Strong Cipher Suites Only	No	When checked, this field specifies that the minimum strength of the cipher suites used can be specified with the <code>bw.plugin.security.strongcipher.minstrength</code> custom engine property. For more information about this property, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i>. The default value of the property disables cipher suites with an effective key length below 128 bits. When this field is unchecked, only cipher suites with an effective key length of up to 128 bits can be used.

Advanced SOAP Settings

For each operation, you can specify advanced configuration options. To access the advanced configuration options, perform the following:

1. Select Partner Link.
2. Select the operation.

The Advanced SOAP Settings dialog appears. The following sections describe the tabs on this dialog.

Input Headers

The Input Headers tab allows you to specify a SOAP message parts used for any SOAP Undescribed headers expected in the Outgoing Request.

Use the +, X, and arrow keys to add, delete and move the input header messages in the list.

The Input Headers tab has the following fields.

Field	Description
Part Name	This field contains a drop-down list with all parts of the specified SOAP message. You can specify a cardinality for the part, if applicable.
Message Name	The message in a WSDL file to use as the Undescribed input header for the outgoing SOAP request. This WSDL file does not have to be the same as the file used for output headers or the file used to configure the port type and operation of the incoming request. Use the Browse button in this field to search for valid WSDL File resources and to select the desired message in the WSDL File.
Undescribed	Select this check box to declare the header as Undescribed header.
Cardinality	This field contains a drop-down list with two options: Repeating and Optional.

Output Headers

The Output Headers tab allows you to specify a SOAP message part used for any SOAP undescribed headers expected in the Incoming Response.

Use the +, X, and arrow keys to add, delete and move the input header messages in the list.

The Output Headers tab has the following fields.

Field	Description
Part Name	This field contains a drop-down list with all parts of the specified SOAP message. You can specify a cardinality for the part, if applicable.
Message Name	The message in a WSDL file to use as the undescribed output header for the SOAP incoming response. This WSDL file does not have to be the same as the file used for input headers or the file used to configure the port type and operation of the incoming request. Use the Browse button in this field to search for valid WSDL File resources and to select the desired message in the WSDL File.

Field	Description
Undescribed	Select this check box to declare the header as Undescribed header.
Cardinality	This field contains a drop-down list with two options: Repeating and Optional.

Input Context

The Input Context tab allows you to specify a Context Resource that contains transport-specific data. The data you provide to the context resource is available to the process definition that implements the operation.

The Input Context tab has the following fields.

Field	Description
Context	The Context Resource that specifies the schema you want to use. The schema of this resource should include elements for each data element you want to provide to the output Headers for the message.
Mapping Panel	The mapping panel is similar to the mapping panel provided in the Input tab of activities. This panel allows you to map transport-specific information. Using this you can send the information as part of the Undescribed header in a SOAP Request message to the Service. The mapping panel appears only when a Context resource is specified in the Context field.
inputMessage	The root class for the input of the SOAP Request Reply activity. This class contains all input items for the activity. If this is a document request, the input is named inputMessage. If the request is an RPC, the input is the name of the operation.

Transport Properties - HTTP

Headers

Accept	This field specifies media types that are acceptable for response

Field	Description
	messages for the incoming request. For example, <code>text/*</code>, <code>text/html</code>. Media types are described in the HTTP specification. If no <code>Accept</code> header field is present, then it is assumed that the client accepts all media types.
Accept-Charset	This field specifies the character sets that are acceptable for response messages for the incoming request. For example, <code>iso-8859-5</code>, <code>unicode-1-1</code>. Character sets are described in the HTTP specification. If no <code>Accept-Charset</code> header is present, then it is assumed that the client accepts any character set.
Accept-Encoding	This field specifies the content-coding values that are acceptable for response messages. For example <code>compress</code> , <code>gzip</code> . For more information about this header field, see the HTTP specification.
Cookie	For more information about this field, see the HTTP specification.
Pragma	This field is used to include implementation-specific directives that might apply to the receiver. For more information about using this field, see the HTTP specification.

undescribed Headers

Authentication Credentials

user name	Username for logging into the HTTP server. This overrides the value specified in the Identity resource. Note: You can override the user name only if the activity is configured for Basic or NTLM authentication.
password	Password for logging into the HTTP server. This overrides the value specified in the Identity resource.

Field	Description
	Note: You can override the password only if the activity is configured for Basic or NTLM authentication.
domain	Domain for logging into the HTTP server. This overrides the value specified in the configuration. Note: You can override the domain name only if the activity is configured for NTLM authentication.

Output Context

The Output Context tab allows you to specify a Context Resource that contains data the process definition implementing the operation has provided. The data can then be mapped to Undescribed Headers of the output message returned to the client.

The Output Context tab has the following fields.

Field	Description
Shared Context	The Context Resource that specifies the schema you want to use. The schema of this resource should include elements for each data element you want to provide to the output Undescribed Headers for the message.
Expose Security Check box	Selecting this check box meant for Security Context Propagation and Security Context, makes it available as a transport data in the mapping panel.
Mapping Panel	The mapping panel is similar to the mapping panel provided in the Input tab of activities. This panel allows you to map Input Context data provided by the implementation of the operation to the transport-specific Undescribed headers of the reply message sent to the client. The mapping panel appears only when a Context resource is specified in the Context field.
outputMessage	The root class for the output of the SOAP Request Reply

Field	Description
	activity. This class contains all output items for the activity. If this is a document request, the output is names outputMessages. If the request is an RPC, the output is the name of the operation with "Response" appended.
mimeEnvelopeElement	This element contains the reply message attachments when mime attachments are used. This element contains a repeating element named mimePart that contains each mime attachment.
mimeHeaders	This element contains the mime header for each mimePart. Mime headers can contain the following information: • content-disposition • content-type • content-transfer-encoding • content-id • other mime header information Note: When the content type is specified as "text/*" (for example, "text/xml"), the attachment content is expected to be in either the textContent input element or the file name storing the attachment is expected to be in the fileName input element. When the content type is anything other than "text/*", the attachment content is expected to be in either the binaryContent input element or the file name storing the attachment is expected to be in the fileName input element.
binaryContent textContent fileName	This element contains the mime attachment. The element can be one of the following: • binaryContent – content of the attachment when the attachment is binary data. • textContent – content of the attachment when the attachment is text data.

Field	Description
	• fileName – the file name of the attachment when the Write to File field is checked on the Input Attachment Bindings sub-tab of the Advanced tab.

Transport Properties - HTTP

Headers

Allow	This field lists the set of methods supported by the resource identified by Request URI.
Content-Type	This field indicates the media type of the entity body for the outgoing message and the incoming response. Media types are described in the HTTP specification. An example of the media type is <code>text/html; charset=ISO-8850-4</code> .
Content-Length	This field indicates the size of the entity body (in decimal number of OCTETs) of the response message.
Content-Encoding	This field is used as a modifier to the content-type. When present, its value indicates what additional content encodings have been applied to the entity-body, and thus what decoding mechanisms must be applied in order to obtain the media-type referenced by the Content-Type header field. Content-Encoding is primarily used to allow a document to be compressed without losing the identity of its underlying media type. For more information about this field, see the HTTP specification.
Date	The date and time when the response message was sent.
Location	This field is used to redirect the receiver to a location other than the Request-URI for completion of the request or for identification of a new resource.
Set-Cookie	For more information about this field, see the HTTP specification.

Field	Description
Pragma	This field is used to include implementation-specific directives that might apply to the receiver. For more information about using this field, see the HTTP the HTTP Specification.

Fault Context

The Fault Context tab allows you to select a shared Context Resource that is set with SOAP fault attributes. The data can then be mapped to Fault elements of the Fault message returned to the client.

The Fault Context tab has the following fields.

Field	Description
Context	The Context Resource that specifies the SOAP fault schema you want to use. The schema of this resource should include elements for each data element you want to provide the Fault elements for the fault message. Note: For Fault Context propagation you must map Fault elements to the Context data.
Expose Security Context	Selecting this check box meant for Security Context Propagation and Security Context, makes it available as a transport data in the mapping panel.
Mapping Panel	The mapping panel is similar to the mapping panel provided in the Input Context tab of activities. This panel allows you to map data provided by the implementation of the operation to the transport-specific Undescribed headers of the reply message sent to the client. The mapping panel appears only when a Context resource is specified in the Shared Context field. The Transport Data shows the Global Variables and the Context Data shows the Context resource element.

Receive Partner Notification

Activity



The **Receive Partner Notification** activity can be used in an ActiveMatrix BusinessWorks Process Definition to invoke notification services over SOAP. The notification service sends a message that is received by the invoker.

For more information about services, partners and invoking partner services, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.



Note: The **Receive Partner Notification** activity supports only SOAP over JMS messages.

Configuration

The Configuration tab has the following fields

Field	Global Var?	Description
Name	No	The name to appear as the label for the receive partner notification activity in the process definition.
Description	No	Short description of the activity.
Partner	No	The partner whose service is to be invoked by the activity. The drop down menu lists all the partners with port types having atleast one notification service such as, OUT-only MEP.
Operation	No	Select the operation that is to be invoked from the drop down menu. The drop down menu lists all the available OUT only operations for the selected partner.

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output

The output for the receive partner notification activity is the following:

Output Item	Datatype	Description
output message	complex	The output message for the operation contains the output message parts as defined for the selected operation.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity.

For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
InvalidEnvelopeException	The incoming message contains an invalid envelope.
InternalException	There is an error during the internal processing.
ConfigurationException	There is an error in the configuration.
DeserializationException	The incoming message cannot be deserialized.

Service

Shared Configuration



A Service shared configuration resource describes an interface and its associated operations. The also specifies a set of endpoint bindings that expose the service to potential clients over specific transports.

i Note: The user interface for the Service shared configuration resource can change depending upon the additional products you have installed. For example, the ActiveMatrix BusinessWorks BPEL Extension offers more options when configuring the Service resource. When you install a product that extends the Service resource, a new field named Implementation Type appears that allows you to specify the type of service you want to configure depending upon the product you select. For more information about configuring a Service resource using those products, see the documentation for the products you have installed. This section describes the base ActiveMatrix BusinessWorks Service resource

Before creating a Service resource, you must create the following resources:

- A WSDL file that contains an abstract interface description. The interface can include one or more operations.
- Process definitions provide the implementation for each operation in the service's interface. You can specify the same process definition for more than one operation.

- Either an HTTP Connection or JMS Connection shared configuration resource that specifies the transport used by clients of the service.
- If you want to pass context information from the incoming request (for example, user name or security certificate), create a [Context Resource](#) that specifies the context data you want to use.



Warning: During deployment, Service resources must be added to a Process Archive. Although it is possible to add a Service resource to a Shared archive, the resource does not function properly when located in a Shared Archive.

Add your Service resources only to Process Archives.

Also when creating your enterprise archive file, the menu item Tools > Create Project EAR only adds processes with process starters to the Process Archive for the project. You must add any Service resources to the Process Archive by selecting the Processes tab and clicking the Add a Process Starter to this archive button.

To define a Service resource, follow this general procedure:

Procedure

1. Provide an abstract WSDL file (see [WSDL](#)) that describes the interface to the service.
2. For each operation in the interface, specify a process definition that implements the operation.
3. Specify one or more endpoint bindings to expose the service to potential clients.
4. For each endpoint binding, specify a transport configuration (either HTTP Connection or JMS Connection).
5. Specify any transport-specific information, such as endpoint URL for HTTP transports or JMS Destination Name for JMS transports.
6. Specify any SOAP details, such as SOAP Version or encoding for each operation.
7. Specify attachments, headers, or context information for input and output messages, if necessary.



Note: Only Bytes Message must be used when using MTOM attachments

For more information about services, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the resource.
Description	No	Short description of the resource.
Enable for AMX	No	Selecting this check box allows you to configure this Service resource in ActiveMatrix BusinessWorks. It also allows you to configure an ActiveMatrix BusinessWorks endpoint for the existing services.

Note: Select this check box only if you want to expose your Service to theActiveMatrix BusinessWorks world.

Warning: If this check box is selected and cleared, you loose the Partner Link Configuration information from the partner binding table.

Tabs visible after selecting the check box are:

- Services
- References
- Properties
- WSDL Source
- Overview

Following tabs are visible when this check box is *not* selected (by default):

- Configuration
- Partner Binding
- WSDL Source

Field	Global Var?	Description

- Overview

Implementation	No	This field allows you to add abstract WSDL files that describe the service. Click the + button to add a WSDL file. Click the X button to remove a WSDL file. Click the Rename Current Service Port to change the name of the currently selected service port.

Service Port Sub-Tabs

Each service port have a sub-tab containing the configuration for the service port.

Interface Namespace	No	Namespace for the interface file of the service. You can click the Go To Referenced Resource to view the interface file.

Operations Sub-Tab

This sub-tab contains a table of all operations defined in the interface file. This table allows you to specify the resource that implements each operation.

Operation Name	No	Name of the operation. This column is a read-only list of operations contained in the interface file.
Operation Implementation	No	Double click on this field and use the Browse button to locate the resource that implements each operation. Process definitions that implement an operation must have an input, output, and error schemas that match the input, output, and fault messages of the operation. You can leave the implementation for an operation unspecified. In this case, an exception is thrown when a client attempts to invoke the unimplemented operation.

Field	Global Var?	Description

Note: You can drag and drop a process definition from the project tree to the operation implementation cell. If the signature of the operation matches the process definition, the process is inserted. If the process definition or the WSDL changes, the icon for the process definition in the cell changes to a warning triangle indicating the operation and implementation signatures no longer match.

Endpoint Bindings Sub-Tab

This sub-tab allows you to specify one or more endpoints that expose the service as well as the transport bindings for each endpoint.

Use the + button to add endpoints, use the X button to delete endpoints, and use the arrow buttons to move the selected endpoint in the list.

Endpoint Name	No	Specify a name for the endpoint in this field.
Endpoint Type	No	Specify the type of endpoint by selecting from the list of types. - Local – A local endpoint is available to consumers within the same process engine. This type of endpoint provides a highly efficient way for local services to invoke each other without incurring the overhead of using a network transport. - SOAP – A SOAP endpoint is used to expose the service by way of the SOAP protocol to other partner services or web clients

Transport Sub-Tab

Transport	No	Use the Browse button to select a transport. In this release, HTTP Connection or JMS Connection resources can be used as transports.

Field	Global Var?	Description
Use Basic Authentication	Yes	This field is available when an HTTP transport is selected. When checked, this field specifies that incoming SOAP requests must supply a valid username and password. The user name and password specified in the incoming request must exist in the domain (users are created and managed in the domain using TIBCO Administrator).
Endpoint URI	No	This field is available when an HTTP transport is selected. This field specifies the Endpoint URI that clients can use to access the service.
JMS Destination	Yes	This field is available when a JMS transport is selected. This field specifies the name of the destination for incoming JMS messages for this service. The syntax of the destination name is specific to the JMS provider you are using. For more information about destination names, see your JMS provider documentation. Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
JMS Destination Type	No	This field is available when a JMS transport is selected. This field specifies whether the JMS Destination is a Topic or a Queue.
JMS Message Type	No	This field is available when a JMS transport is selected. This field specifies the type of incoming messages on the specified destination. It can be either Text Message or Bytes message.

Field	Global Var?	Description
Enable W3C SOAP JMS 1.0 Compliance	No	Select the check box to comply with the W3C SOAP Over JMS 1.0 specification. By default, the check box is not selected and the service complies with the TIBCO specifications.
Target Service		This field is available when a JMS transport is selected. This field is available only when the check box Enable W3C SOAP JMS 1.0 Compliance is selected. Specify the target service value, if any. Note: If Target Service is configured, the value is reflected in the WSDL as a query parameter.
Reply Using Incoming Delivery Mode	No	This field is available when a JMS transport is selected. Select the check box if you want to send the reply using the incoming delivery mode. By default, the check box is not selected.
JMS Delivery Mode (Response)	Yes	This field is available when the Reply using Incoming Delivery Mode field is not selected. The delivery mode of the response message. Can be one of the following: • PERSISTENT: signifies the messages are stored and forwarded. • NON-PERSISTENT : messages are not stored and may be lost due to failures in transmission. • TIBCO EMS Reliable : this mode is only available when using TIBCO Enterprise Message Service. For more information about this mode, see the TIBCO Enterprise Message Service documentation.

Field	Global Var?	Description
By default, the field selection is PERSISTENT.		
Acknowledge Mode	No	This field is available when a JMS transport is selected. This field specifies the acknowledgment mode to be used for incoming messages. It can be one of the following: • Auto : the message is automatically acknowledged when it is received. • Client: the message is acknowledged when the process implementing the operation ends successfully. • Dups OK : the message is acknowledged automatically when it is received. JMS provides this mode for lazy acknowledgment, but ActiveMatrix BusinessWorks acknowledges messages upon receipt. • TIBCO EMS Explicit : (only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, are redelivered instead of all messages in the session. The session is not blocked and one session handles all incoming messages for each process instance. • Note: In the ActiveMatrix BusinessWorks 5.10 release, a change occurs in the behavior of this feature. Now only the messages that are not confirmed, are redelivered. • TIBCO EMS Explicit Dups OK : (only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, are redelivered instead of all messages in the session.

Field	Global Var?	Description
		The session is not blocked and one session handles all the incoming messages for each process instance. The messages however, would be lazily acknowledged.
JMS Application Properties		This field is available when a JMS transport is selected. Any application-specific message properties that is part of the message. This is specified by the JMS Application Properties shared configuration object.
JMS Application Properties (Response)		This field is available when a JMS transport is selected. Any application-specific message properties that is part of the response message. This is specified by the JMS Application Properties shared configuration object.
Max Sessions	No	This field is available when a JMS transport is selected and Client is selected as the Acknowledge Mode. If the JMS Destination Type is Topic, this field is read-only. If the JMS Destination Type is Queue, the value of this field can be altered. This field specifies the maximum number of JMS sessions to create for incoming queue messages. When a JMS queue message is received, the session is blocked until the message is acknowledged. Because the acknowledgment can come at a later time when the process ends, this field allows you to specify a maximum number of new sessions to be created for handling incoming messages. Once the maximum number of sessions is reached, no new incoming messages can be processed. Once an incoming message is confirmed, the total number of active sessions is decreased and another incoming message can be processed.

Field	Global Var?	Description
Operations	No	The operation that this process definition implements. The drop-down list in this field is populated with the operations contained in the specified port type in the specified WSDL. You can choose one operation to implement per process definition. If you want to implement multiple operations, you must create a process definition for each.

SOAP Details Sub-Tab

Default Style	No	The SOAP binding style for operations that do not explicitly set their binding style. You can specify either Document or RPC style.
SOAP Version	No	Specify either SOAP 1.1 or SOAP 1.2 for the version of the SOAP specification to which incoming messages should comply. For more information, see SOAP Specification Compliance.
Operations	No	Select each operation from the drop down list to specify the SOAP Action, Style, or Encoding for the operation.
SOAP Action	Yes	The soapAction that is expected from incoming SOAP requests. This field is required, and by default, the operation name is used as the soapAction. Each operation in an interface must have a unique value for SOAP Action. For more information about soapAction, see the SOAP specification.
Style	No	The SOAP binding style for the selected operation. You can specify either Document or RPC style, or you can specify that the value in the Default Style field should be

Field	Global Var?	Description
		used.
Encoding	No	The encoding type for the body of the SOAP input and output messages. This can be either literal or encoded. Encoded messages support more complex datatypes such as SOAP arrays. When encoded is specified, you can optionally specify the namespace for input, output, and fault messages.
Input Message Namespace	No	The namespace for input messages.
Output Message Namespace	No	The namespace for output messages.
Fault Message Namespace	No	The namespace for fault messages.

Advanced SOAP Settings

For each operation, you can specify advanced configuration options. To access the advanced configuration options, perform the following:

Procedure

1. Click the **Endpoint Bindings** Tab.
2. Select the desired endpoint and click the SOAP Details sub-tab.
3. Select the operation in the Operations field.
4. Click the **Advanced** button next to the operation name.

The Advanced SOAP Settings dialog appears. The following sections describe the tabs on this dialog.

Input Attachments

For more information about message attachments, see [Messages With Attachments](#).

The Input Attachments tab has the following fields.

Field	Description
Mime Content Type	The table containing this column allows you to specify the mime content type of each message part. This is especially useful for describing parts that contain attachments. For more information on the valid syntax of the content type, see the mime specification . If specified, the content type appears in the WSDL for the specified part. Use the X, +, and arrow keys to delete, add, and move parts in the list.
Message Part	The message part that you would like to specify the content type for. This field is a drop down list of message parts.
Write to File	Checking this field specifies that incoming messages whose attachments exceed the specified threshold size should be written to a file instead of stored in memory. This allows you to accept large incoming attachments without consuming a great deal of memory. When this field is checked, the Directory and Threshold Size fields appear. Leaving this field unchecked specifies that attachments of incoming messages should be kept in memory. Once written, the files created by using this option are not deleted automatically. You must manage the storage used by these files and delete them when they are no longer used.  Although Write to File exists for SwA and MTOM, Soap over JMS completely depends on the heap size configuration as JMS specifications do not support writing JMS messages in parts as is the case with Soap over HTTP, which can be written over wire in parts. Hence, writing large JMS messages to JMS after configuring Write to File, completely depends on the Heap size.
Directory	This field only appears when the Write to File field is checked. The directory to write attachments that are above the specified threshold.

Field	Description
	The process engine does not attempt to create the directory if the specified directory does not exist. Therefore, create the directory before starting the process engine.
Threshold Data Size (bytes)	This field only appears when the Write to File field is checked. The maximum size (in bytes) of an incoming message attachment that can be kept in memory. Attachments larger than the specified size are written to a file in the specified directory. The file's name is output so that subsequent activities in the process definition can access the file and read its contents. Specifying zero (0) in this field causes all incoming messages to be saved to a file.

Input Headers

The Input Headers tab allows you to specify a SOAP message parts used for any SOAP headers expected in the incoming request. The specified input headers are also included in the concrete bindings in the generated WSDL file.

Use the +, X, and arrow keys to add, delete and move the input header messages in the list.

The Input Headers tab has the following fields.

Field	Description
Message Name	The message in a WSDL file to use as the input header for the incoming SOAP request. This WSDL file does not have to be the same as the file used for output headers or the file used to configure the port type and operation of the incoming request. Use the Browse button in this field to search for valid WSDL File resources and to select the desired message in the WSDL File.
Part Name	This field contains a drop down list with all parts of the specified SOAP message. You can specify a namespace or encoding for the part, if applicable.

Field	Description
Namespace	Namespace for the selected input header message.
Encoding	Encoding to use for the selected input header message.
Undescribed	Select this check box to declare the header as Undescribed header.
Cardinality	This field contains a drop-down list with two options: Repeating and Optional.

Output Attachments

For more information about message attachments, see [Messages With Attachments](#).

The Output Attachments tab has the following fields.

Field	Description
Attachment Style	This field allows you to choose the attachment style for the outbound/response messages. Two attachment styles are available: SwA and MTOM.

Attachments

Mime Content Type	The table containing this column allows you to specify the mime content type of each message part. This is especially useful for parts that contain attachments. For more information on the valid syntax of the content type, see the mime specification. If specified, the content type appears in the WSDL for the specified part. Use the X, +, and arrow keys to delete, add, and move parts in the list.
Message Part	The message part that you would like to specify the content type for. This field is a drop down list of message parts.

Output Headers

The Output Headers tab allows you to specify a SOAP message parts used for any SOAP headers expected in the outgoing reply. The specified output headers are also

included in the concrete bindings in the generated WSDL file.

Use the +, X, and arrow keys to add, delete and move the input header messages in the list.

The Output Headers tab has the following fields.

Field	Description
Message Name	The message in a WSDL file to use as the output header for the SOAP request. This WSDL file does not have to be the same as the file used for input headers or the file used to configure the port type and operation of the incoming request. Use the Browse button in this field to search for valid WSDL File resources and to select the desired message in the WSDL File.
Part Name	This field contains a drop-down list with all parts of the specified SOAP message. You can specify a namespace or encoding for the part, if applicable.
Encoding	Encoding to use for the selected output header message.
Undescribed	Select this check box to declare the header as Undescribed header.
Cardinality	This field contains a drop-down list with two options: Repeating and Optional.

Fault Headers

This Fault Headers tab enables the user to configure headers for declared faults. With this you are able to configure different schemas to be mapped to the headers for different Fault messages.

This Fault Headers tab has the following fields.

Field	Description
soapFaults	The Faults available for configuration are listed in the soapFaults drop-down.

Field	Description
Part Name	This field contains a drop-down list with all parts of the specified soapFault message.
Message Name	The message in a WSDL file to use as the fault header for the incoming soapFault request. Use the Browse button in this field to search for valid WSDL File resources and to select the desired fault message in the WSDL File.

Input Context

The Input Context tab allows you to specify a [Context Resource](#) that contains transport-specific data. The data you provide to the context resource is available to the process definition that implements the operation.

The Input Context tab has the following fields.

Field	Description
Shared Context	The Context Resource that specifies the schema you want to use. The schema of this resource should include elements for each data element from the transport information you want to provide to a process definition that implements the operation.
Expose Security Context	This field is available for either HTTP or JMS transports. This field is only available when a Context resource is specified in the Shared Context field. Places the information from the user's security context (either authentication or SSL certificate information) into the Context/SecurityContext element of the Input Context. This is a fairly expensive operation and SSL certificates can consume memory resources, so check this option only if you require information from the user's security context for later use in your process definition.
Mapping Panel	The mapping panel is similar to the mapping panel provided in the Input tab of activities. This panel allows you to map transport-specific information from the incoming request to the data elements of the specified Context Resource.

Field	Description
	The mapping panel appears only when a Context resource is specified in the Shared Context field. For more information about mapping and transforming input data, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.

When the Expose Security Context field is checked, the following schema is available in the Mapping Panel field.

Element	Datatype	Description
Context	complex	This element holds the optional SecurityContext element and the RemoteAddress and RemoteHost elements described below.
SecurityContext	complex	This output element is only available when the Expose Security Context field is checked on the Configuration tab. This element contains information about the requestor's authentication or certificate information, depending upon the security scheme used. For more information, see the description of the following elements contained in the SecurityContext element.
Transport	complex	This element is only available when HTTP is the transport, this element is not available when JMS is the transport. This element contains the transport-level security context. This includes the following: • CertificateToken— complex element containing the subject, issuer, serial number, fingerprint, and X509 certificate. • UsernameToken – complex element containing the username, password, and nonce (random number, if one is set by the calling application).
WSSE	complex	This element contains the WSS policy security context.

Element	Datatype	Description
		This includes the Authentication and Signature elements. See the description of these elements below.
Authentication	complex	This element contains the authentication information from the client. The type of information in this elements depends upon how the client authenticated. This element can contain one of the following: - UsernameToken – complex element containing the username, password, and nonce (random number, if one is set by the calling application). - KeyInfo – the client’s public key.
Signature	complex	This element contains information from the user’s digital signature. This element contains the following: - Value – this element contains the raw signature sent by the client. - Token – either a UsernameToken or a KeyInfo element. A UsernameToken element contains the username, password, and nonce, if available. A KeyInfo element contains information about the client’s digital certificate or public key.
RemoteAddress	String	IP address of the remote host that sent the HTTP request. This element is only available when HTTP is the specified transport. This element is not available for the JMS transport.
RemoteHost	String	Host name of the remote host that sent the HTTP request. This element is only available when the Enable DNS Lookups field is checked on the Advanced tab of the HTTP Connection used by this

Element	Datatype	Description
		activity.
		This element is only available when HTTP is the specified transport. This element is not available for the JMS transport.

Output Context

The Output Context tab allows you to specify a [Context Resource](#) that contains data the process definition implementing the operation has provided. The data can then be mapped to Headers of the output message returned to the client.

The Output Context tab has the following fields.

Field	Description
Shared Context	The Context Resource that specifies the schema you want to use. The schema of this resource should include elements for each data element you want to provide to the output Headers for the message.
Mapping Panel	The mapping panel is similar to the mapping panel provided in the Input tab of activities. This panel allows you to map data provided by the implementation of the operation to the transport-specific headers of the reply message sent to the client. The mapping panel appears only when a Context resource is specified in the Shared Context field. For more information about mapping and transforming input data, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.

Fault Context

The Fault Context tab allows you to select a fault context resource that is set with SOAP fault attributes. The data can then be mapped to Fault Elements of the Fault message returned to the client.

The Fault Context tab has the following fields.

Field	Description
Shared Context	The Context Resource that specifies the SOAP fault schema you want to use for mapping with the Fault Elements.
Mapping Panel	The mapping panel is similar to the mapping panel provided in the Input tab of activities. This panel allows you to map data provided by the implementation of the operation to the transport-specific Fault Elements of the fault message sent to the client. The mapping panel appears only when a Context resource is specified in the Shared Context field. The Context Data shows the Context resource element.

Partner Binding

The Partner Binding tab allows you to bind partners invoked by all operations in a service on the Partner Binding tab of the Service. Only partners that require binding appear on this tab. The Partner Binding tab has the following fields:

Field	Description
Partner Bindings	The Partner Bindings field allows you to specify binding information for partner definitions. The Partner Name and Process fields in the table are non-editable. Partner Name - name of the partner as specified in the Process Definition. Process - The process in which the partner is declared. PortType Namespace - is a non-editable field displaying the PortType Namespace corresponding to the Port Type selected. Partner Link - the partner link configuration for the partner. This field can optionally be specified by double-clicking on the field and browsing the list of available partner link resources in the project.
Dynamic Call Processes	The Dynamic Call Processes field allows you to specify a list of processes that can be dynamically invoked by processes invoked by all operations in the service. You can add or remove processes from the list.

i Note: Processes declared in the Dynamic Call Processes field are automatically included in Enterprise Archive (EAR) files.

WSDL Source

The WSDL Source tab displays the concrete WSDL interface file that is generated based on the information specified on the Configuration tab of this resource. This tab has the following fields.

Field	Global Var?	Description
Service URI	No	The Service URI portion of the URL that can be used to retrieve the WSDL file. For clients to retrieve WSDL files from ActiveMatrix BusinessWorks, you must define a process definition that accepts HTTP requests and uses the Retrieve Resources activity to generate the WSDL file. For more information, see Using Retrieve Resources. The URL to retrieve WSDL files is the following: <pre>http://<host>:<port>/<serviceURI>?wsdl</pre> where <serviceURI> is the value in this field.
Target Namespace	No	Target namespace of the Service. Note: The Target Namespace of the service should be an abstract WSDL if embedding Interface or Types.
Embed In WSDL	No	Check one or more of the following to embed the selected item(s) in the concrete WSDL file instead of using import statements: Interface – When checked, specifies that all referenced WSDL files should be included inline in the concrete WSDL file. Otherwise, all referenced WSDL files are

Field	Global Var?	Description
		imported. Note: To use Abstract WSDL in the Partners tab of the Client Process Definition containing Invoke Partner activity, uncheck Interface. - Types – When checked, specifies that all referenced XSD files should be included inline in the concrete WSDL file. Otherwise, all referenced XSD files are imported. - JNDI Properties – When checked, specifies that JNDI properties should be included inline in the service specification in the WSDL Source tab.
WSDL Code	No	This field displays the concrete WSDL file that describes this service. Click the Save WSDL File button to save the concrete WSDL file to disk, if desired.

Overview

The Overview tab provides a tree view of the service that you can expand and collapse to view the exposed endpoints and operations for the service.

Set Context

Activity



The **Set Context** activity allows you to set the value of a [Context Resource](#) shared configuration resource. This is useful if you want to supply data from the process definition implementing an operation to the service agent. For example, you may want to set some headers of the outgoing reply message.

For more information about context separation for Service resources and operation implementations, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Context Configuration	No	The Context Resource whose value you would like to change.
Output New Value	No	When checked, the new value of the context resource is placed into this activity's output. The value is then available to subsequent activities in the process definition. If not checked, this activity has no output.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
<schema>	varies	The schema of the context resource specified in the Context Configuration field on the Configuration tab is the input schema for this activity.

Output

The output for the activity is the following.

Output Item	Datatype	Description
<schema> or no output	varies	When the Output New Value field on the Configuration tab is unchecked, this activity has no output. When the field is checked, the new value of the context resource is output. The schema for the output is determined by the schema specified for the context resource.

SOAP Palette

Simple Object Access Protocol (SOAP) is a lightweight, XML-based protocol for the exchange of information between applications. The **SOAP** palette, along with the [WSDL Palette](#) allow you to implement web services. The **SOAP** palette allows you do the following:

- create web services by using process definitions
- generate WSDL files for web service clients
- invoke one-way, request/reply web services
- use either HTTP or JMS as the transport for web services
- send or receive SOAP requests with attachments

SOAP Specification Compliance

The SOAP specification is maintained by the World Wide Web Consortium (W3C). You can find the SOAP specification at www.w3.org/TR/SOAP. ActiveMatrix BusinessWorks can send and receive messages that comply with either the 1.1 or 1.2 version of the SOAP specification. For SOAP 1.2 messages, not all of the features described in the specification are available in ActiveMatrix BusinessWorks.

Receiving SOAP Messages with the SOAP Event Source

The SOAP Event Source process starter can be configured to receive either SOAP 1.1 or 1.2 compliant messages. You specify the version of the incoming messages in the SOAP Version field on the Configuration tab of the SOAP Event Source. SOAP Reply and SOAP Fault activities in the same process definition use the same SOAP message version as specified in the SOAP Event Source. SOAP Event Source can handle inbound SOAP messages with attachments with either SwA or MTOM attachment styles.

Using the SOAP Request/Reply Activity

The **SOAP Request/Reply** activity sends either a SOAP 1.1 or 1.2 compliant message depending upon the SOAP version specified in the WSDL file. If a SOAP 1.2 message is required, ActiveMatrix BusinessWorks sends a message that complies with the SOAP 1.2 specification. When the SOAP server returns a Fault message, the SOAP 1.2 fault message is mapped to the ActiveMatrix BusinessWorks fault schema as described in [Fault Schemas](#).

Fault Schemas

The specification of the schema for SOAP Fault messages has changed between release 1.1 and 1.2 of the SOAP specification. ActiveMatrix BusinessWorks uses the same schema for SOAP Fault messages regardless of the version of the specification. Because of this, the content of Fault messages that comply with SOAP 1.2 is mapped to the ActiveMatrix BusinessWorks SOAP Fault schema.

[SOAP Fault schema relationship](#) describes the relationship of the data elements between the ActiveMatrix BusinessWorks SOAP Fault schema and the SOAP Fault 1.1 and 1.2 schemas..

SOAP Fault schema relationship

ActiveMatrix BusinessWorks SOAP Fault Schema Element	SOAP Fault Schema Element 1.1	SOAP Fault Schema Element 1.2
faultcode	faultcode	Code/Value Note: Code/Subcode is not present.
faultstring	faultstring	Reason/Text[1] Note: There is only one Text element within the Reason element.
faultfactor	faultfactor	Role
detail	detail	Detail

Retrieve Resources

Activity



The **Retrieve Resources** activity generates a WSDL file containing a concrete service description of any process definition that has a [SOAP Event Source](#) process starter. This allows clients to access the WSDL for a web service that is implemented by a ActiveMatrix BusinessWorks process definition. The client can then use the WSDL file to invoke the web service.

The **Retrieve Resources** activity can also be used to retrieve any other resources, including XSDs and WSIL.

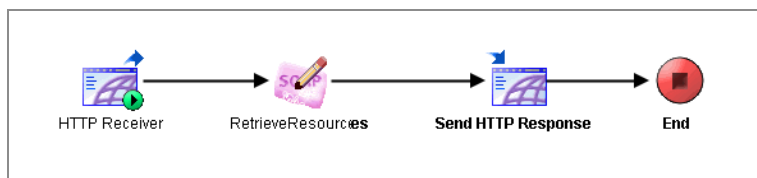
Using Retrieve Resources

This activity is normally used in conjunction with an [HTTP Receiver](#) process starter and a [Send HTTP Response](#) activity. The HTTP Receiver process starter can receive an HTTP request and pass it along to the **Retrieve Resources** activity. When a request is made to retrieve a WSDL file for a process definition that has a [SOAP Event Source](#) process starter, Retrieve Resources creates a WSDL file dynamically. This file contains the concrete service definition of the specified ActiveMatrix BusinessWorks process definition. The WSDL file or other requested resources are then placed into the output for this activity.

You can then pass the output of the **Retrieve Resources** activity to the **Send HTTP Response** activity to send the response back to the requestor.

[Process definition using the Retrieve Resources activity](#) illustrates a process definition using the **Retrieve Resources** activity.

Figure 26: Process definition using the Retrieve Resources activity



Note: The HTTP Receiver process starter can be configured to use the HTTPS protocol for SSL communication when retrieving resources. For more information, see [HTTP Receiver](#).

When sending the HTTP response, set the Headers/Content-Type item on the Input tab to the appropriate type for the type of resource retrieved. Use "text/xml" when this activity is used to retrieve a WSDL file. For more information on the format of the URL for the HTTP request for retrieving resources, see [Format of URLs for HTTP Requests](#).

For more information about binding the HTTP request to this activity's input and binding this activity's output to an HTTP response, see the descriptions of the Input and Output tabs for this activity.

If you want to test a process definition containing this activity, you must load the process definition containing this activity along with the process definition containing the SOAP Event Source process starter into the test window. For more information about loading multiple process definitions when testing, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Format of URLs for HTTP Requests

To use the **Retrieve Resources** activity, the HTTP request must have one of the following forms:

- `http://<host>:<port>/<path>/<resourceName>?wsdl`
- `http://<host>:<port>/<path>/<resourceName>`
- `http://<host>:<port>/inspection.wsil`

Option	Description
<host>	Host name of the machine that is listening for the HTTP request.
<port>	Port is the port number on the machine for incoming HTTP requests. This is specified on the Configuration tab of the HTTP Receiver process starter.
<path>	The location of the resource in the ActiveMatrix BusinessWorks project tree.
Note: Any special characters in the path or resource name (such as &, =, or space) are mangled in the generated WSDL file. It is a good idea to avoid special characters in the folder and resource names.	

Option	Description
<resourceName>	The name of the resource to retrieve. If you are retrieving a WSDL file for a process definition, use the name of the process definition followed by ?wsdl. If you are retrieving other resources (such as XSD files, images, and so on), specify the name of the resource, such as mySchema.xsd, myProcess.process, or myImage.gif. Note: Use <i>resourceName.serviceagent</i> to reference a Receive Partner Notification resource
inspection.wsil	Used to retrieve a Web Services Inspection Language (WSIL) file containing a list of all configured web services. Any process definitions with SOAP Event Source process starters in the project get contained in the WSIL file.

Examples

The following is an HTTP request that retrieves the WSDL file for the process named GetPurchaseOrder stored in the folder Purchasing:

```
http://purch:8877/Purchasing/GetPurchaseOrder?wsdl
```

The following is an HTTP request that retrieves a WSIL file containing all process definitions with SOAP Event Source process starters:

```
http://purch:8877/inspection.wsil
```

The following is an HTTP request that retrieves the PurchaseOrder.xsd file:

```
http://purch:8877/Purchasing/Schemas/PurchaseOrder.xsd
```

The following is an HTTP request that retrieves the WSDL file for the Service resource named BuyerService :

```
http://server:8877/BuyerService.serviceagent?wsdl
```

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
resourcePath	string	The resource name (and its location in the project tree) that you want to retrieve. This overrides to the <path> and <resourceName> portion of the incoming HTTP request that is used to retrieve ActiveMatrix BusinessWorks resources. For a description of the URL format for HTTP requests, see Format of URLs for HTTP Requests.
filter	string	This overrides the query string of the incoming HTTP request to specify the type of filter to use when retrieving resources. The only available filter is "wsdl" to retrieve the concrete WSDL for a process definition that implements a SOAP service. You can use this input element to allow incoming HTTP requests to omit the query string when retrieving concrete

Input Item	Datatype	Description
		WSDL files. You can specify an XPath expression in this element that examines the incoming request and sets this element to "wsdl" when the incoming request matches a process definition name.
hostname	string	The hostname to specify for the concrete service endpoint in the dynamically generated WSDL file. Note: This value is used when creating the import statements (WSDL descriptions, WSIL, and XSDs).
port	string	The port number to specify for the concrete service endpoint in the dynamically generated WSDL file. Note: This value is used when creating the import statements (WSDL descriptions, WSIL, and XSDs).

Output

The output for the activity is the following.

Output Item	Datatype	Description
outputClass	complex	The root class for the output of the Retrieve Resources activity. This class contains all output items for the activity.
resourceData	string	The resource that can be used by a client to invoke the specified process as a web service. You can pass this data back to the HTTP requestor by using a Send HTTP Response activity and binding this value to the asciiContent of the response.
resourceBytes	binary	The retrieved resource as binary data. This is useful if

Output Item	Datatype	Description
		binary data, such as an image, is part of the resource. If the retrieved resource is only binary data, the resourceData string element is not available. You can pass this data back to the HTTP requestor by using an HTTP Send Response activity and binding this value to the binaryContent of the response.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
ServerException	An error occurred and the resource could not be sent to the client.

Built-in Resource Provider

ActiveMatrix BusinessWorks provides an alternative way to retrieve resources using a feature called Built-in Resource Provider.

This feature enables the ActiveMatrix BusinessWorks engine to listen on a configured port and retrieve resources such as a WSDL file, XSD file, Process file, Image file, and so on. This out-of-the-box solution performs tasks similar to that of the **Retrieve Resource** activity, but eliminates the need to create a process definition to use the **Retrieve Resource** activity. For details about the process definition, refer to [Process definition using the Retrieve Resources activity](#).

Since the feature works the same as **Retrieve Resource** activity, the client side functionalities with respect to **Retrieve Resource** activity remain the same.

However, the only exception is that this feature does not support Client authentication for SSL.

ActiveMatrix BusinessWorks provides parameters to enable or disable the feature, and override the defaults configuration. You can set these values in the `bwengine.xml` to be reflected in the TIBCO Administrator after deploying the project.

BW.platform.services.retreiveresources.Enabled

This property specifies whether the Built-in Resource Provider feature is enabled or not. Setting it to true enables the feature. By default it is set to false.

BW.platform.services.retreiveresources.Httpport

This property specifies the port number that is configured to listen for incoming HTTP requests. This property is set to 8010 by default.

BW.platform.services.retreiveresources.isSecure

This property specifies if the incoming requests must use the HTTPS (secure socket layer) protocol. The HTTPS protocol authenticates the server to the client. By default, this property is set to false.

BW.platform.services.retreiveresources.enableLookups

This property, when set to true, enables the HTTP client to look up a Domain Name System and resolve the IP address to a DNS Name.

Setting this property to true adversely affects the throughput. Hence this property can be enabled only when required. By default, this property is set to false.

BW.platform.services.retreiveresources.defaultEncoding

This property specifies the encoding to use for the URL if no charset is specified in the Content Type header of the message. By default, this property is set to ISO-8859-1.

BW.platform.services.retreiveresources.identity

This property specifies the Identity resource that contains the HTTP Server's digital certificate and private key. This property is available when the `bw.platform.services.retreiveresources.isSecure` property is set to true.

BW.platform.services.retreiveresources.Hostname

This property specifies the hostname to which the request has been sent. It is useful to identify the exact host in a multi-host configuration to whom the request has been sent. By default, this property is set to localhost.

Note: When an Enterprise Archive file, generated with ActiveMatrix BusinessWorks 5.4.x or earlier versions, is deployed in TIBCO Administrator, the built-in resource provider parameters are not available in TIBCO Administrator. To enable specifying the built-in resource provider parameters, recreate the Enterprise Archive file with ActiveMatrix BusinessWorks 5.6.0 and deploy in TIBCO Administrator.

SOAP Event Source

Process Starter



The SOAP Event Source process starter creates a process instance for incoming SOAP requests. SOAP is a standard protocol for invoking web services. This allows you to create a web service using process definitions.

At runtime, a client can retrieve the WSDL file for a process containing this process starter using an HTTP request (For more information, see the description of the [Retrieve Resources](#) activity). Once the WSDL file is retrieved, the client can perform a SOAP request to invoke the web service.

WSDL File and SOAP Event Source

A WSDL File specifies the characteristics (input message, output message, fault message, and so on) of a web service. The SOAP Event Source process starter configuration determines the concrete WSDL file that describes the web service that the process definition implements.

The Configuration tab of this process starter requires that you specify a Port Type. This port type is taken from a WSDL file resource. Instead of creating a new WSDL File resource, you can use an existing concrete service description. However, the concrete description of the service get generated based on the configuration of this process starter.

i Note: Only Bytes Message must be used when using MTOM attachments.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Port Type	No	The Port Type field is a two-part field. The first part is the namespace of the stored WSDL file that contains the service description. This is specified or imported as a WSDL File resource. For more information about specifying abstract and concrete service descriptions, see WSDL Palette. The second part of the Port Type field is the name of the port type in the WSDL file that contains the operation you want to implement with this process definition. This is a drop down list that is populated after the WSDL file is specified.
Operation	No	The operation that this process definition implements. The drop down list in this field is populated with the operations contained in the specified port type in the specified WSDL. You can choose one operation to implement per process definition. If you want to implement multiple operations, you must create a process definition for each.
Transport	No	The transport used by the incoming SOAP request. The transport can be one of the following: HTTP – specify a HTTP Connection shared configuration resource.

Field	Global Var?	Description
		JMS – specify a JMS Connection shared configuration resource. Both HTTP and JMS allow you to use the SSL protocol for secure connections.
SOAP Version	No	Specify either SOAP 1.1 or SOAP 1.2 for the version of the SOAP specification to which incoming messages should comply. For more information, see SOAP Specification Compliance.
Expose Security Context	No	Places the information from the user's security context (either authentication or SSL certificate information) into the Context/SecurityContext output element. This is a fairly expensive operation and SSL certificates can consume memory resources, so check this option only if you require information from the user's security context for later use in your process definition.

Transport Details

The Transport Details tab allows you to specify additional configuration information based on the type of transport used for the SOAP request. For more information about SOAP over the JMS transport, see [SOAP Messages Over the JMS Transport](#).

The Transport Details tab contains the following fields:

Field	Global Var?	Description
HTTP Transport		
Use Basic Authentication	No	When checked, this field specifies that incoming SOAP requests must supply a valid username and password.

Field	Global Var?	Description
		The user name and password specified in the incoming request must exist in the domain (users are created and managed in the domain using TIBCO Administrator).
Endpoint URI	No	Specify endpoint URI here. This is optional.

JMS Transport

JMS Destination	No	Name of the JMS destination for incoming requests. The syntax of the destination name is specific to the JMS provider you are using. For more information about destination names, see your JMS provider documentation. Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field and a destination type in the JMS Destination Type field. The Browse button displays a list of configured destinations in the JMS server that are of the specified type.
JMS Destination Type	No	Specify either Queue or Topic as the destination type.
Durable Subscription	No	This field is only available when Topic is selected in the JMS Destination Type field. This field specifies a durable subscription. You must specify a name in the Subscription name field that is registered with the JMS application as the durable subscription name.
Subscription Name	No	This field is only available when Topic is selected in the JMS Destination Type field. This field specifies the subscription name registered with

Field	Global Var?	Description
		the JMS application for durable subscriptions. This field is only available when the Durable subscription field is checked. Once a durable subscription is created, it can only be removed by accessing the administration tool of the JMS provider. For more information about managing durable subscriptions, see the documentation of your JMS provider.
JMS Message Type	Yes	The type of the message. This can be one of the following: • Bytes Message – A stream of bytes. • Text Message – The message is a <code>java.lang.String</code>.
Acknowledge Mode	Yes	This field is available when a JMS transport is selected. This field specifies the acknowledgement mode to be used for incoming messages. It can be one of the following: • Auto : the message is automatically acknowledged when it is received. • Client: the message is acknowledged when the process implementing the operation ends successfully. • Dups OK : the message is acknowledged automatically when it is received. JMS provides this mode for lazy acknowledgement, but ActiveMatrix BusinessWorks acknowledges messages upon receipt. • TIBCO EMS Explicit : (only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, are redelivered instead of all messages in the session. The session is not blocked and one session handles all incoming messages for each process instance.

Field	Global Var?	Description
		Note: In the ActiveMatrix BusinessWorks 5.10 release, a change occurs in the behavior of this feature. Now only the messages that are not confirmed, are redelivered. TIBCO EMS Explicit Dups OK : (only available for TIBCO Enterprise Message Service) a message which is not acknowledged using the Confirm activity before the process instance ends, are redelivered instead of all messages in the session. The session is not blocked and one session handles all the incoming messages for each process instance. The messages however, would be lazily acknowledged.
Max Sessions	No	This field is available when Client is selected as the Acknowledge Mode. If the JMS Destination Type is Topic, this field is read-only. If the JMS Destination Type is Queue, the value of this field can be altered. This field specifies the maximum number of JMS sessions to create for incoming queue messages. When a JMS queue message is received, the session is blocked until the message is acknowledged. Because the acknowledgement can come at a later time when the process ends, this field allows you to specify a maximum number of new sessions to create to handle incoming messages. Once the maximum number of sessions is reached, no new incoming messages can be processed. Once an incoming message is confirmed, the total number of active sessions is decreased and another incoming message can be processed.
JMS Application Properties		Any application-specific message properties that are part of the message. This is specified by the JMS Application

Field	Global Var?	Description
Properties shared configuration object.		
Enable W3C SOAP JMS 1.0 Compliance	No	Select the check box to enable the W3C SOAP Over JMS 1.0 specification. By default, the check box is not selected and the service complies with the TIBCO specifications.
Target Service	No	This field is available only when the check box Enable W3C SOAP JMS 1.0 Compliance is selected. Specify the target service value, if any. Note: If Target Service is configured, it's value is reflected in the WSDL as a query parameter.

Messages With Attachments

SOAP messages can have message parts that contain attachments. For a message part to contain an attachment, specify the Special Type in the Type field of the Part Details section of the [Message](#) resource when creating a WSDL file configuration. For more information about creating WSDL files, see [WSDL Palette](#).

Depending on the SOAP version being used, messages with attachments can conform to one of the following styles:

- Messages with Attachments (SwA) - available with SOAP version 1.1 and 1.2.
- Message Transmission Optimization Mechanism (MTOM) - available with SOAP version 1.1 and 1.2.

Messages with Attachments (SwA)

SOAP clients that send messages with attachments in the SwA style must conform to the SOAP Messages with Attachments specification (<http://www.w3.org/TR/SOAP-attachments>).

The output schema element for a message part that contains an attachment; holds the content ID of the attachment instead of holding the actual attachment data. The actual data for attachments is always contained within the `mimeEnvelopeElement` of the output schema for this resource. This element contains a repeating element named `mimePart` that holds the list of attachments in the SOAP message. The attachment list can be correlated to the message parts containing the attachments by using the `content-id` `mimeHeader` element.

Message Transmission Optimization Mechanism (MTOM)

Message Transmission Optimization Mechanism (MTOM) provides another way of sending binary content or attachment processing by serializing SOAP messages with attachments. Optimization is only available for element content that is in a canonical lexical representation of `xs:base64Binary` data type. MTOM conforms to the specification <http://www.w3.org/TR/soap12-mtom>.

For an outbound SOAP Message with MTOM attachments sent by SOAP Request Reply or SOAP Send Reply activities, any element of type `xs:base64Binary` (or an extension of `xs:base64Binary`) in the SOAP response message is treated as an MTOM attachment and appears as a separate MIME part on the wire.

For an inbound SOAP Message with MTOM attachments received by the SOAP Event Source or SOAP Request Reply activities, the SOAP message have an XOP include reference to the attachment. Any XOP include reference in the SOAP message gets replaced by the corresponding attachment's content encoded in Base64. Any MIME attachment that is not referenced from the SOAP Message using XOP include reference gets ignored by the MTOM Processing layer.

Note that the attachment is a part of the SOAP Infoset.

For more information about MTOM, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Advanced

The Advanced tab contains the following fields:

Field	Description
General Sub-Tab	
Service URI	The Service URI portion of the URL that can be used to retrieve the WSDL file. For clients to retrieve WSDL files from ActiveMatrix BusinessWorks, you must define a process definition that accepts HTTP requests and uses the Retrieve Resources activity to generate the WSDL file. For more information, see Using Retrieve Resources on page 628. The URL to retrieve WSDL files is the following: <code>http://<host>:<port>/<serviceURI>?wsdl</code> where <code><serviceURI></code> is the value in this field.
SOAP Action	The soapAction that is expected from incoming SOAP requests. For more information about soapAction, see the SOAP specification.
WSDL Namespace	The namespace for the concrete WSDL file specified by this activity.
Style	The SOAP binding style. You can specify either Document or RPC style.
Encoding	The encoding type for the body of the SOAP input and output messages. This can be either literal or encoded. Encoded messages support more complex datatypes such as SOAP arrays. When encoded is specified, you can optionally specify the namespace for input, output, and fault messages.
Input Message Namespace	The namespace for input messages.
Output Message Namespace	The namespace for output messages.

Field	Description
Fault Message Namespace	The namespace for fault messages.
Embed Interface	When checked, specifies that all referenced WSDL files should be included inline in the concrete WSDL file. Otherwise, all referenced WSDL files are imported.
Embed Types	When checked, specifies that all referenced XSD files should be included inline in the concrete WSDL file. Otherwise, all referenced XSD files are imported.
Embed JNDI Properties	When checked, specifies that JNDI properties should be included inline in the service specification in the WSDL Source tab.

Input Attachments Binding Sub-Tab

Mime Content Type	The table containing this column allows you to specify the mime content type of each message part. This is especially useful for describing parts that contain attachments. For more information on the valid syntax of the content type, see the mime specification. If specified, the content type appears in the WSDL for the specified part. Use the X, +, and arrow keys to delete, add, and move parts in the list.
Message Part	The message part that you would like to specify the content type for. This field is a drop down list of message parts.
Write to File	Checking this field specifies that incoming messages whose attachments exceed the specified threshold size should be written to a file instead of stored in memory. This allows you to accept large incoming attachments without consuming a great deal of memory. When this field is checked, the Directory and Threshold Size fields appear.

Field	Description
	Leaving this field unchecked specifies that attachments of incoming messages should be kept in memory. Once written, the files created by using this option are not deleted automatically. You must manage the storage used by these files and delete them when they are no longer used.  Although Write to File exists for SwA and MTOM, Soap over JMS completely depends on the heap size configuration as JMS specifications do not support writing JMS messages in parts as is the case with Soap over HTTP, which can be written over wire in parts. Hence, writing large JMS messages to JMS after configuring Write to File, completely depends on the Heap size.
Directory	The directory to write attachments that are above the specified threshold. The process engine does not attempt to create the directory if the specified directory does not exist. Therefore, create the directory before starting the process engine.
Create Non-Existing Directories	When checked, all directories in the path specified in the Directory field are created, if they do not already exist. If this field is unchecked and there are one or more directories in the specified path in the Directory field that do not exist, an exception is raised.
Threshold Data Size (bytes)	The maximum size (in bytes) of an incoming message attachment that can be kept in memory. Attachments larger than the specified size are written to a file in the specified directory. The file's name is output so that subsequent activities in the process definition can access the file and read its contents. Specifying zero (0) in this field causes all incoming messages to be saved to a file.

Output Attachments Binding Sub-Tab

Attachment Style	This field allows you to choose the attachment style for the

Field	Description
	outbound/response messages. Two attachment styles are available: SwA and MTOM.
Mime Content Type	The table containing this column allows you to specify the mime content type of each message part. This is especially useful for parts that contain attachments. For more information on the valid syntax of the content type, see the mime specification. If specified, the content type appears in the WSDL for the specified part. Use the X, +, and arrow keys to delete, add, and move parts in the list.
Message Part	The message part that you would like to specify the content type for. This field is a drop down list of message parts.

Input Headers Sub-Tab

The Input Headers sub-tab allows you to specify a SOAP message parts used for any SOAP headers expected in the incoming request. The specified input headers are also included in the concrete bindings in the generated WSDL file.

When an incoming request supplies a header, the SOAP Event Source process starter places the information contained in the header into the output process variables for the SOAP Event Source. The header information is then available to subsequent activities in the process definition.

Use the +, X, and arrow keys to add, delete and move the input header messages in the list.

Message Name	The message in a WSDL file to use as the input header for the incoming SOAP request. This WSDL file does not have to be the same as the file used for output headers or the file used to configure the port type and operation of the incoming request. Use the Browse button in this field to search for valid WSDL File resources and to select the desired message in the WSDL File.

Field	Description
Part Name	This field contains a drop down list with all parts of the specified SOAP message. You can specify a namespace or encoding for the part, if applicable.
Binding	Binding used for the part. This field must be "soap".

Output Headers Sub-Tab

The Output Headers sub-tab allows you to specify a SOAP message parts used for any SOAP headers expected in the outgoing reply. The specified output headers are also included in the concrete bindings in the generated WSDL file.

Use the +, X, and arrow keys to add, delete and move the input header messages in the list.

Message Name	The message in a WSDL file to use as the output header for the SOAP request. This WSDL file does not have to be the same as the file used for input headers or the file used to configure the port type and operation of the incoming request. Use the Browse button in this field to search for valid WSDL File resources and to select the desired message in the WSDL File.
Part Name	This field contains a drop down list with all parts of the specified SOAP message. You can specify a namespace or encoding for the part, if applicable.
Binding	Binding used for the part. This field must be "soap".

WSDL Source

This tab displays the WSDL file that can be used to call the web service implemented by this process definition. This information is presented for display purposes only, and this file cannot be edited. You can copy this WSDL file and send it to anyone who wants to invoke the web service.

Normally, an application would use a web request to retrieve the WSDL file of a web service (for more information about sending a WSDL file as a response to a web

requestsee the description of the [Retrieve Resources](#) activity). This tab allows you to see the WSDL file that contains the concrete service description for this process.

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

Output

The output for the activity is the following.

Output Item	Datatype	Description
inputMessage or <operationName>	depends upon type specified in WSDL	The message parts specified in the WSDL file for the incoming SOAP request. The incoming request could contain multiple parts and attachments, and each part would be a message defined in the WSDL file. In case of MTOM attachments, the attachments

Output Item	Datatype	Description
		appear embedded in the message.
mimeEnvelopeElement	complex	This element contains the message attachments when MIME attachments with SwA style are used. This element contains a repeating element named mimePart that contains each MIME attachment.
mimeHeaders	complex	This element contains the mime header for each mimePart. Mime headers can contain the following information: • content-disposition • content-type • content-transfer-encoding • content-id • other mime header information Note: When the content type is specified as "text/*" (for example, "text/xml"), the attachment content is expected to be in either the textContent output element or the file name storing the attachment is expected to be in the fileName output element. When the content type is anything other than "text/*", the attachment content is expected to be in either the binaryContent output element or the file name storing the attachment is expected to be in the fileName output element.
binaryContent textContent fileName	choice	This element contains the mime attachment. The element can be one of the following: • binaryContent – content of the attachment when the attachment is binary

Output Item	Datatype	Description
		data. • textContent – content of the attachment when the attachment is text data. • fileName – the file name of the attachment when the Write to File field is checked on the Input Attachment Bindings sub-tab of the Advanced tab.
headers	complex	The header data supplied by the SOAP request. The structure of this input item is defined by the schema on the Input Headers sub-tab of the Advanced tab.
Header.<partName>	complex	Headers can contain multiple parts. The headers element contains all header parts supplied by the incoming SOAP request. These parts are specified on the Input Headers sub-tab of the Advanced tab. The header parts are stored in an element named Header.<partName> where <partName> is the name of the part specified on the Input Headers tab. Each Header.<partName> element also contains an attribute named @mustUnderstand. This boolean value is set to true in the incoming request if the client specifies that the server must understand the header part.
Context	complex	This element holds the optional SecurityContext element and the RemoteAddress and RemoteHost elements described below.
SecurityContext	complex	This output element is only available when the Expose Security Context field is checked on the Configuration tab. This element contains information about the requestor's authentication

Output Item	Datatype	Description
		or certificate information, depending upon the security scheme used. For more information, see the description of the following elements contained in the SecurityContext element.
Transport	complex	This element is only available when HTTP is the transport, this element is not available when JMS is the transport. This element contains the transport-level security context. This includes the following: • CertificateToken— complex element containing the subject, issuer, serial number, fingerprint, and X509 certificate. • UsernameToken – complex element containing the username, password, and nonce (random number, if one is set by the calling application).
WSSE	complex	This element contains the WSS policy security context. This includes the Authentication and Signature elements. See the description of these elements below.
Authentication	complex	This element contains the authentication information from the client. The type of information in this elements depends upon how the client authenticated. This element can contain one of the following: • UsernameToken – complex element containing the username, password, and nonce (random number, if one is set by the calling application). • KeyInfo – the client’s public key.

Output Item	Datatype	Description
Signature	complex	This element contains information from the user's digital signature. This element contains the following: - Value – this element contains the raw signature sent by the client. - Token – either a UsernameToken or a KeyInfo element. A UsernameToken element contains the username, password, and nonce, if available. A KeyInfo element contains information about the client's digital certificate or public key.
RemoteAddress	string	IP address of the remote host that sent the HTTP request. This element is only available when HTTP is the specified transport. This element is not available for the JMS transport.
RemoteHost	string	Host name of the remote host that sent the HTTP request. This element is only available when the Enable DNS Lookups field is checked on the Advanced tab of the HTTP Connection used by this activity. This element is only available when HTTP is the specified transport. This element is not available for the JMS transport.

TransportProperties - HTTP Transport

Method	string	The HTTP method to use for the request. All HTTP 1.1 methods are supported, but the CONNECT method is unnecessary because ActiveMatrix BusinessWorks automatically uses the correct method when connecting by way of

Output Item	Datatype	Description
		a proxy server. If no method is specified in this element, the GET method is used by default.
RequestURI	string	The address portion of the request. This is the portion of the URI before the question mark (?).
HTTPVersion	string	Version field of the HTTP request.
Command	string	Command portion of the HTTP request.
QueryString	string	The query string portion of the request. This is the portion after the question mark (?).
Header	complex	The header fields to send for the request. This element is specified on the Input Headers tab, and you can use this input item to supply values for the header when sending the request.
Protocol	string	Can be either HTTP or HTTPS depending upon the protocol used by the request.
Port	integer	The port number that the HTTP server uses for incoming requests. Specifying a value for this input item overrides any value specified on the Configuration tab.
Headers	Complex	
Accept	string	This field specifies media types that are acceptable for response messages for the incoming request. For example, <code>text/*,text/html</code> . Media types are described in the HTTP specification. If no Accept header field is present, then it is assumed that the client accepts all media types.

Output Item	Datatype	Description
Accept-Charset	string	This field specifies the character sets that are acceptable for response messages for the incoming request. For example, iso-8859-5, unicode-1-1. Character sets are described in the HTTP specification. If no Accept-Charset header is present, then it is assumed that the client accepts any character set.
Accept-Encoding	string	This field specifies the content-coding values that are acceptable for response messages. For example, compress, gzip. For more information about this header field, see the HTTP specification.
Content-Type	string	This field indicates the media type of the entity body sent to the receiver. Media types are described in the HTTP specification. An example of the media type is text/html; charset=ISO-8850-4.
Content-Length	string	This field indicates the size of the entity body (in decimal number of OCTETs) sent to the receiver and takes into account the encoding of the message body.
Connection	string	This field allows the requestor to specify options desired for this connection. For example, the option close specifies that the requestor would like the connection to be closed when the request is complete.
Cookie	string	For more information about this field, see the HTTP specification.
Pragma	string	This field is used to include implementation-

Output Item	Datatype	Description
		specific directives that might apply to the receiver. For more information about using this field, see the HTTP specification .

TransportProperties - JMS Transport

JMSHeaders

JMSDestination	string	Destination where a message is sent.
JMSReplyTo	string	The JMS application where the reply message should be sent.
JMSDeliveryMode	string	The delivery mode of the message. Can be one of the following: • Persistent: signifies the messages are stored and forwarded. • Non-Persistent: messages are not stored and may be lost due to failures in transmission. • TIBCO EMS Reliable: using TIBCO Enterprise Message Service, the JMSDeliveryMode header can be TIBCO EMS Reliable. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
JMSMessageID	string	The unique ID of the message.
JMSTimestamp	long	The time a message was handed off to a JMS provider to be sent. The time is expressed as the amount of time, in milliseconds, since midnight, January 1, 1970, UTC.

Output Item	Datatype	Description
JMSExpiration	long	The expiration time of the message. The time is expressed as the amount of time, in milliseconds since midnight, January 1, 1970, UTC. If set to 0, the message does not expire. For outgoing messages, this is set by specifying the JMSExpiration field on the activity's Advanced tab.
JMSRedelivered	boolean	Typically this item is set when the JMS provider has delivered the message at least once before.
JMSPriority	integer	The priority of the message. Priority is a value from 0-9. Higher numbers signify a higher priority (that is, 9 is a higher priority than 8). For outgoing messages, the default priority is 4. This is set on the activity's Advanced tab.
JMSCorrelationID	string	This ID is used to link a response message with its related request message. This is usually the message ID of a request message when this field is found in a reply message. For more information about JMSCorrelationID, see the following topics: • JMS Queue Requestor • JMS Queue Sender • JMS Topic Publisher • JMS Topic Requestor • Reply To JMS Message
JMSType	string	The JMS provider-supplied string to describe the type of the message. Some JMS providers

Output Item	Datatype	Description
		use this property to define messages in the provider's repository. For more information about the use of this field, see your JMS provider documentation.
JMSProperties		
JMSXUserID	string	The ID of the user sending the message. This property is optionally set on incoming messages by the JMS application. This property is not set on outgoing messages.
JMSXAppID	string	The ID of the application sending the message. This property is optionally set on incoming messages by the JMS application. This property is not set on outgoing messages.
JMSXProducerTXID	string	The transaction identifier of the transaction where this message was produced. This property is optionally set on incoming messages by the JMS application. This property is not set on outgoing messages.
JMSXConsumerTXID	string	The transaction identifier of the transaction where this message was consumed. This property is optionally set on incoming messages by the JMS application. This property is not set on outgoing messages.
JMSXRcvTimestamp	integer	The time the JMS server delivered the message to the consumer. The time is expressed as the amount of time, in milliseconds since midnight, January 1, 1970, UTC. This property is optionally set on incoming

Output Item	Datatype	Description
		messages by the JMS server. This property is not set on outgoing messages.
JMSXDeliveryCount	integer	The number of delivery attempts for this message.
JMSXGroupID	string	The identity of the message group this message is a part of. You can set this property on outgoing messages to group messages into a numbered sequence.
JMSXGroupSeq	integer	The sequence number of this message in its group. You can set this property on outgoing messages to group messages into a numbered sequence.

Error Output

The Error Output tab lists the following possible exceptions that can be thrown by this activity.

Activity Errors
InvalidEnvelopeException
InternalException
ConfigurationException
DeserializationException
ServerException

SOAP Request Reply

Activity



The **SOAP Request Reply** activity performs a request on the specified web service and optionally expects a reply from the web service. You can invoke both document and RPC web services with this activity.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Service	No	This two-part field specifies the namespace and the name of the web service. The service specified must be a concrete service that has a binding to an actual network endpoint. You can use the Browse button to bring up the Select a Resource dialog. This dialog allows you to select the available services and operations from the specified WSDL file. The Port, Operation, and Endpoint URL fields get populated by the choices made in this dialog.
Port	No	The port in the WSDL file containing the desired operation. The drop down list in this field is populated based on the specified service.
Operation	No	The operation that this activity invokes. This field is automatically populated when an operation is selected using the Select a Resource dialog (For more information, see the Service field).

Field	Global Var?	Description
SOAP Action	Yes	The soapAction header for the operation. For more information about soapAction, see the SOAP specification .
Timeout (sec)	Yes	The time to wait (in seconds) for the operation to complete. Note: You can also select Timeout as milliseconds (msec). Seconds (sec) is default.
Attachment Style	No	This field allows you to choose the attachment style. Two attachment styles are available: SwA and MTOM.

Reload Button

The **Reload** Button reloads the WSDL file. If the WSDL file has changed since it was last loaded, the configuration, input, output, and error output are updated with the changes.

Test Transport Connection

The **Test Transport Connection** button is only available when the transport is JMS. This button tests the connection to the JMS server.

Transport Details

The Transport Details tab allows you to specify additional configuration information based on the type of transport used for the SOAP request. When HTTP is the transport, the HTTP sub-tab appears on this tab. When JMS is the transport, the JNDI and JMS sub-tabs appear. For more information about SOAP over the JMS transport, see [SOAP Messages Over the JMS Transport](#).

The Transport Details tab contains the following fields:

Field	Global Var?	Description
HTTP Sub-Tab		
Endpoint URL	Yes	The URL of the web service that you are sending a request to. This field is automatically populated with the choice made when selecting the service. You can specify URLs that use the HTTPS (secure sockets) protocol. If you specify HTTPS as the protocol, the SSL field becomes enabled.
Use HTTP Proxy	No	When checked, this field specifies that an HTTP proxy server is used to connect to the SOAP server. Enabling this field causes the Proxy field to appear.
Proxy	No	Appears when the Use HTTP Proxy field is enabled. This field specifies the Proxy Configuration shared configuration resource that describes the HTTP proxy server. For more information, see Proxy Configuration .
HTTP Authentication	No	Specifies what authentication should be used . An authentication can be of the following three types: • NONE - is default. • BASIC - to continue with the current implementation of using username and password credentials. The Identity field appears when this field is enabled. • NTLM - Identity and Domain fields appear when NTLM is selected. Identity is a reference of the Username and Password combination.
Identity	No	This is an Identity resource that contains the client's username and password. This identity is used to perform only Basic and NTLM authentication.
Domain	No	Specifies the domain to be used for Authentication. The user name and password specified in the Identity field must

Field	Global Var?	Description

exist in the domain.

JNDI Sub-Tab

Context URL	Yes	This is the URL to the JNDI service provider (javax.naming.Context.PROVIDER_URL). The value for this field is taken from the WSDL file, if it is available in the <jndi:context> element in the service specification. You can override the supplied value, if you want to use a different JNDI service provider. For the syntax of the URL, see your JNDI provider documentation.
User Name	Yes	User name to use when logging into the JNDI server (javax.naming.Context.SECURITY_PRINCIPAL). If the JNDI provider does not require access control, this field can be empty.
Password	Yes	Password to use when logging into the JNDI server (javax.naming.Context.SECURITY_CREDENTIALS). If the JNDI provider does not require access control, this field can be empty.
Context Factory	Yes	The initial context factory class for accessing JNDI (javax.naming.Context.INITIAL_CONTEXT_FACTORY). The value for this field is taken from the WSDL file, if it is available in the <jndi:context> element in the service specification. You can override the supplied value, if you want to use a different context factory. Note: ActiveMatrix BusinessWorks attempts to find the class. However, you may need to add the Java file supplied by your JNDI service provider to the CLASSPATH environment variable to use JNDI.

Field	Global Var?	Description
Connection Factory	Yes	The QueueConnectionFactory object stored in JNDI. This object is used to create a queue connection with a JMS application. The value for this field is taken from the WSDL file, if it is available in the <jms:connectionFactory> element in the service specification. You can override the supplied value, if you want to use a different connection factory. For more information about creating and storing QueueConnectionFactory objects, see your JNDI provider documentation.
SSL	No	Specifies whether you want to use SSL for the connection to the JNDI server. SSL is used when the Use SSL? check box is selected. Click the Configure SSL button to configure the SSL connection parameters. For more information, see Configure SSL.
SSL Password	Yes	Password to use during the SSL connection.
Validate Security Context	Yes	Checking this field ensures that the security context provided on this tab is the one used to connect to the JNDI server. Some application servers store the security context on the thread used to establish the JNDI connection. Therefore the first activity to use this resource establishes the security context, and then subsequent activities use the same security context, unless this field is checked. Checking this field ensures that each activity that uses this resource examines the security context to determine if the activity uses the same security context as the security context established on the thread. If they are different, the activity's configured security context is used. Checking this field causes additional overhead for activities that use this resource. The only supported application server that requires this field to be checked is WebLogic.

Field	Global Var?	Description
Properties	No	Any additional properties to supply for the connection. You specify a name, datatype, and value for each property. These properties are typically vendor-specific. For more information about the available properties, see your JNDI provider documentation.
Note: Support for values given below for the JNDI Sub-tab is added for TIBCO ActiveMatrix BusinessWorks™ 5.16.1 HF-002 or above.		
OAuth Client ID	Yes	The Client ID issued to the client during the client registration process.
Client Secret	Yes	The Client Secret issued to the client during the client registration process.
Access Token URL	Yes	The endpoint for the authentication server. This is used to exchange the authorization code for the access token.
Grant Type	No	Client Credentials. Currently, only Client Credentials as a grant type is supported.
Authentication Type	No	Two types of authentication are supported: OAuth 2.0 and Basic Authentication. • If you select Basic Authentication, then the User Name and Password fields appear. • If you select OAuth 2.0, then the User Name, OAuth Client ID, Client Secret, Access Token URL and OAuth TLS fields appear.
OAuth TLS	No	Selecting this check box enables the support for generation access token from TLS based OAuth server when encrypting or decrypting Messages.

Field	Global Var?	Description
OAuth Disable Verify Hostname	No	Indicates whether the name on the server's certificate must be verified against the server's host name. If the server's host name is different from the name on the certificate, the TLS/SSL connection fails. The name on the certificate can be verified against another name by specifying OAuth Expected Hostname. Selecting this checkbox disables verification of OAuth Hostname. Deselecting this check box displays the OAuth Expected Hostname field. Default: This check box is selected.
Expected Hostname	Yes	Optional. The expected name of the remote host. The default is None.
Server Trust File	Yes	The location of the certificate. Following certificate extension supported in this File picker: pem crt cer cert

JMS Sub-Tab

User Name	Yes	User name to use to authenticate to the JMS server.
Password	Yes	Password to use to authenticate to the JMS server.
JMS Destination	Yes	Name of the JMS destination for the SOAP request. The syntax of the destination name is specific to the JMS provider you are using. For more information about destination names, see your JMS provider documentation.

Field	Global Var?	Description
		Note: If you are using TIBCO Enterprise Message Service as your JMS provider, you can use the Browse button next to this field after specifying a valid connection in the JMS Connection field. The Browse button displays a list of configured destinations in the JMS server that are appropriate for this activity.
Message Type	Yes	The type of the message. This can be one of the following: • Bytes Message – A stream of bytes. • Text Message – The message is a java.lang.String.
Delivery Mode	Yes	The delivery mode of the message. Can be one of the following: • PERSISTENT: signifies the messages are stored and forwarded. • NON-PERSISTENT: messages are not stored and may be lost due to failures in transmission. • TIBCO EMS Reliable: this mode is only available when using TIBCO Enterprise Message Service. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
Expiration	Yes	Corresponds to JMSEExpiration property that specifies how long the message can remain active (in seconds). If set to 0, the message does not expire. This field is set in seconds, but the JMSEExpiration property is displayed in milliseconds.
Priority	Yes	Priority of the message. You may set the priority to a value from 0-9. The default value is 4.
Use Shared JMS	No	Check this field if you want to specify a JMS Connection

Field	Global Var?	Description
Configuration		shared configuration resource to override the JMS transport parameters in the concrete WSDL. When checked, this JMS Connection field appears.
JMS Connection	No	This field only appears when the Use Shared JMS Configuration field is checked. This field allows you to specify a JMS Connection shared configuration resource that contains the connection information you want to use to override the JMS transport parameters in the concrete WSDL.
Enable W3C SOAP JMS 1.0 Compliance		This is a read-only field that displays whether the service is W3C SOAP Over JMS 1.0 compliant or not.
Target Service		This field is available only when the check box Enable W3C SOAP JMS 1.0 Compliance is selected. Specify the target service value, if any. Note: If Target Service is configured, the value is reflected in the WSDL as a query parameter.
Application Properties (Input Message)		Any application-specific message properties that are a part of the input message.
Application Properties (Output Message)		Any application-specific message properties that are a part of the output message.

Configure SSL

This button allows you to configure the SSL connection parameters for connecting to the SOAP server. This button brings up a dialog with the following fields:

Field	Description
Trusted Certificates Folder	Folder in the project containing one or more certificates from trusted certificate authorities. This folder is checked when connecting to a server to ensure that the connection is to a server that is trusted. This prevents connections to rogue servers that attempt to impersonate trusted servers.
Identity	This is an Identity resource that contains the client's digital certificate and private key. For more information, see <i>TIBCO Designer™ Palette Reference</i>.
Verify Host Name	This field specifies that the host name of the SOAP server should be checked against the host name listed in the server's digital certificate. This provides additional verification that the host name you believe you are connecting to is in fact the desired host. If the host name specified in the Endpoint URL field on the Configuration tab is not an exact match to the host name specified in the server's digital certificate, the connection is refused. Note: If you specify an equivalent hostname (for example, an IP address) in the Endpoint URL field, but the name is not an exact match of the hostname in the host's digital certificate, the connection is refused.
Strong Cipher Suites Only	When checked, this field specifies that the minimum strength of the cipher suites used can be specified with the <code>bw.plugin.security.strongcipher.minstrength</code> custom engine property. For more information about this property, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i>. The default value of the property disables cipher suites with an effective key length below 128 bits. When this field is unchecked, only cipher suites with an effective key length of up to 128 bits can be used.

Messages With Attachments

SOAP messages can have message parts that contain attachments. For a message part to contain an attachment, specify the Special Type in the Type field of the Part Details

section of the [Message](#) resource when creating a WSDL file configuration. For more information about creating WSDL files, see [WSDL Palette](#).

Depending on the SOAP version being used, messages with attachments can conform to one of the following styles:

- Messages with Attachments (SwA) - available with SOAP version 1.1 and 1.2.
- Message Transmission Optimization Mechanism (MTOM) - available with SOAP version 1.1 and 1.2.

Messages with Attachments (SwA)

SOAP clients that send messages with attachments in the SwA style must conform to the SOAP Messages with Attachments specification (<http://www.w3.org/TR/SOAP-attachments>).

The input or output schema element for a message part that contains an attachment holds the content ID of the attachment instead of holding the actual attachment data. The actual data for attachments is always contained within the `mimeEnvelopeElement` of the output schema for this resource. This element contains a repeating element named `mimePart` that holds the list of attachments in the SOAP message. The attachment list can be correlated to the message parts containing the attachments by using the content-id `mimeHeader` element.

Message Transmission Optimization Mechanism (MTOM)

Message Transmission Optimization Mechanism (MTOM) provides another way of sending binary content or attachment processing by serializing SOAP messages with attachments. Optimization is only available for element content that is in a canonical lexical representation of `xs:base64Binary` data type. MTOM conforms to the specification <http://www.w3.org/TR/soap12-mtom>.

Any element of type `xs:base64Binary` (or an extension of `xs:base64Binary`) in the SOAP response message is treated as an MTOM attachment and appears as a separate MIME part on the wire. The SOAP message have an XOP include reference to the attachment. Any XOP include reference in the SOAP message get replaced by the corresponding attachment's content encoded in Base64. Any MIME attachment that is not referenced from the SOAP Message using XOP include reference get ignored by the MTOM Processing layer. Note that the attachment is a part of the SOAP Infoset. For more information about MTOM and Write to File, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Overriding the Content Type MimeHeader

Currently, the ActiveMatrix BusinessWorks activities hard codes the Content Type mimeHeader with the value "application/octet-stream" while sending MTOM messages.

Now you can over ride the value of contentType mime header if the Content Type attribute (with namespace <http://www.w3.org/2005/05/xmlmime>) has been declared corresponding to a binary element in the message schema while sending MTOM messages

To enable this feature, add the Content Type attribute as explained below:

```
<xs:element name = "picture" xmlns:ns1 = "http://www.w3.org/2005/05/xmlmime">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base = "xs:base64Binary">
        <xs:attribute ref = "ns1:contentType"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

This feature is applicable for all activities in the **SOAP** and **Service Palette** that send outbound MTOM messages.

Advanced

The Advanced tab is used to specify the behavior of attachment for the reply message. This is useful if large messages are sent and you want to write them to files instead of storing them in memory. The Advanced tab has the following fields.

Field	Description
Write to File	Checking this field specifies that incoming messages whose attachments exceed the specified threshold size should be written to a file instead of stored in memory. This allows you to accept large incoming attachments without consuming a great deal of memory. When this field is checked, the Directory and Threshold Size fields appear. Leaving this field unchecked specifies that attachments of incoming messages should be kept in memory.

Field	Description
	Note: Once written, the files created by using this option are not deleted automatically. You must manage the storage used by these files and delete them when they are no longer used.  Although Write to File exists for SwA and MTOM, Soap over JMS completely depends on the heap size configuration as JMS specifications do not support writing JMS messages in parts as is the case with Soap over HTTP, which can be written over wire in parts. Hence, writing large JMS messages to JMS after configuring Write to File, completely depends on the Heap size.
Directory	The directory to write attachments that are above the specified threshold. The process engine does not attempt to create the directory if the specified directory does not exist. Therefore, create the directory before starting the process engine.
Create Non-Existing Directories	When checked, all directories in the path specified in the Directory field are created, if they do not already exist. If this field is unchecked and there are one or more directories in the specified path in the Directory field that do not exist, an exception is raised.
Threshold Size (bytes)	The maximum size (in bytes) of an incoming message attachment that can be kept in memory. Attachments larger than the specified size are written to a file in the specified directory. The file's name is output so that subsequent activities in the process definition can access the file and read its contents. Specifying zero (0) in this field causes all incoming messages to be saved to a file.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
inputMessage or <operationName>	complex	The root class for the input of the SOAP Request Reply activity. This class contains all input items for the activity. If this is a document request, the input is named inputMessage. If the request is an RPC, the input is the name of the operation.
_configData	complex	Specifies the values to use to override transport-specific configuration values, if desired. The content of the _configData element depends upon the type of transport used.
endpointURL	string	Available only when HTTP is the transport. Specifies the Endpoint URL of the service. Specifying a value in this input item overrides any value specified in the Endpoint URL field on the Configuration tab.
host	string	Available only when HTTP is the transport. Specifies the host machine where the SOAP service resides as host:port. This should be used when sending a message through a tunnel.

Transport Properties - HTTP

Headers

Accept	string	This field specifies media types that are acceptable for response messages for the request. For example, text/*, text/html. Media types are described in the HTTP specification. If no Accept header field is specified, then the server assumes that all media types are acceptable.

Input Item	Datatype	Description
Accept-Charset	string	This field specifies the character sets that are acceptable for response messages for the request. For example, iso-8859-5, unicode-1-1. Character sets are described in the HTTP specification. If no Accept-Charset header is specified, then the server assumes that any character set is acceptable.
Accept-Encoding	string	This field specifies the content-coding values that are acceptable for response messages. For example, compress, gzip. For more information about this header, see the HTTP specification.
Cookie	string	A name/value pair (also known as a cookie) containing information that the HTTP server may be expecting. You can set the Cardinality field for this element to Repeating (*) to specify more than one cookie. You can also specify multiple name/value pairs in a single non-repeating element by separating each pair with a comma (for example, "name1=value1, name2=value2"). You set the custom engine property <code>bw.plugin.http.protocol.single-cookie-header</code> to true, multiple name/value pairs are sent as a single header element. For more information about custom engine properties, see <i>TIBCO ActiveMatrix BusinessWorks™ Administration</i>.
Pragma	string	This field is used to include implementation-specific directives that might apply to the receiver. For more information about using this field, see the HTTP specification.

Input Item	Datatype	Description
Authentication Credentials		
username	string	The username for logging into the HTTP server. This overrides the value specified in the Identity resource. Note: You can override the username only if the activity is configured for Basic or NTLM authentication.
password	string	The password for logging into the HTTP server. This overrides the value specified in the Identity resource. Note: You can override the password only if the activity is configured for Basic or NTLM authentication.
domain	string	Specify the domain to be used. This overrides the value specified in the configuration. Note: You can override the domain name only if the activity is configured for NTLM authentication.
soapAction	string	Specifies the soapAction header for the operation. For more information about soapAction, see the SOAP specification. Specifying a value in this input item overrides any value specified in the Action field on the Configuration tab.
timeout	number	The time to wait (in seconds) for the operation

Input Item	Datatype	Description
		to complete.
mimeEnvelopeElement	complex	This element contains the message attachments when mime attachments are used. This element contains a repeating element named mimePart that contains each mime attachment.
mimepart		
mimeHeaders	complex	This element contains the mime header for each mimePart. Mime headers can contain the following information: • content-disposition • content-type • content-transfer-encoding • content-id • other mime header information Note: When the content type is specified as "text/*" (for example, "text/xml"), the attachment content is expected to be in either the textContent input element or the file name storing the attachment is expected to be in the fileName input element. When the content type is anything other than "text/*", the attachment content is expected to be in either the binaryContent input element or the file name storing the attachment is expected to be in the fileName input element.
binaryContent textContent fileName	choice	This element contains the mime attachment. The element can be one of the following:

Input Item	Datatype	Description
		• <code>binaryContent</code> – content of the attachment when the attachment is binary data. • <code>textContent</code> – content of the attachment when the attachment is text data. • <code>fileName</code> – the file name of the attachment when the attachment is stored in the file system.

Transport Properties - JMS

JMSTargetDestination	string	Available only when JMS is the transport. Specifies the JMS destination (queue) name used to send messages.
JMSReplyDestination	string	Available only when JMS is the transport. Specifies the JMS destination (queue) name used to send the response. The JMSReplyDestination specified has to be unique for every message. Set the JMSReplyDestination to a non-static variable such as Job-Id to use the same JMSReplyDestination multiple times.
JMSCorrelationID		This ID is used to link a response message with its related request message. This property is usually set to the message ID of the message you are replying to, but any value can be used. Note: This is applicable only when JMSReplyDestination is configured with SOAP over JMS. The JMSCorrelationID of the reply message is set as follows:

Input Item	Datatype	Description
		- When the JMSCorrelationID field is configured with a unique value, the JMSCorrelationID property of the request message is set with the unique value. The same value is also set as the JMSCorrelationID of the corresponding response message. - When the JMSCorrelationID input element is not configured, the value of the message ID of the JMS Request message is set as the JMSCorrelationID property of the response message.
JMSPriority	number	Available only when JMS is the transport. Specifies the priority of the message.
JMSExpiration	number	Available only when JMS is the transport. Corresponds to JMSExpiration property that specifies how long the message can remain active (in seconds).

Output

The output for the activity is the following.

Output Item	Datatype	Description
outputMessage or <operationName>Response	complex	The root class for the output of the SOAP Request Reply activity. This class contains all output items for the activity. If this is a document request, the output is named outputMessage. If the request is an RPC, the output is the name of the operation with "Response" appended.

Output Item	Datatype	Description
mimeEnvelopeElement	complex	This element contains the reply message attachments when mime attachments are used. This element contains a repeating element named mimePart that contains each mime attachment.
mimeHeaders	complex	This element contains the mime header for each mimePart. Mime headers can contain the following information: • content-disposition • content-type • content-transfer-encoding • content-id • other mime header information Note: When the content type is specified as "text/*" (for example, "text/xml"), the attachment content is expected to be in either the textContent output element or the file name storing the attachment is expected to be in the fileName output element. When the content type is anything other than "text/*", the attachment content is expected to be in either the binaryContent output element or the file name storing the attachment is expected to be in the fileName output element.
headers	complex	Any output headers defined in the WSDL file. The headers are stored in an element named Header.<partName> where <partName> is the element name of the output header schema. The Header.<partName> element also contains an item named @mustUnderstand. This boolean

Output Item	Datatype	Description
		value is set to true if the server specifies that the SOAP client must understand the header.
binaryContent textContent fileName	choice	This element contains the mime attachment. The element can be one of the following: • binaryContent – content of the attachment when the attachment is binary data. • textContent – content of the attachment when the attachment is text data. • fileName – the file name of the attachment when the Write to File field is checked on the Output Attachment Bindings tab.

Transport Properties - HTTP

Headers

Allow	string	This field lists the set of methods supported by the resource identified by RequestURI.
Content-Type	string	This field indicates the media type of the entity body for the outgoing message and the incoming response. Media types are described in the HTTP specification. An example of the media type is text/html; charset=ISO-8850-4.
Content-Length	string	This field indicates the size of the entity body (in decimal number of OCTETs) of the response message.
Content-Encoding	string	This field is used as a modifier to the content-type. When present, its value indicates what additional content encodings have been applied

Output Item	Datatype	Description
		to the entity-body, and thus what decoding mechanisms must be applied in order to obtain the media-type referenced by the Content-Type header field. Content-Encoding is primarily used to allow a document to be compressed without losing the identity of its underlying media type. For more information about this field, see the HTTP specification.
Date	string	The date and time when the response message was sent.
Location	string	This field was used to redirect the receiver to a location other than the Request-URI for completion of the request or for identification of a new resource.
Set-Cookie	string	For more information about this field, see the HTTP specification.
Pragma	string	This field is used to include implementation-specific directives that might apply to the receiver. For more information about using this field, see the HTTP specification.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see TIBCO ActiveMatrix BusinessWorks™ Error Codes.

Exception	Thrown When...
defaultFaultElement	The server returned a default SOAP fault.

Exception	Thrown When...
ActivityTimeoutException	The activity reached the specified timeout and a response has not been received from the server.
<customFault>	The SOAP server has returned a custom fault. These faults are defined in the WSDL file and are listed as a choice element.

Errors when HTTP is the transport

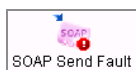
HttpClientException	The SOAP server has reported a client error.
HttpServerException	The SOAP server has reported an internal error.
HttpCommunicationException	An HTTP communications error occurred.

Errors when JMS is the transport

JMSMessageCreateException	An error occurred when creating the request message.
JMSSessionCreateException	An error occurred when creating the JMS session to send the request message.
JMSSendException	An error occurred when attempting to send the JMS message.
JMSReceiveException	An error occurred when attempting to receive the JMS reply message.

SOAP Send Fault

Activity



The **SOAP Send Fault** activity sends a SOAP fault to the client if an error occurs during processing of a SOAP request.

This activity is used when the SOAP Event Source process starter handles incoming SOAP requests. The WSDL file for the request can have one or more fault messages defined for an operation (For a description of creating fault schemas for operations, see

[WSDL Palette](#)). The **SOAP Send Fault** activity can use the fault schema to send application-specific data about the fault back to the client that made the SOAP request.

 **Note:** This activity cannot be used if the SOAP request is one way.

SOAP faults are required to send the fault code and the fault string. These values are required input items on the Input tab. For example, "Client" can be used for client faults and "Server" can be used for server faults. For more information about the syntax of SOAP fault codes, see the SOAP specification at www.w3.org/TR/SOAP.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Reply To	No	The SOAP Event Source process starter.
Fault Details	No	The name of the fault message to send. This field provides a drop down list of the fault messages defined in the WSDL file specified for the SOAP Event Source process starter. You must select one of the fault messages to return to the client, and that message becomes part of the input schema for this activity's Input tab. The option <Default> is always available and can be used to send the default fault message.

Transport

The Transport tab has the following fields.

Field	Global Var?	Description
JMS Transport		
JMS Delivery Mode	No	The delivery mode of the message. Can be one of the following: • PERSISTENT: signifies the messages are stored and forwarded. • NON-PERSISTENT: messages are not stored and may be lost due to failures in transmission. • TIBCO EMS Reliable: this mode is only available when using TIBCO Enterprise Message Service. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
JMS Application Properties	No	Any application-specific message properties that are part of the message. This is specified by the JMS Application Properties shared configuration resource.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

ActiveMatrix BusinessWorks uses the same schema for Fault messages regardless of the version of the SOAP protocol. For more information, see [SOAP Specification Compliance](#).

The input for the activity is the following.

Input Item	Datatype	Description
faultcode	string	The SOAP specification defines a specific syntax for faultcodes returned by a SOAP fault. For more information on how to specify a faultcode, see the SOAP specification.

Input Item	Datatype	Description
faultstring	string	The faultstring is a string message to pass back to the client describing the fault.
faultactor	string	The faultactor element is intended to provide information about who caused the fault to happen in the message path. It is similar to the SOAP actor attribute but instead of indicating the destination of the header entry, it indicates the source of the fault. The value of the faultactor attribute is a URI identifying the source of the fault. You may use the faultactor element to indicate explicitly that this ActiveMatrix BusinessWorks process generated the fault.
detail	complex	The fault message specified by the Fault Details field on the Configuration tab. This message is used to pass application-specific data back to the client in the event of a fault.
_configData	complex	This element is available only for a JMS transport. Specifies the values to use to override transport-specific configuration values. The content of the _configData element depends upon the type of transport used.

Output

The activity has no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
HTTP Transport	
ServerException	An error occurred and the reply could not be sent to the client.
JMS Transport	
JMSMessageCreateException	The JMS message could not be created.
JMSSessionCreateException	The JMS session could not be created.
JMSSendException	The JMS send operation failed.

SOAP Send Reply

Activity



The **SOAP Send Reply** activity sends a reply to an application that sent a SOAP request. This activity is primarily used in process definitions that implement web services. When a [SOAP Event Source](#) is used as the process starter, the **SOAP Send Reply** activity is used to send the reply to the request that starts the process instance.



Note: This activity cannot be used if the SOAP request is one way

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process

Field	Global Var?	Description
		definition.
Description	No	Short description of the activity.
Reply To	No	The SOAP Event Source process starter that received the request.

Transport

The Transport tab has the following fields.

Field	Global Var?	Description
JMS Transport		
Delivery Mode	No	The delivery mode of the message. Can be one of the following: • PERSISTENT: signifies the messages are stored and forwarded. • NON-PERSISTENT: messages are not stored and may be lost due to failures in transmission. • TIBCO EMS Reliable : this mode is only available when using TIBCO Enterprise Message Service. For more information about this mode, see the TIBCO Enterprise Message Service documentation.
JMS Application Properties	No	Any application-specific message properties that are part of the message. This is specified by the JMS Application Properties shared configuration resource.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
outputMessage or <operationName>		The output schema of the operation for the selected SOAP request. Each of the schema elements are available in this dialog for mapping.
Header.<headerPart>	complex	Headers can contain multiple parts. The headers element contains all header parts required in a SOAP reply. These parts are specified on the Output Headers tab of the SOAP Event Source activity. The header parts are stored in an element named Header.<partName> where <schema-name> is the name of the schema specified on the Output Headers tab. Each Header.<partName> element also contains an item named @mustUnderstand. This boolean value is set to true in the outgoing reply if you require that the client must understand the header part.
mimeEnvelopeElement	complex	This element contains the reply message attachments when mime attachments are used. This element contains a repeating element named mimePart that contains each mime attachment.
_configData	complex	This element is available only for a JMS transport. Specifies the values to use to override transport-specific configuration values, if

Input Item	Datatype	Description
		desired. The content of the <code>_configData</code> element depends upon the type of transport used.

Output

The activity has no output.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
HTTP Transport	
ServerException	An error occurred and the reply could not be sent to the client.
JMS Transport	
JMSMessageCreateException	The JMS message could not be created.
JMSSessionCreateException	The JMS session could not be created.
JMSSendException	The JMS send operation failed.

SOAP Messages Over the JMS Transport

The HTTP transport does not provide guaranteed message delivery. The SOAP protocol is not specific to any transport, so SOAP messages in theory can be bound to any

transport. JMS provides a highly reliable transport that can guarantee message delivery. ActiveMatrix BusinessWorks supports sending and receiving SOAP messages over the HTTP and JMS transports.

As of Release 5.3.0, there is no standard specification for SOAP messages over JMS. TIBCO has proposed a specification with the following features:

- One-way and request/response operations
- The support is based on SOAP 1.1, SOAP 1.2, and WSDL 1.1

What's in the JMS Message

The following sections describe the content of the JMS message containing a SOAP request.

JMS Header

[JMS Message Headers and Properties for SOAP messages](#) describes the JMS Message Headers and Properties.

JMS Message Headers and Properties for SOAP messages

JMS Message Headers and Properties	Description
TIBCO Compliant	
Content_Type	This header must be one of the following: • application/xml for SOAP 1.1 or SOAP 1.2 without attachments • multipart/related for SOAP 1.1 or SOAP 1.2 Note: Content type parameters are also allowed, for example, charset=utf-8.
Mime_Version	Must be the string "1.0".
SoapAction	SOAPAction value as defined in the WSDL.

JMS Message Headers and Properties	Description
JMSDestination	This contains the destination name.
JMSReplyTo	This contains the reply to address to be used by service provider to return data. Currently, only temporary queues can be used for reply messages.
JMSMessageID	This contains a message ID generated by the JMS provider.

W3C SOAP Over JMS 1.0 Specification

SOAPJMS_targetService	This contains the Target Service value as defined in the WSDL.
SOAPJMS_bindingVersion	Must be the string "1.0"
SOAPJMS_contentType	This contains the format of the message sent.
SOAPJMS_soapAction	This contains the SOAPJMS_soapAction value as defined in the WSDL
SOAPJMS_isFault(boolean)	This contains the value of the SOAP JMS Fault.
SOAPJMS_requestURI	This contains the value of the location.

JMS Body

The JMS message body contains a binary representation of the actual SOAP message. Below is an example of a SOAP message.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
  "http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body SOAP-ENV:encodingStyle=
    "http://schemas.xmlsoap.org/soap/encoding/">
    <ns0:getCatalog xmlns:xs=
      "http://www.w3.org/2001/XMLSchema"
```

```

xmlns:soap-enc=
  "http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns0="http://InputMessageNamespace"/>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

WSDL Extensions

The following sections describe extensions to the WSDL standard to support the JMS protocol.

WSDL Binding

TIBCO Standard

The WSDL Binding for SOAP Over JMS varies only slightly from the SOAP Over HTTP binding. The transport attribute under soap:binding indicates that JMS is being used. The attribute must have the value: `http://www.tibco.com/namespaces/ws/2004/soap/binding/JMS`.

For example:

```

<soap:binding style="document"
  transport="http://www.tibco.com/namespaces/ws/2004/soap/binding/JMS"/>

```

In addition to the soap:binding transport information, there must be a JMS binding element that describe the message type used for transmission. Currently, the only supported message types are `ByteMessage` and `TextMessage`. This looks like the following:

```

<jms:binding messageFormat="bytes"/>

```

An example of the complete WSDL binding section is show below.

```

<wsdl:binding name="RetailJMSBinding" type="tns:RetailerPortType">
  <soap:binding style="document" transport=
    "http://www.tibco.com/namespaces/ws/2004/soap/binding/JMS"/>
  <jms:binding messageFormat="bytes"/>
  <wsdl:operation name="getCatalog">
    <wsdl:documentation>The operation has no documentation</wsdl:documentation>
    <soap:operation style="document" soapAction="RetailGetCatalog"/>
  </wsdl:operation>
</wsdl:binding>

```

```

<wsdl:input>
  <soap:body use="encoded"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    namespace="http://InputMessageNamespace" parts=""/>
</wsdl:input>
<wsdl:output>
  <soap:body use="encoded"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    namespace="http://OutputMessageNamespace" parts="return"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>

```

W3C SOAP Over JMS 1.0 Compliance

When the service is W3C SOAP Over JMS 1.0 compliant, the transport attribute under soap:binding has the value : <http://www.w3.org/2010/soapjms/>.

For example:

```

<soap:binding style="rpc" transport="http://www.w3.org/2010/soapjms"/>

```

Additionally the JNDI namespace in WSDL is also <http://www.w3.org/2010/soapjms/>

An example of the complete WSDL binding section with W3C SOAP Over JMS 1.0 specification is as follows.

```

<wsdl:binding name="RetailJMSBinding" type="tns:RetailerPortType">
  <soap:binding style="document" transport=
    "http://www.w3.org/2010/soapjms"/>
  <jms:binding messageFormat="bytes"/>
  <wsdl:operation name="getCatalog">
    <wsdl:documentation>The operation has no documentation</wsdl:documentation>
    <soap:operation style="document" soapAction="RetailGetCatalog"/>
    <wsdl:input>
      <soap:body use="encoded"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://InputMessageNamespace" parts=""/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://OutputMessageNamespace" parts="return"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

```
</wsdl:operation>
</wsdl:binding>
```

WSDL Service

TIBCO Standard

The service section of the WSDL also contains extensions to support SOAP over JMS. The extension is defined with the following form where '?' means optional.

```
<wsdl:service>
  <jndi:context />?
  <jms:connectionFactory />?
  <wsdl:port>
    <jndi:context />?
    <jms:connectionFactory />?
    <jms:targetAddress />
  </wsdl:port>
</wsdl:service>
```

In this definition, both `jndi:context` and `jms:connectionFactory` elements can possibly appear at two locations and they must appear at least once. When they appear under `wsdl:service`, the definition applies to all ports inside the service. When they appear under `wsdl:port`, the definition applies to only that port and it overrides the definitions that appear at the service level if it exists. If they do not appear at the service level, then each port must define them.

The `jms:connectionFactory` element is the JMS connection factory name that is used to connect with JMS service, using the JNDI context supplied in the `jndi:context` element. Below is an example of using TIBCO Enterprise Message Service as the JMS provider.

```
<jms:connectionFactory>QueueConnectionFactory
</jms:connectionFactory>
```

In a simple configuration, the `jms:connectionFactory` is the same for both the Service Provider and the Service Clients. However, both provider and clients can use different connection factories, provided that they all create connections to the same JMS provider instance.

The `jms:targetAddress` element contains address information which the service provider is going to listen for messages. Below is an example of using TIBCO Enterprise Message Service as the JMS provider with a queue name of MyQueue.

```
<jms:targetAddress destination="queue">MyQueue</jms:targetAddress>
```

The `jndi:context` element holds one or more `jndi:property` elements. Each of the `jndi:property` elements describes a JNDI name/value pair. For example,

```
<jndi:context>
  <jndi:property name="java.naming.factory.initial"
    type="java.lang.String">
    com.tibco.tibjms.naming.TibjmsInitialContextFactory
  </jndi:property>
  <jndi:property name="java.naming.provider.url"
    type="java.lang.String">
    tibjmsnaming://localhost:7222
  </jndi:property>
</jndi:context>
```

MimePar

W3C SOAP Over JMS 1.0 Complaint

With W3C Compliance the `jndi:property` elements described in the JNDI name and pair value are displayed as :

```
<jms:jndiInitialContextFactory>com.tibco.tibjms.naming.TibjmsInitialContextFactory</jms:jndiInitialContextFactory>
<jms:jndiURL>tibjmsnaming://localhost:7222</jms:jndiURL>
```

Query Parameters

With W3C SOAP Over JMS 1.0 compliance, property names are displayed as query parameters with the value of `SOAPJMS_requestURI` within the WSDL (attribute location and under `<soap:address>`) binding . They are supported in the client activities when referring to concrete WSDLs.

Property Names	Description
deliveryMode	JMS property for sending message
Priority	JMS property for sending message
timeToLive	JMS property for sending message
jndiConnectionFactoryName	Jndi property for connection factory name
jndiInitialContextFactory	Jndi property for Initial Context Factory name
jndiURL	Jndi property for JNDI connection
targetService	Property to dispatch the request

Fault Codes

Additional fault codes are introduced with the W3C SOAP Over JMS 1.0 compliance. The following table lists the fault codes.

Error Fault	Condition
mismatchedSoapAction	If using SOAP 1.2, and the SOAPJMS_contentType property has an action parameter, and its value does not match with SOAPJMS_soapAction value.
missingContentType	If SOAPJMS_contentType property is missing.
missingRequestURI	If SOAPJMS_requestURI property is missing.
missingTargetService	If mentioned in WSDL and SOAPJMS_targetService property is missing.
targetServiceNotAllowedInRequestURI	If targetService parameter is included in the SOAPJMS_requestURI.
unrecognizedBindingVersion	if the value of the SOAPJMS_bindingVersion property does not match the fixed value 1.0.

MimeParser

Activity



This activity is responsible for parsing the MTOM messages into a SOAP message that contains the binary attachments as an inline data of the SOAP message. The output can be either in a Text or a Binary format.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Input Style	No	The Input message format. (Currently only MTOM is available)
Output Style	No	The Output message format. You can choose either Text or Binary format for the output messages.

Advanced

The Advanced tab contains the following field:

Field	Global Var?	Description
GenerateStrictParsingErrors	No	When checked, this field generates strict parsing validation errors.

Input

The Input tab contains the following fields:

Input Item	Datatype	Description
MimeBinaryData	Binary	Incoming binary data

Output

The Output for the activity is the following.

Output Item	Datatype	Description
SOAPOutput		
soapMessage	Binary or String	Depends upon the type specified in the Output Style format
StrictValidationErrors	Complex	This appears only when the GenerateStrictParsingErrors check box is selected.
ErrorString	String	Shows strict parsing validation error logs.

Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity.

Exception	Thrown When...
XOPParseException	Specifies XOP related exceptions found in the input.
MimeParseException	Specifies MIME related exceptions found in the input.
InvalidXMLException	Specifies invalid xml related exception found in the input.

TCP/IP Palette

This section describes the resources in the **TCP/IP** palette.

Overview of the TCP/IP Palette

The **TCP/IP** palette provides activities and resources that can send and receive data using the TCP/IP protocol. TCP/IP is often the underlying communication layer for higher-level protocols such as HTTP or FTP. You could use the resources in the **TCP** palette to communicate with HTTP or FTP servers, but the activities in the [HTTP Palette](#) or [FTP Palette](#) are best suited for this purpose. This palette is intended for use when communicating with TCP/IP and there is no corresponding ActiveMatrix BusinessWorks palette for the protocol. For example, you may have a custom application that communicates with other applications by way of TCP/IP. You could use the **TCP/IP** palette to handle incoming and outgoing data from this application.

Using the TCP/IP Palette Activities

The general sequence of events when using the TCP/IP palette resources is the following:

Procedure

1. Create a TCP Connection shared configuration resource that describes the connection information.
2. One of the following occurs depending upon whether the process definition is a TCP/IP client or server:

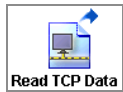
Procedure

- a. When a process definition acts as a TCP/IP client communicating with a TCP/IP server, a connection must be opened using the [TCP Open Connection](#) activity.

- b. When a process definition acts as a TCP/IP server waiting for incoming TCP/IP requests, the process either starts with a [TCP Receiver](#) process starter or has a [Wait for TCP Request](#) activity that waits for an incoming connection request.
3. In the process definition the [Read TCP Data](#) and [Write TCP Data](#) activities are used to read and write data to the TCP connection.
4. Once all processing is complete, the [TCP Close Connection](#) activity is used to close the TCP connection.

Read TCP Data

Activity



The **Read TCP Data** activity reads data from an open TCP connection. The connection must be opened by an activity or process starter that executed previously in the process definition. Activities that can open a TCP connection are: [TCP Receiver](#), [TCP Open Connection](#), and [Wait for TCP Request](#). The activity that opens a TCP connection places a handle to the open connection in its connectionKey output element. The connectionKey is mapped to this activity's input.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Type	No	Type of data to be read. This can be either Text or Binary. When Text is specified, the Separator, User Defined Separator, and Encoding fields appear. When Binary is specified, the Bytes to Read field appears.

Field	Global Var?	Description
Separator	Yes	The type of separator. This activity reads data from the TCP connection until the separator or the end of the data stream is encountered.
User Defined Separator	Yes	When User-defined separator is chosen from the list in the Separator field, this field appears. This field specifies the separator character to use. When Binary is selected in the Type field, a user-defined separator can be a string representation of a single byte value or it can be comma-separated byte values (for example, 0x0a, 0x0b, 0x0c). For valid string representations of byte values, refer to the Java documentation for the <code>decode(String n)</code> method of <code>java.lang.Byte</code>.
Encoding	Yes	When text data is specified, this field determines the character encoding of the text data.
Bytes to Read	Yes	When binary data is specified, this field specifies the number of bytes that should be read by this activity. If this field is left blank, then the activity reads until the end of the data stream is encountered.
Timeout	Yes	The time to wait (in milliseconds) for this activity to complete.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
connection	binary	The handle to the connection that you want to read data

Input Item	Datatype	Description
		from. This connection is obtained from a previously executed activity in the process definition that opened the connection.
encoding	String	When text data is being read, this specifies the character encoding of the text. This value overrides the value specified in the Encoding field on the Configuration tab.
separator	String	When text data is being read, this specifies the separator to write after the body element is written to the TCP connection. This value overrides the value specified on the Configuration tab.
bytesToRead	number	When binary data is being read, this specifies the number of bytes that should be read. This value overrides the value specified in the Bytes to Read field on the Configuration tab.
timeout	number	Time to wait (in milliseconds) for this activity to complete.

Output

The output for the activity is the following.

Output Item	Datatype	Description
data	String	When text data is specified, this output element contains the text read from the TCP connection. The separator character, if specified, is not included in the output.
binaryData	binary	When binary data is specified, this output element contains the binary data read from the TCP connection.

Output Item	Datatype	Description
endOfStreamReached	boolean	This element is set to true when the end of the data stream has been reached. This is useful when the activity is in an iteration loop and the activity is configured to read a number of bytes with each iteration. This element can be used in the condition of the loop to determine when the last set of bytes has been read.

TCP Close Connection

Activity



The **TCP Close Connection** activity closes a TCP connection opened by a previously executed activity or process starter. Activities that can open a TCP connection are: [TCP Receiver](#), [TCP Open Connection](#), and [Wait for TCP Request](#). The activity that opens a TCP connection places a handle to the open connection in its connectionKey output element. The connectionKey is mapped to this activity's input.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
connection	binary	The handle to the connection that you want to close. This connection is obtained from a previously executed activity in the process definition that opened the connection.

Output

The activity has no output.

TCP Connection

Shared Configuration



The TCP Connection is a shared configuration resource that specifies the connection information for the TCP server. This resource is used when a process definition acts as a TCP client connecting to a remote server or when a process definition acts as a TCP server accepting incoming TCP connections.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.

Field	Global Var?	Description
Description	No	Short description of the activity.
Host	No	When a process definition acts as a client, this field specifies the host name or host IP address of the TCP server to connect to. When a process definition acts as a server, this field specifies the hostname or IP address of the machine where the process engine is running. You can specify "localhost" or if the machine has more than one network interface card, you can specify the IP address of the card you want to use to accept TCP/IP connections.
Port	Yes	This field specifies the port number on which a TCP server is listening for requests.
Enable Pooling (client side)	No	Specifies whether connection pooling should be enabled on the client side. Selecting this check box enables the WhenExhaustedAction, Maximum Connections, maxwaitTime and idleTimeout fields. By default, connection pooling is disabled. For more information, see Connection Pooling.
WhenExhaustedAction	No	Specify the action to be taken when the maximum number of connections is consumed and the connection pool is exhausted: • Fail:When the pool is exhausted, the TCP Open Connection fails and an exception is thrown. • Grow:When the pool is exhausted, a new connection is created. There is no limit to the number of TCP

Field	Global Var?	Description
		connections in the pool. Block:When the pool is exhausted, the TCP Open Connection activity is blocked until a new connection is available, or till the maximum wait time is reached. The default value is Block.
Maximum Connection	Yes	This field is available when the action Fail or Block is selected in the field WhenExhaustedAction. Specifies the maximum number of TCP Connections in a pool. The default is 10.
maxWaitTime(in ms)	Yes	This field is available when the Block action is selected in the field WhenExhaustedAction. Specifies the maximum amount of time the TCP Open Connection activity must wait till it gets a connection from the pool. When less than or equal to 0, it may be blocked indefinitely. The default is 10 seconds.
idleTimeout (in ms)	Yes	Specifies the minimum amount of time a TCP Connection may sit idle in the pool before it is eligible for eviction. The default is -1. When less than or equal to 0 the eviction is disabled.

Advanced

The Advanced tab has the following fields

Field	Global Var?	Description
enableDNSLookup	No	Selecting this field enables a Domain Name System (DNS) lookup so that the IP address is resolved to a DNS name. This can adversely affect the throughput, so enable this field only when required.

Connection Pooling

When you enable pooling, ActiveMatrix BusinessWorks creates a pool of TCP connections. TCP Connection pooling is available only on the client side and is disabled by default.

The maximum size of the pool can be specified using the Maximum Connection configuration field. A separate pool of TCP connections is created for every shared resource.

When pooled connections are required, the **TCP Open Connection** activity retrieves the connections from the pool. A check to identify stale connections is also done at the time of retrieval. Connections when freed are returned to the pool by the **TCP Close Connection** activity.

Overriding the host, port or local host values in the **TCP Open Connection** activity results in the creation of connections outside the pool, even if pooling is enabled on the referred shared resource.

TCP Open Connection

Activity



The **TCP Open Connection** activity opens a connection to a TCP server. After establishing the connection, the activity places a handle to the open connection in the connectionKey output element. This connection key can be used by subsequent activities in the process definition to read data from, write data to, or close the connection.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
TCP Connection	No	The TCP Connection shared configuration resource that describes the TCP server you want to connect to.
Local Host	No	Specifies the IP address on the local machine that you want to use for the TCP client.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
connection	complex	A complex element that contains the information about the TCP server you want to connect to.
host	String	The name or IP address of the TCP server you want to connect to. This value overrides the value specified by the TCP Connection resource on the Configuration tab.
port	number	The port number on which the TCP server is listening for requests. This value overrides the value specified by the TCP Connection resource on the Configuration tab.

Output

The output for the activity is the following.

Output Item	Datatype	Description
connection	binary	The handle to the open connection. This connection can be used by subsequent activities in the process definition to specify the TCP connection to write data to, read data from, or close.

TCP Receiver

Process Starter



The TCP Receiver process starter starts a new process when a client requests a TCP connection. When a connection request occurs, the process starter places a handle to the open connection in the connectionKey output element. This connection key can be used by subsequent activities in the process definition to read data from, write data to, or close the connection.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
TCP	No	The TCP Connection shared configuration resource that

Field	Global Var?	Description
Connection		describes the port on which the local machine is listening for incoming TCP/IP requests.
Keep-Alive	No	To specify whether the TCP connections should use the keep-alive feature of sockets, select the Keep-Alive check box.

Note: TCP Keep-Alive is different from HTTP Keep-Alive.

Setting TCP Keep-Alive Parameters

Set the following OS level parameters to make the TCP Keep-Alive work.

Operating System	Parameter Wait Time before Connection Probe	Parameter Interval between Retry Probes	Parameter for Maximum Retry Probes	Unit of Measure
AIX	tcp_keepidle	tcp_keepintvl	tcp_keepcnt	half-second
HP-UX 11i	tcp_time_wait_interval	tcp_keepalive_interval	tcp_keepalives_kill (1)	Milliseconds
Linux	tcp_keepalive_time	tcp_keepalive_intvl	tcp_keepalive_probes	Seconds
Solaris	tcp_time_wait_interval	tcp_keepalive_interval	NA	Milliseconds
Windows	KeepAliveTime	KeepAliveInterval	TcpMaxDataRetransmission	Milliseconds

Misc

The Misc tab contains the following fields:

Field	Description
Sequencing Key	This field can contain an XPath expression that specifies which processes should run in order. Process instances with sequencing keys that evaluate to the same value get executed sequentially in the order the process instance was created. For more information about controlling the execution order of process instances and about XPath expressions, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Custom Id	This field can contain an XPath expression that specifies a custom ID for the process instance. This ID is displayed in the View Service dialog of TIBCO Administrator, and it is also available in the <code>\$_processContext</code> process variable.

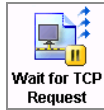
Output

The output for the activity is the following.

Output Item	Datatype	Description
connection	binary	The handle to the open connection. This connection can be used by subsequent activities in the process definition to specify the TCP connection to write data to, read data from, or close.
IP	string	IP address of the client sending the TCP request.
Host	string	Host name of the client sending the TCP request. This field is available only when the <code>enableDNSLookup</code> field in the Advanced tab of TCP Shared Connection is selected.
Port	integer	Port number of the client sending the TCP request.

Wait for TCP Request

Activity



The **Wait for TCP Request** activity waits for a TCP client connection request. When a connection request occurs, the activity places a handle to the open connection in the connection output element. This connection key can be used by subsequent activities in the process definition to read data from, write data to, or close the connection.

For information on running the "Wait for" activities in a multi-engine mode, see [Scalability With Incoming Events](#) in *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
TCP Connection	No	The TCP Connection shared configuration resource that describes port on which the local machine is listening for incoming TCP/IP requests.
Keep-Alive	No	To specify whether the TCP connections should use the keep-alive feature of sockets, select the Keep-Alive check box.
Note: TCP Keep-Alive is different from HTTP Keep-Alive.		

Event

The Event tab has the following fields.

Field	Description
Candidate Event Key	This field cannot be used. Specifying a value in this field does not cause the value to be compared with the key field in the activity input.
Event Timeout (msec)	A message may arrive before this activity is executed. This field specifies the amount of time (in milliseconds) a message waits if it is received before this activity is executed. If the event timeout expires, an error is logged and the event is discarded. If no value is specified in this field, the message waits indefinitely. If zero is specified, the event is discarded immediately, unless this activity has already been reached.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
key	string	This element cannot be used. Specifying a value for this element does not cause the value to be compared to the value of the Candidate Event Key field on the Event tab.
processTimeout	integer	The time (in milliseconds) to wait for the incoming TCP request. An error is returned if the request is not received in this time limit.

Output

The output for the activity is the following.

Output Item	Datatype	Description
connection	binary	The handle to the open connection. This connection key can be used by subsequent activities in the process definition to specify the TCP connection to write data to, read data from, or close.

Write TCP Data

Activity



The **Write TCP Data** activity sends data on the specified TCP connection. The connection must be opened by an activity or process starter that executed previously in the process definition. Activities that can open a TCP connection are: [TCP Receiver](#), [TCP Open Connection](#), and [Wait for TCP Request](#). The activity that opens a TCP connection places a handle to the open connection in its connectionKey output element. The connectionKey is mapped to this activity's input.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Type	No	Type of data to be written. This can be either Text or Binary. When Text is specified, the Separator, User Defined

Field	Global Var?	Description
Separator, and Encoding fields appear.		
Separator	Yes	The type of separator to use. The separator is written to the TCP connection after the specified data is written.
User Defined Separator	Yes	When User-defined separator is chosen from the list in the Separator field, this field appears. This field specifies the separator character to use. When Binary is selected in the Type field, a user-defined separator can be a string representation of a single byte value or it can be comma-separated byte values (for example, 0x0a, 0x0b, 0x0c). For valid string representations of byte values, refer to the Java documentation for the <code>decode(String n)</code> method of <code>java.lang.Byte</code>.
Encoding	Yes	When text data is specified, this field determines the character encoding to use for the text data.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
connection	binary	The handle to the connection that you want to write data to. This connection is obtained from a previously executed activity in the process definition that opened the connection.
body	String or binary	The data to write to the TCP connection. The datatype of this field is either String or binary, depending upon which type of data is selected in the Type field on the

Input Item	Datatype	Description
Configuration tab		
encoding	String	When text data is being written, this specifies the character encoding of the text. This value overrides the value specified in the Encoding field on the Configuration tab.
separator	String	When text data is being written, this specifies the separator to write after the body element is written to the TCP connection. This value overrides the value specified on the Configuration tab.

Output

The activity has no output.

Transaction Palette

Transactions allow you to perform a group of activities as a single unit of work. All activities in the group either complete or commit together or one or more failures in the group cause the transaction to roll back or undo the changes made during the transaction. For more information about transaction groups, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

JTA UserTransaction Configuration

Shared Configuration



The JTA UserTransaction Configuration shared configuration resource contains the information required to communicate with a Java Transaction API (JTA) transaction manager. This resource is used by transaction groups that use the JTA UserTransaction type. For more information about transaction groups, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Use Shared JNDI Configuration	No	When this field is checked, the JNDI Configuration field appears that allows you to choose a JNDI Configuration shared configuration resource.

Field	Global Var?	Description
		When this field is unchecked, the JNDI Context Factory, JNDI Context URL, JNDI User Name, and JNDI Password fields appear.
JNDI Configuration	No	This field only appears when the Use Shared JNDI Configuration field is checked. This field allows you to choose a JNDI Configuration shared configuration object that specifies the JNDI connection information. For more information, see JNDI Configuration.
JNDI Context Factory	No	This field only appears when the Use Shared JNDI Configuration field is unchecked. The InitialContextFactory to use when connecting to the JNDI server (javax.naming.Context.INITIAL_CONTEXT_FACTORY). This field provides a drop down list of InitialContext Factories supplied by each supported application server vendor. You can supply a different InitialContextFactory other than the default options in the drop down list.
JNDI Context URL	Yes	This field only appears when the Use Shared JNDI Configuration field is unchecked. Specifies the URL of the server (javax.naming.Context.PROVIDER_URL). An example URL is provided when one of the supported JNDI context factory classes is selected. For more information about the syntax of the URL, see your JNDI provider documentation . When using IBM WebSphere, you should use the JNDI Configuration resource instead of specifying the JNDI configuration on this resource. For more information, see IBM WebSphere and JNDI Context Caching.

Field	Global Var?	Description
JNDI User Name	Yes	This field only appears when the Use Shared JNDI Configuration field is unchecked. User name to use when logging into the JNDI server (javax.naming.Context.SECURITY_PRINCIPAL). If the JNDI provider does not require access control, this field can be empty.
JNDI Password	Yes	This field only appears when the Use Shared JNDI Configuration field is unchecked. Password to use when logging into the JNDI server (javax.naming.Context.SECURITY_CREDENTIALS). If the JNDI provider does not require access control, this field can be empty.
JNDI User Transaction Name	No	The name registered with the JNDI server or application server for JTA UserTransactions. When you select one of the classes from the drop down list in the JNDI Context Factory field, an example transaction name appears in this field. For the syntax of the transaction name, see your JNDI provider or application server documentation.
Transaction Timeout (sec)	Yes	Time in seconds to wait for transactions that use this resource to complete. An exception is raised if the transaction does not complete in the time specified by this field. Enter zero (0) in this field to use the default timeout set by the transaction manager.
Connection Retries	Yes	Maximum number of attempts to make to connect to the transaction manager.

Test Connection Button

The **Test Connection** button allows you to test that the specified configuration fields result in a valid connection to the transaction manager.

Transaction State

Activity



The **Transaction State** activity can be used in a transaction group to obtain the current state of the transaction or to set the transaction to a rollback only state. When a transaction is set to the rollback only state, the only outcome of the transaction that is permitted is a rollback.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Options	No	This field specifies the actions this activity can perform. Select one of the following options: • <code>getStatus()</code> – When this option is selected, the current status of the transaction is returned in the output of this activity. • <code>setRollbackOnly()</code> – When this option is selected, the current transaction is set to a rollback only state. The only outcome allowed in this state is for the transaction to rollback. You would use this option if an error condition is detected and you want to rollback the transaction without raising an exception. You can only use this option when using a transaction manager that supports the <code>setRollbackOnly</code> method.

Input

This activity has no input.

Output

The output for the activity is the following.

Output Item	Datatype	Description
Status	integer	The numeric code that corresponds to the current status of the transaction. The numeric status codes vary for each transaction manager, but most transaction managers use the value from <code>javax.transaction.Status</code> . For your transaction manager for a description of the possible status codes, see the documentation.

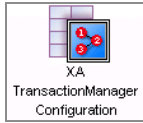
Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
TransactionStateException	An error occurred when attempting to retrieve or change the status of the transaction.
java.lang.Exception	The Java exception class.
exceptionClassName	The name of the exception class.
exceptionMessage	The text of the exception message.

XA TransactionManager Configuration

Shared Configuration



The XA TransactionManager Configuration shared configuration resource contains the information required to communicate with an XA-compliant transaction manager. This resource is used by transaction groups that use the XA Transaction type. For more information about transaction groups, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Transaction Manager	No	Name of the transaction manager service. Chose one of the following: - JOTM – For more information about Java Open Transaction Manager, see http://jotm.objectweb.org/. - Custom – Use this option when you are using an XA-compliant transaction manager. You must also specify the location of the transaction manager class files. You can optionally specify properties to pass to the transaction manager. For more information about configuring third-party transaction managers with ActiveMatrix BusinessWorks, see <i>TIBCO ActiveMatrix BusinessWorks™ Process Design</i>.
Transaction Timeout (sec)	Yes	Time in seconds to wait for transactions that use this resource to complete. An exception is raised if the transaction does not

Field	Global Var?	Description
		complete in the time specified by this field. Enter zero (0) in this field to use the default timeout set by the transaction manager.
Custom Transaction Manager	No	This field only appears when Custom is selected in the Transaction Manager field. This field allows you to select the location of the class for the implementation of the custom transaction manager. You can use an AliasLibrary resource to specify the location of the class files and then select that resource in the Library field. The class then appears in the Wrapper Class field. Alternatively, you can select the class using the browse button in the Wrapper Class field.
Optional Properties	No	This field only appears when you choose Custom in the Transaction Manager field. This field allows you to specify optional input parameters for the custom transaction manager implementation. You can add, remove, or move the parameters using the +, X, and arrow keys to the right of the table. Specify the name of the parameter, its datatype, and the value of the parameter.

WSDL Palette

The **WSDL** palette is used for creating, editing, validating, importing, and viewing WSDL (Web Service Description Language) files. These files are shared resources that describe web services. The WSDL palette's resources are used to create and reuse abstract definitions, namely:

- messages, which describe the data being exchanged
- portTypes, which define a set of operations
- and operations, which describe the activities supported by the web service

You can learn more about WSDL at <http://www.w3.org>.

WSDL

Resource

A WSDL file is an XML instance of the WSDL XML Schema, found at <http://schemas.xmlsoap.org/wsdl/>. A WSDL resource is composed of a series of definitions that describe the format and roles of messages used in web services. To create a new WSDL resource (with an empty set of definitions), drag the WSDL icon from the palette panel into the project panel. Once the WSDL resource has been created and selected in the project, the key components of abstract WSDL definitions – messages, portTypes, and operations – become available in the **WSDL** palette.

Configuration

The configuration panel is accessible when working in the model view (which loads by default). The fields in the Configuration tab are described below.

Field	Description
Resource	This is the name of the file when persisted. The *.wsdl suffix gets added

Field	Description
Name	automatically.
Description	Short description of the activity. This field is optional.
Target Namespace	This is the target name space of the WSDL document. This field get populated by default with an example target namespace (http://xmlns.example.com). It is recommended that you enter a unique and meaningful namespace for each WSDL file. (In the WSDL source, this field represents the value of the targetNamespace attribute on the definitions element.)
Definition Name	A name for the definitions. Entries in this field are optional. Names must begin with a letter and may contain numbers, underscores, dashes, and periods. (In the WSDL source, this field represents the value of the name attribute for the definitions element.
Copy From URL	Once a new WSDL resource has been added to the project, you can copy the definitions of an existing WSDL file. To do so, enter the URL of the existing WSDL file to the Copy From URL field. Click Copy URL to retrieve the WSDL file and make a copy in the project. Once the copy is complete, all of the WSDL definitions are visible and editable. For example, if your WSDL file is located on the local MS Windows machine, you can use a URL like <code>file:///c:/wsdl/order.wsdl</code>. Instead of using this field, if you have the source code of your WSDL file, you can also copy it onto the clipboard and click the Display Source View button (For a description of toolbar buttons, see Toolbar). You can then paste the source of your WSDL into the source window. To locate a WSDL file by browsing a WSIL or UDDI registry, use the Browse UDDI button. (For a description of the browsing interface, see Copying WSDL Definitions.) Once the WSDL file has been selected, click Copy URL. Warning: The Copy URL and Browse UDDI buttons add the WSDL from the external resource once. There is no refresh option. If the external WSDL file changes, the Copy URL must be performed again.

Toolbar

When a WSDL resource, or one of its components, is selected in your project, four extra buttons appear on the toolbar. These buttons are described in [WSDL toolbar buttons](#).

WSDL toolbar buttons

Button	Description
	Opens the model view. The model view is the default view so, by default, this button is inactive. For more information on the model view, see Editing WSDL Resources.
	Opens the content view. For more information on the content view, see Editing WSDL Resources.
	Opens the source view. For more information on the source view, see Editing WSDL Resources.
	Checks changes made in the content view to verify that the WSDL resource conforms to the WSDL schema. Errors are reported in an errors panel, which is displayed at the bottom of the content view.

Editing WSDL Resources

The key components (elements) of an abstract WSDL file are messages, portTypes, and operations. Three different views are available when editing the base WSDL file or one of its components. These editing views, accessed using the toolbar buttons (see [WSDL toolbar buttons](#)) are described below.

View	Description
Model	This is the default view. When a WSDL resource is selected, the Configuration tab provides fields for setting document level properties as well as an interface for copying existing WSDL definitions. For more information, see Configuration).

View	Description
	To add a Message or PortType component to the WSDL resource, drag the representative icon from the WSDL palette into design panel. The Message and PortType components can then be selected and edited. - When a Message component is selected in the project, the Configuration tab provides fields for naming and describing the component and a table for adding part components. For more information, see Message. - When a PortType component is selected, the Configuration tab provides fields for naming and documenting the PortType component. For more information, see PortType. To add an Operation component to the PortType, drag the representative icon from the WSDL palette into the design panel. The Operation component can then be selected and edited. - When an Operation component is selected, the Configuration tab provides fields for naming and describing the component and a table for specifying the messages to be used as input and output. For more information, see Operation. Note: The WSDL palette's model view enables the creation of an "abstract" WSDL file that may be used in a web service where additional SOAP endpoints and binding information are added. To specify binding and endpoint information in the WSDL file, use the content or source view.
Content	The content view provides a tabular view of the WSDL file, with the components (elements and attributes) of the WSDL file listed on the left, and the component values listed on the right. The content view provides access to all of the elements and attributes used to build WSDL files, which appear as choices on the factory bar located at the bottom of the view. The content view is useful when viewing concrete WSDL files or when you need to add elements to your resource that are not available in the model view - such as binding or service elements. The Content view provides the same editing environment used to create XML instance resources. For details on the content view, review the documentation for the Instance resource, in <i>TIBCO Designer™ Palette reference</i>.

View	Description
Source	The source view allows you to edit the source code of your document directly. If you are comfortable with WSDL syntax, editing directly in the WSDL source may be useful. To make the other panels reflect the changes you've made in the source, click 'Reparse' at the top of the panel. Any errors found upon reparse get reported.

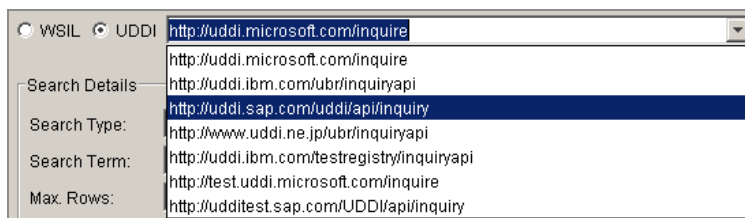
Copying WSDL Definitions

To copy the definitions of an existing WSDL file, you must first add a new WSDL resource to your project. If you know the location of the WSDL file you want to copy, enter the URL into the Copy From URL field and click Copy URL. The definitions from the external file get copied into your WSDL resource.

If you are not sure of the location of the WSDL file, a WSIL and UDDI Registry Browser is provided, enabling you to browse well-known WSIL and UDDI directories. To access the browser, click **Browse UDDI**.

Use the radio buttons at the top of the browser to indicate whether you want to browse a WSIL directory or a UDDI registry. For the selected registry type, you can type in the location of the registry you want to browse in the field adjacent to the radio buttons or you can use this field's drop down menu to select from a list of popular registries, as shown in [Specifying a UDDI registry](#).

Figure 27: Specifying a UDDI registry



Once you have selected the registry to browse, use the Search Details section to select the type of services you want to browse. The Search Details section has the following fields.

Field	Description
Search Type	Use this field to specify the facet you want to use to locate the WSDL file. When browsing a WSIL registry, searches are conducted based on service name only. When browsing a UDDI registry, for a description of the available search types, see UDDI search types .
Search Term	To filter the registry's services, enter a search term corresponding to the facet selected in the Search Type field.
Max Rows	Specify the number of services you would like to be returned.
Case-sensitive	Indicate if the case of the search term should be considered when locating services.
Order Asc.	Indicate if the results should be ordered ascending. (Files are returned in descending order by default.)
Exact Match	Indicate if the search term must be matched exactly.
Timeout (msec)	Specify the amount of time permitted before a search is aborted.

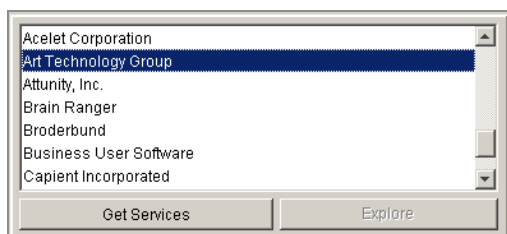
UDDI search types

Search Type	Description
Service Name	In the Search Term field, specify a service name.
Business Name	In the Search Term field, specify business name.
Business Key	In the Search Term field, specify a uuid key for a business.
TModel Key	

Search Type	Description
ALL WSDLs	In the Search Term field, specify a service name. It searches TModels of type wsdlSpec.
NAICS	In the Search Term field, specify a NAICS number. For more information, see http://www.naics.com .
ISO 3166	In the Search Term field, specify the code for a particular country. For a list of ISO 3166 codes, see http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html
D-U-N-S	In the Search Term field, enter the nine digit D-U-N-S number. For more information, see http://www.dnb.com .
Geo Web	In the Search Term field, enter a Microsoft geoweb classification. For example, the value 514267 would return businesses from London.
SIC Code	In the Search Term field, enter the four digit SIC code. For more information, see http://www.siccode.com .
UNSPSC	In the Search Term field, enter a UNSPSC code. For more information, see http://eccma.org/unspsc/ .

Depending on the Search Type specified in the Search Details, either the Get Services or Explore buttons become active. The Explore option returns a list of company names matching the search criteria, such as the companies with a particular NAICS or SIC code. (To see the services provided one of the companies on the list, select the company and click Get Services.) The results returned by the Explore button are shown in [Companies returned by the Explore button](#).

Figure 28: Companies returned by the Explore button



The Get Services option displays the WSDL files matching the search criteria in the table below the Search Details section, as shown in [List of returned services](#).

Figure 29: List of returned services

Name	Description	WSDL or Access URL
Up to date NASCAR Winston Cup Statistics for cu...		http://www.einsteinware.com/webservices/nas...
Converts a given amount in digits to its equivalent...		http://www28.brinkster.com/hegdes/NumberT...
Converts a given amount in digits to its equivalent...		http://www28.brinkster.com/hegdes/NumberT...
Web service to calculate bond prices, yield to mat...		http://www.financialwebservices.ltd.uk/axis/ser...
Web service to calculate bond prices, yield to mat...		http://www.financialwebservices.ltd.uk/axis/ser...
ies that a given Singapore National Registration I...		http://www.rightsecurity.biz/wsdl/NRICWebServ...
ies that a given Singapore National Registration I...		http://www.rightsecurity.biz/wsdl/NRICWebServ...
Comparison of two pictures.		http://otourdes.europe.webmatrixhosting.net/E...
Comparison of two pictures.		http://otourdes.europe.webmatrixhosting.net/E...

From the list of returned WSDL files, highlight the file you want to copy and click OK. The URL for the service appears in the Copy From URL field of the Configuration tab of the WSDL resource. Click Copy URL. The definitions from the external file get copied into your WSDL resource.

External References

WSDL or XSD on an HTTP network location can be imported in a local WSDL without physically importing them into the project repository. These external references are restricted to imports only, that is, <wsdl:import> and <xsd:import> elements from WSDL and XSD namespaces, respectively.

Changes made to the external references are included automatically. This behavior is the same as with any locally imported WSDL or XSD.

The property `java.property.com.tibco.xml.tns.parse.ImportSpecialResolution` should be set to true in `designer.tra` to enable external WSDL support.

Message

Component

The Message component is used to define the type of information being exchanged in a web service. One or more Message components can be added to a WSDL resource selected in your project.

Configuration

The configuration panel is accessible when working in the model view (which loads by default). The fields in the Configuration tab are described below.

Field	Description
Message Name	A name for the message. In the target namespace, message names must be unique and must begin with a letter and may contain numbers, underscores, dashes, and periods. (In the WSDL source, this field represents the value of the name attribute on the message element.)
Description	Optional short description of the activity. This information is added to the WSDL source as documentation.
Part Table	Messages contain one or more parts that provide references to type information. Use this table to add part components to your message component. The Part Table contains the following columns: • Name – Double click in this field to enable editing. Once activated, enter or edit the name of the part here. In the target namespace, part names must be unique and must begin with a letter and may contain numbers, underscores, dashes, and periods. (In the WSDL source, this field represents the value of the name attribute for the part element.) • Type – This field displays the type information for the part. (In the WSDL source, this field represents the value of the type or element attribute of the part element.) The type may be a built-in XML Schema simple type or an element or complex type defined in an XML Schema. This field is not editable. The value for this field is specified by way of the Part Details fields. When working in the Part Table, use the “+” button to create a new row, the “X” to delete rows, and the arrows to re-order parts.
Part Details	The Part Details options, used to specify type information for the part, appears when a row is selected in the Part Table. Indicate whether the reference should be to a type or element by clicking the respective radio button. Select Type to reference an XSD simple type

Field	Description
	(built-in or user-defined) or complex type. Select Element to reference an element defined in an XML Schema.
	Type information is selected using the Type drop down menu. If the XML Type Reference option is selected, or if your part references an element, use the Browse button. The Select a Resource window appears, enabling you to select a type or element by browsing the project or namespace directories, or by searching for a specific name.

Toolbar

When a Message component is selected in your project, four extra buttons appear on the toolbar. These buttons are described in [WSDL toolbar buttons](#).

Editing Message Components

There are three views available for editing message information. The editing views, accessed using toolbar buttons (see [WSDL toolbar buttons](#)), are described in [Editing WSDL Resources](#).

PortType

Component

The PortType component describes the operations that are supported by the web service. One or more PortType components can be added to a WSDL resource selected in your project. Once the portType component has been created and selected in the project, the Operation component becomes available in the **WSDL** palette.

Configuration

The configuration panel is accessible when working in the model view (which loads by default). The fields in the Configuration tab are described below.

Field	Description
Port Type Name	The name for the portType component. In the target namespace, portType names must be unique and must begin with a letter and may contain numbers, underscores, dashes, and periods. (This field represents the value of the name attribute for the portType element.)
Description	Optional short description of the portType activities. This information is added to the WSDL file as documentation.

Toolbar

When a PortType component is selected in your project, four extra buttons appear on the toolbar. These buttons are described in [WSDL toolbar buttons](#).

Editing PortType Components

The key component (element) of a PortType is an operation. Three different views are available when adding or editing operation components. These editing views, accessed using toolbar buttons (see [WSDL toolbar buttons](#)), are described in [Editing WSDL Resources](#).

Operation

Component

The Operation component describes the message formats for the incoming and outgoing messages required by the activities supported by the portType. One or more Operation components can be added to a PortType component selected in your project.

Configuration

The configuration panel is accessible when working in the model view (which loads by default). The fields in the Configuration tab are described below.

Field	Description
Operation Name	The name of the Operation component. In the target namespace, names must be unique and must begin with a letter and may contain numbers, underscores, dashes, and periods. (In the WSDL source, this field represents the value of the name attribute for the operation element.)
Description	Optional short description of the operation. This information is added to the WSDL resource as documentation.
Message Table	Use this table to specify the format for the operation's input and output messages. The Message Table contains three columns: • Message Kind – Use the drop down menu to select either input, output, or fault. By default an input message (which is accepted by the operation) and an output message (which is sent by the operation) are listed on the first two rows of the table. Because only one input and one output message is allowed per operation, any additional message rows are added as fault messages (which are used for error reporting). • Name – Double click in this field to enable editing. Once activated, enter the name of the input, output, or fault component here. (In the WSDL source, this field represents the value of the name attribute for either the input or output element.) • Message – This field displays the name of the message component that describes the desired format. (In the WSDL source, this field represents the value for the message attribute of the input or output element.) This field is not editable. The value for this field is specified using the Message Details fields. When working in the Message Table, use the “+”, “X”, and arrows to create, delete, and re-order operation messages, respectively.
Message Details	When a particular row in the Message Table is selected, the namespace box defaults to the target namespace of the current WSDL file. In the field below, you can select a message component defined in this namespace using the drop down menu. To select a message component defined in a different WSDL resource,

Field	Description
	use the Browse button. The Select a Resource window appears, enabling you to select a message by browsing the project or namespace directory, or by searching for a message name. When a message from a different namespace is selected, the appropriate import namespace declarations are automatically added to the document.

Toolbar

When a Operation component is selected in your project, four extra buttons appear on the toolbar. These buttons are described in [WSDL toolbar buttons](#).

Editing Operation Components

There are three views available for editing operation information. The editing views, accessed using toolbar buttons (see [WSDL toolbar buttons](#)) are described in [Editing WSDL Resources](#).

XML Activities Palette

The **XML Activities** palette provides activities for parsing XML strings into schemas and rendering schemas into XML strings.

Parse XML

Activity



The **Parse XML** activity takes a binary XML file or an XML string and processes it, turning it into an XML schema tree based on the XSD or DTD specified.

The preferred way to parse XML files is to use a [Read File](#) activity set to binary mode to read the XML file. Then pass the binary file contents to the **Parse XML** activity.

Parsing Date and Datetime Strings

In XML documents parsed by the **XML Parser** activity, datetime values are read in according to the ISO-8601 standard, as described in the XML Schema specification. For example, the value:

```
2002-02-10T14:55:31.112-08:00
```

is 55 minutes, 31 seconds, and 112 milliseconds after 2PM on February 10th, 2002 in a time zone that is 8 hours, 0 minutes behind UTC.

If no time zone field is present, the value is interpreted in the time zone of the machine that is performing the parsing. This can lead to complications if you are processing XML from different time zones, so you are encouraged to always use time zones.

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Input Style	No	Can be binary, text, or dynamic. In binary mode (the default and preferred choice), the binary content is read. The encoding used for parsing the content is either the value specified in the forceEncoding input item, the encoding specified in the XML header of the xmlBinary input item, or UTF-8 (the default encoding, if no encoding is specified). In text mode, an XML string is passed as an input item. In dynamic mode, a choice is offered for input. You can either supply binary or text input. You can use a choice statement and set substitution in the mapping to supply the correct type of input at run time.
Validate Output	No	When checked, specifies that the output of the activity should be validated against the schema specified in the Output Editor tab.

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
xmlString	string	The XML string to parse. This input element is available when the Input

Input Item	Datatype	Description
		Style configuration field is set to text.
xmlBinary	complex	Available when the Input Style configuration field is set to binary. This input item contains a required bytes item for which you specify the input XML bytes. This input item also contains an optional forceEncoding item. This specifies the encoding to use when parsing the XML. If this element is not specified, the encoding specified in the XML header of the byte element is used to parse the XML. If the encoding value is not specified in the XML header, the default encoding (UTF-8) is used. You can specify any valid XML encoding value listed in IANA-Charsets. For a complete list, see http://www.iana.org/assignments/character-sets. Encoding names are case insensitive.
xmlBinary xmlString	choice	Available when the Input Style configuration field is set to dynamic. This allows you to specify a choice statement and set substitution and supply the correct input type at runtime.

Output Editor

The Output Editor tab allows you to define a schema for the XML output. For a complete description of using the Output Editor tab, see [Specifying Data Schema](#).

Output

The output for the activity is the following.

Output Item	Datatype	Description
XML Schema	complex	The parsed XML schema is the output of this activity. The contents of the schema are determined by the schema defined on the Output Editor tab.

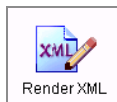
Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
XMLParseException	An exception occurred when parsing the data.
UnsupportedEncodingException	An unsupported encoding was specified in the forceEncoding input item. You can specify any encoding supported by Java on the system where ActiveMatrix BusinessWorks is running.
MissingByteCountException	

Render XML

Activity



The **Render XML** activity takes an instance of an XML schema element and renders it as a stream of bytes containing XML or an XML string. The schema is processed based on the XSD file specified.

Rendering Date and Datetime Strings

When ActiveMatrix BusinessWorks generates datetime strings, it always uses UTC time. For example, the time 55 minutes, 31 seconds, and 112 milliseconds after 2PM on February 10th, 2002 would be represented as the following by the **Render XML** activity:

```
2002-02-10T14:55:31.112Z
```

Configuration

The Configuration tab has the following fields.

Field	Global Var?	Description
Name	No	The name to appear as the label for the activity in the process definition.
Description	No	Short description of the activity.
Output Style	No	Specifies whether the output should be text or binary data.
Validate Input	No	When checked, specifies that the data input to the activity should be validated against the schema specified in the Input Editor tab.
Format Using Default Namespace Prefix	No	
Encoding	Yes	This field is only available when the Output Style field is set to text. This field specifies the encoding value to place into the XML header of the rendered XML data. You can specify any valid XML encoding value listed in IANA-Charsets. For a complete list, see http://www.iana.org/assignments/character-sets. Encoding names are case insensitive.

Advanced

The Advanced tab allows you to specify which elements in the input schema can contain CDATA sections. CDATA sections allow you to enclose text within an element that should not be treated as XML.

The + button allows you to add elements of the input schema to the list. When you click the + button, a dialog appears that allows you to locate and select the input schema by location in the project or by namespace. Use the By Location tab if your input schema is located in the project directory. Use the By Namespace tab if you constructed the input schema on the Input Editor tab.

Use the X button to remove elements from the list and the up and down arrows to move elements in the list.

Input Editor

The Input Editor tab allows you to define a schema for the incoming XML data you want to render. For a complete description of using the Input Editor tab, see [Specifying Data Schema](#).

Input

For more information about mapping and transforming input data, see *TIBCO ActiveMatrix BusinessWorks™ Process Design*.

The input for the activity is the following.

Input Item	Datatype	Description
byteEncoding	string	This is only available when the Output Style field on the Configuration tab is set to binary. This specified the encoding value to place into the XML header in the rendered XML output. This is also the encoding used to encode the binary data. If this item is not specified, the default encoding of the Java Virtual Machine used by the process engine is used to encode

Input Item	Datatype	Description
		the binary data. This value may not be a valid XML encoding. You can specify any valid XML encoding value listed in IANA-Charsets. For a complete list, see http://www.iana.org/assignments/character-sets. Encoding names are case insensitive.
XML Schema	string	The XML schema to render. The contents of the schema are determined by the schema specified on the Input Editor tab.

Output

The output for the activity is the following.

Output Item	Datatype	Description
xmlString	string	Available when text is selected in the Output Style field on the Configuration tab. This item contains the rendered XML string.
xmlBytes	bytes	Available when binary is selected in the Output Style field on the Configuration tab. This item contains a stream of bytes representing the rendered XML.

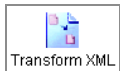
Error Output

The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
XMLRenderException	An exception occurred when rendering the data.
UnsupportedEncodingException	An unsupported encoding was specified in the byteEncoding input item. You can specify any encoding supported by Java on the system where ActiveMatrix BusinessWorks is running.
MissingByteCountException	

Transform XML

Activity



The Transform XML activity allows you to transform an input XML document into the output specified by the given [XSLT File](#) shared configuration resource.

Normally, transformation of data occurs by mapping process variables to an activity's input and applying XPath expressions to perform any transformation. If you have an XSLT file that you are using for transformations, or if an outside source supplies an XSLT file, this activity allows you to use the XSLT file instead of manually creating the mappings.

For more information about creating and editing XSLT files, see the XSLT specification at <http://www.w3.org/TR/xslt>. For more information about creating an XSLT File shared configuration resource, see [XSLT File](#).

For an example of using the Transform XML activity, see [Example of Transforming XML](#).

Configuration

The Configuration tab has the following fields.

Field	Description
Name	The name to appear as the label for the activity in the process definition.
Description	Short description of the activity.
Stylesheet	The XSLT File shared configuration resource to use when transforming the XML. You can also specify an XSLT stylesheet in the stylesheet input element, if desired. Stylesheets specified in the activity input override the stylesheet specified in this field. For more information, see XSLT File.
Input and Output Style	Specifies whether the input and output is binary data, text, or an XML tree.
XSLT Engine	Specifies the XSLT Engine to be used for the transformation. Following are the XSLT engines available in the drop down list: • TIBCO XML - This is the default XSLT engine provided by TIBCO. • Saxon-B XSLT 2.0 - This engine provides basic conformance to XSLT 2.0. The engine provides all features of the languages except schema-aware processes. For details about the Saxon-B XSLT 2.0 engine, see http://saxon.sourceforge.net/.

Input

The input for the activity is the following.

Input Item	Datatype	Description
xmlBytesIn or xmlString or xmlTree	binary	The XML to transform. If the input/output style is set to binary, this element is named xmlBytesIn. If the input/output style is set to text, this element is named xmlString. If the input/output style is set to tree, this element is named xmlTree.

Input Item	Datatype	Description
		When an xmlTree is the input element, you can drag and drop an XML tree into the (any Element) to map a copy of an XML tree to the activity's input. For an example of this, see Input for the example Transform XML activity .
styleSheet	string	This optional element contains the XSLT stylesheet to use to transform the XML input. This stylesheet overrides the stylesheet specified in the Stylesheet field on the Configuration tab.
parameter	complex	This repeating element contains the input parameters for the style sheet. Each input parameter is specified as name/value pairs. The name is a string that corresponds to the name of the parameter specified in the XSLT stylesheet parameter list.
name	string	The name of the stylesheet input parameter.
value	string	The value to supply for the specified stylesheet input parameter.
isXMLDocument	boolean	Specifies whether the parameter is an XML document.

Output

The output for the activity is the following.

Output Item	Datatype	Description
xmlString or xmlOutput or treeOutput	text or binary or complex	A binary or text value or an XML tree containing the output schema specified by the XSLT file. The element is named xmlString and is of type text when

Output Item	Datatype	Description
		text is selected for the input/output style on the Configuration tab. The element is named xmlOutput and is of type binary when binary is selected. The element is named treeOutput and is a complex type when tree is selected.

Error Output

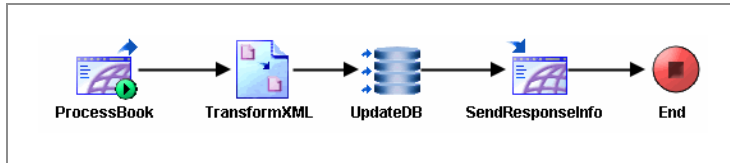
The Error Output tab lists the possible exceptions that can be thrown by this activity. For more information about error codes and corrective action to take, see *TIBCO ActiveMatrix BusinessWorks™ Error Codes*.

Exception	Thrown When...
XMLTransformException	An exception occurred when transforming the data.
UnsupportedEncodingException	Data was supplied that is encoded in an unsupported encoding. You can specify any encoding supported by Java on the system where ActiveMatrix BusinessWorks is running.
MissingByteCountException	

Example of Transforming XML

The following is an example of using the **Transform XML** activity to change an input XML document into the desired output schema. In this example, books are cataloged and processed. The user enters a book's information into a web interface, and the web input form supplies a new catalog number and the date it was entered into the catalog. The process definition accepts the incoming web request, transforms the XML and adds the catalog number and date, enters the data into a database, then returns the results to the user. [A process definition using the Transform XML activity](#) illustrates the process definition for this example.

Figure 30: A process definition using the Transform XML activity



The **Transform XML** activity uses an XSLT file that accepts two input parameters, `catalogNumber` and `catalogDate`. These input parameters are added as elements to the Book schema. The following is the source of the XSLT File shared configuration resource for the example **Transform XML** activity:

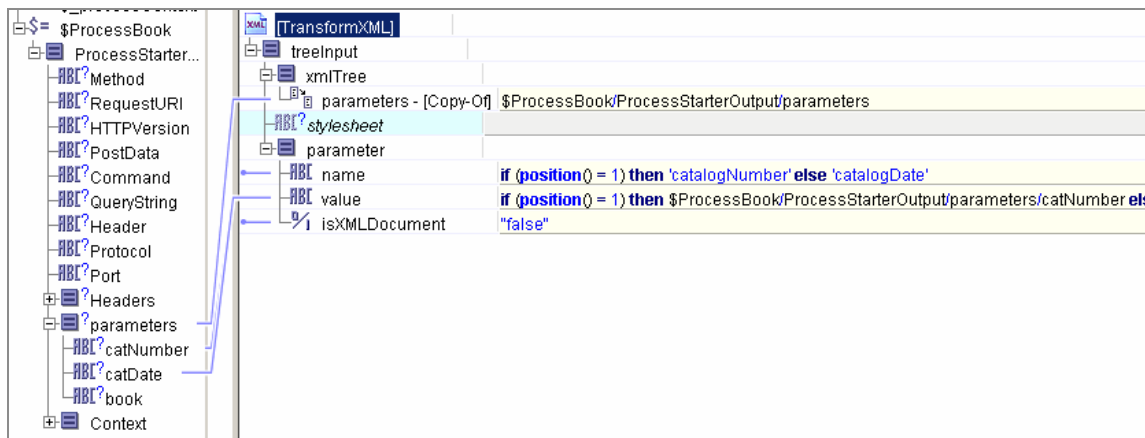
```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
  xmlns:pfx="http://www.books.org">
  <xsl:param name="catalogNumber"> <!--type="string"--></xsl:param>
  <xsl:param name="catalogDate"> <!--type="string"--></xsl:param>
  <xsl:template match="/*">
    <pfx:BookStore>
      <xsl:for-each select="Book">
        <xsl:call-template name="handle-book">
          <xsl:with-param name="inDate" select="$catalogDate"/>
        </xsl:call-template>
      </xsl:for-each>
    </pfx:BookStore>
  </xsl:template>
  <xsl:template name="handle-book">
    <xsl:param name="inDate"> <!--type="string"--></xsl:param>
    <pfx:Book>
      <pfx:Title><xsl:value-of select="Title"/></pfx:Title>
      <pfx:Author><xsl:value-of select="Author"/></pfx:Author>
      <pfx:Date><xsl:value-of select="$inDate"/></pfx:Date>
      <pfx:ISBN><xsl:value-of
        select="$catalogNumber"/></pfx:ISBN>
      <pfx:Publisher><xsl:value-of
        select="TIBCO Software Inc."/></pfx:Publisher>
    </pfx:Book>
  </xsl:template>
</xsl:stylesheet>

```

When you configure the **Transform XML** activity, you specify the incoming XML to transform any input parameters to the schema. [Input for the example Transform XML activity](#) illustrates the input for the example **Transform XML** activity.

Figure 31: Input for the example Transform XML activity



The stylesheet has two input parameters. The XPath if statement is used to place the correct names and values into the correct parameters. Each parameter is specified as a name/value pair. Notice that the name of each parameter corresponds to the name specified for that parameter in the XSLT file.

The output of the **Transform XML** activity is a text string containing the XML. In our example process definition, we must insert the transformed data into a database table. To map the transformed data into the database UPDATE statement, the data must first be represented as a schema. To do this, we use the **Parse XML** activity to parse the output of Transform XML into an XML schema.

XSLT File

Shared Configuration



The XSLT File resource allows you to load an XSLT file to use to transform XML schemas using the [Transform XML](#) activity. For more information about using this shared configuration resource in a **Transform XML** activity, see [Transform XML](#).

A transformation expressed in XSLT is called a stylesheet. A stylesheet contains a set of template rules. Each template rule has two parts: a pattern that is matched against nodes in the source document and a template that can be instantiated to form part of the result document.

A transformation expressed in XSLT describes rules for transforming a source document into a result document. The transformation is achieved by associating patterns with templates. A pattern is matched against elements in the source document. A template is instantiated to create all or part of the result document. The structure of the result

document can be completely different from the structure of the source document. In constructing the result document, elements from the source document can be filtered and reordered, and arbitrary structure can be added.

Specifying Data Schema

Many activities in ActiveMatrix BusinessWorks allow you to specify a custom data schema for input or output of the activity. This appendix describes the mechanism for specifying a custom data schema for an activity's input or output.

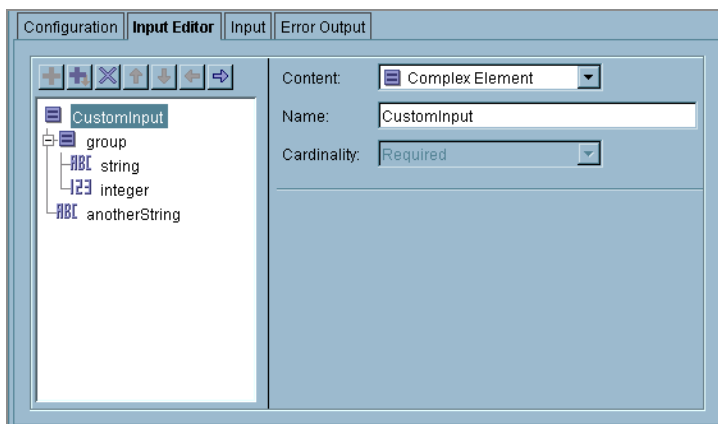
Editor

The Editor tab is used to specify a data schema for input or output of an activity. This is useful when the data does not have a well-known structure. The Editor tab is usually named for the type of schema you are creating. For example, the tab may be named "Input Editor" or "Output Editor".

For example, an email message has a well-known data structure, and therefore does not need a special datatype for its input. A JMS message, however, can have application-specific properties of any datatype. The Editor tab allows you to define the schema for any activities that require a specialized input or output schema.

You can use a simple datatype, or you can define a group of data elements on this tab. You can also reference XML schema or ActiveEnterprise classes stored in the project. Once defined, the schema appears on the appropriate tabs of the activity. The data in the schema then becomes available to other activities in the process definition.

The following illustrates the Editor tab. In this example, the Editor tab is labeled "Input Editor" indicating this is the activity's input.



To define a schema on this tab, use the icons above the schema tree to add, delete, or move data items. Then use the fields of the dialog to specify the datatype of each item.

Field	Description
Content	Defines the content of the element. The other fields that appear depend upon which content type is selected. For more information about the content type, see Content types for schema elements .
Name	The name of the element.
Cardinality	The qualification for the data item. Data items can be specified as one of the following: • Required – the data item is required and must be supplied when the process is called. • Optional (?) – the data item is optional. • Repeating, Zero or More (*) – The data item is a list that has zero or more elements. • Repeating, One or More (+) – The data item is a list that has one or more items.
Type	The type of data. Can be any of the following: • XML Type Reference – must locate the stored XML schema definition. • Any of the datatypes described in Icons for schema datatypes.
Schema Name	Stored XML schema that contains the element or type you want to reference.
Type Name	Type in a stored XML schema you want to reference.

[Content types for schema elements](#) describes the potential content types for data elements.

Content types for schema elements

Content Type	Description	Other Fields for This Content Type
Complex Element	An element that contains other elements. This is like a structure in a programming language. The complex element can contain zero or more elements of other types, and it can contain other complex elements.	Name Cardinality
Element of Type	An element with a specified datatype. You can specify a scalar datatype (string, integer, and so on), you can reference an XML type, or specify the TIBCO ActiveEnterprise Any datatype.	Name Cardinality Type Other fields depending upon the datatype selected
XML Element Reference	A reference to an element in a stored XML schema. For more information about XML schema, see TIBCO Designer documentation .	Cardinality Schema Element
Attribute of Type	An attribute with a specified datatype. You can specify a scalar datatype (string, integer, and so on), you can reference an XML type, or specify the TIBCO ActiveEnterprise Any datatype.	Name Cardinality Type Other fields depending upon the datatype selected.
Sequence	A sequence of elements. Each item in the sequence is a	Cardinality

Content Type	Description	Other Fields for This Content Type
	structure of the sub-elements of this element.	
Choice	A choice of elements. The datatype of this element can be one of the sub-elements defined.	Cardinality
All	The datatype of this element can be all of the datatypes of the sub-elements defined.	Cardinality
XML Group Reference	A reference to an XML group in a stored XML schema. For more information about XML schema, see TIBCO Designer documentation .	Cardinality Schema Model Group
Any Element	A reference to any XML Element. You can use the Coersion button to supply a reference to the XML Element for this item when it appears in the input or process data.	Cardinality Validation
WSDL Message	A reference to a message defined in a WSDL File resource. Use the Browse button to bring up a dialog for locating WSDL files within the project.	WSDL Message

[Icons for schema datatypes](#) describes the datatypes available for data.

Icons for schema datatypes

Icon	Description
	String or character value. You can specify the type of string as one of the following: • String • Normalized String • Token

Icon	Description
	• Language • Name • NC-Name • Q-Name • Name Token • Name Tokens • ID • ID ref • ID refs • Entity • Entites



Integer value. You can specify the size of the integer as one of the following:

- Byte
- Short
- Int
- Long
- Unsigned Byte
- Unsigned Int
- Unsigned Long
- Integer
- Positive Integer
- Negative Integer
- Non-positive Integer
- Non-negative Integer

Icon	Description
	Floating point number. You can specify the size of the schema item as float, double, or decimal.
	Boolean value.
	Date or Time. This can be any of the following datatypes: • Time • Date • Date & Time • Duration • Day • Month • Year • Year & Month • Month & Day
	Base 64 or hexadecimal value.
	An HTTP Uniform Resource Identifier.
	Complex element. Container for other datatypes.
	XML element or group reference.
	Sequence. Signifies that the contained sub-elements are repeated in an ordered sequence.
	Any Type. Represents a schema item with the TIBCO ActiveEnterprise datatype any. This node can be specified as any other datatype or a reference to an XML Type or AE Class.

Icon	Description
	Any Element. Represents a schema item that can be a reference to any XML Element. You can use the Coersion button to supply a reference to the XML Element for this item.
	Choice. Specifies that the schema element can be one of a specified set of datatypes.

TIBCO Product Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [Product Documentation website](#), mainly in HTML and PDF formats.

The [Product Documentation website](#) is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for this product is available on the [TIBCO ActiveMatrix BusinessWorks™ Product Documentation](#) page:

- *TIBCO ActiveMatrix BusinessWorks™ Release Notes*
- *TIBCO ActiveMatrix BusinessWorks™ Administration*
- *TIBCO ActiveMatrix BusinessWorks™ Concepts*
- *TIBCO ActiveMatrix BusinessWorks™ Error Codes*
- *TIBCO ActiveMatrix BusinessWorks™ Getting Started*
- *TIBCO ActiveMatrix BusinessWorks™ Installation*
- *TIBCO ActiveMatrix BusinessWorks™ Palette Reference*
- *TIBCO ActiveMatrix BusinessWorks™ Process Design*

To directly access documentation for this product, double-click the following file:

`TIBCO_HOME/release_notes/TIB_<productID>_version_docinfo.html`

where `TIBCO_HOME` is the top-level directory in which TIBCO products are installed. On Windows, the default `TIBCO_HOME` is `C:\tibco`. On UNIX systems, the default `TIBCO_HOME` is `/opt/tibco`.

Other TIBCO Product Documentation

When working with ActiveMatrix BusinessWorks, you may find it useful to read the documentation of the following TIBCO products:

- TIBCO Designer™: *TIBCO Designer is an easy to use graphical user interface for design-time configuration of TIBCO applications. TIBCO Designer includes online help for each palette.*
- TIBCO Runtime Agent™: *TRA supplies a number of TIBCO and third-party libraries used by ActiveMatrix BusinessWorks.*
- TIBCO Administrator™: *TIBCO Administrator is the monitoring and managing interface for new-generation TIBCO products such as ActiveMatrix BusinessWorks.*
- TIBCO Rendezvous®: *TIBCO Rendezvous software uses messages to enable distributed application programs to communicate across a wide variety of hardware platforms and programming languages.*

How to Access Related Third-Party Documentation

When working with ActiveMatrix BusinessWorks, you may find it useful to read the documentation of the following third-party products:

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our [product Support website](#).
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the [product Support website](#). If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote

on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to [TIBCO Community](#).

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. ("CLOUD SG") SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, "INCLUDED SOFTWARE"). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, and are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG's Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.cloud.com/legal>.

Copyright © 2001-2025. Cloud Software Group, Inc. All Rights Reserved.