Boundless Integrations

The Boundless Integrations B2Bi XPath Functions provide access to additional xpath functions that deal with strictly B2Bi related functionality. In addition to being available in assign statements and XSLT's, all Boundless Integrations Custom XPath Functions include public methods for access in a Translation Map.

Prerequisites:

- Update 3rd Party Vendor JAR: xalan.jar. Available free with first Boundless Integrations Custom XPath Function license (extends xslt4j version 2_7_19 to allow 50 custom xpath functions instead of 30)
- Install 3rd Party Vendor JAR: boundless_integrations_core.jar. Available free with first Boundless Integrations license
- BoundlessIntegrations.properties.in: Available with any Boundless Integrations license (reminder: run setupfiles(.sh or .cmd) after placing in the properties directory of each node)

Installation:

- Download boundless_integrations_xpath_b2bi.jar: After purchasing the B2Bi XPath Functions license, the jar file will be available for download
- Generate, download, and place into the properties directory of each node the BoundlessIntegrations.properties_b2bixpath_license_ext file and the xpathFunctions.properties_boundlessintegrations_b2bi_ext file
- Stop B2Bi Node(s)
- Place in a location accessible to each node and run bin/install3rdParty(.sh or .cmd)
- Start B2Bi Node(s)

License Update:

- If manually paying for the next year's maintenance, do so.
- After payment has been received (whether automatic or manual), generate, download, and place BoundlessIntegrations.properties_b2bixpath_license_ext file into the properties directory of each node.
- Run bin/setupfile(.sh or .cmd).
- Either bounce each node or execute BNDINT_RefreshCache Business Process.

Get CodeList XPath Function:

The Get CodeList XPath Function will look up a value from a code list stored within B2Bi. It always returns a single String value regardless of use.

Assign/XSLT syntax:

- get-codelist(listName, senderItem, receiverItem, fieldName)
    - listName is the name of the code list
    - senderItem is the sender_item value in the code list (blank for all)
    - receiverItem is the receiver_item value in the code list (blank for all)
    - fieldname is the field to return (e.g. TEXT1)

Translation Map methods:

- Object Name: com.boundlessintegrations.xpath.GetCodeList
    - No constructor parameters
- performSearch(String list, String sndrItem, String rcvrItem, String field)
    - listName is the name of the code list
    - senderItem is the sender_item value in the code list (blank for all)
    - receiverItem is the receiver_item value in the code list (blank for all)
    - fieldname is the field to return (e.g. TEXT1)
    - Returns a String value representing the field value

NOTE: There is a standard and extended rule in the Translation Map for pulling a code list value. However, this function will return a value from the database each time instead of the cache within the JVM. If updating code list automatically, this can remove the need to also manually refresh the code list cache on all nodes.

Search WFD XPath Function:

The Search WFD XPath Function will search the WorkFlow Definitions for the default version of all business processes using an xpath expression. When used in an assign statement or XSLT, it returns a node set with an optionally defined result node and row node. When used in a Translation Map, there are functions to search business processes, get the number of processes found, and getting a specific process from the list.

Assign/XSLT syntax:

- search-wfd(xpath [, result [, row]])
    - xpath must return a logical (boolean) value
    - result defaults to “search”
    - row defaults to “wfd”

Translation Map methods:

- Object Name: com.boundlessintegrations.xpath.SearchWorkFlowDef
  - No constructor parameters
- performSearch(String xpathExpression)
  - Void function (returns no value)
  - Searches business processes using xpathExpression and stores within object
- numWFDs()
  - Returns a Number value representing the number of processes found
- getWFD(Number listNum)
  - Returns a String value representing the requested WorkFlow Name
  - Returns a blank value if listNum is out of bounds
  - 1-based input (e.g. getWFD(1) is the first business process)

NOTE: The most practical use of this xpath function is for researching where specific services and adapters are being utilized or where specific business processes are being called. Therefore, Boundless Integrations also provides a Translation Map which can take a delimited input file and return the list of business processes for each request. This Translation Map can then be run using the map test option from the Map Editor without needing to deploy and constantly revise a business process for new searches.

The source map (BNDINT_Search_WFD) is available on the download page for this license. This map takes a comma separated file with three fields:

- assetType:
  - Service: checks the participant name specified
  - AssignTo: checks the to attribute of an assign statement
  - AssignFrom: checks the from attribute of an assign statement
  - AssignLiteral: checks the literal value of an assign statement
  - Custom: any xpath expression that returns a logical value
- assetName:
  - service name, workflow name, xpath expression, etc.
- matchType (ignored for custom xpath expressions):
  - E: Exact match
  - C: "Contains" match

If searching for a workflow name, use AssignLiteral and the full or partial workflow name with the appropriate match type. However, if the workflow is being assigned from a database field, search using AssignFrom and the field name with a "Contains" match. Since it's searching the BPML code, it will not find any values pulled from a database or other soft-coded value.

Here is an example search csv file for the BNDINT_Search_WFD map:

```
Service,BootstrapService,E
Service,Invoke,C
AssignTo,Before,E
AssignTo,ClientAdapter,C
AssignFrom,b2b-profile-id/text(),E
AssignFrom,SessionToken,C
AssignLiteral,30,E
AssignLiteral,HTTP,C
Custom,"count(//condition[contains(string(),'as2-http')]) > 0",E
```

NOTE: When using the Custom assetType, remember to wrap the xpath expression with double quotes if the expression contains a comma.

Here is an example result file (edited down to fit below):

```
Service,BootstrapService,E
...WFD...,BOOTSTRAP_TEST
...WFD...,TEST_WRAPPER
Service,Invoke,C
...WFD...,ACHEnvelope
...WFD...,AFTRouteSendMessage
AssignTo,Before,E
...WFD...,JAVA_TEST
...WFD...,TestTranslation
AssignTo,ClientAdapter,C
...WFD...,AFTRouteFTPPUT
...WFD...,AFTRouteSFTPPUT
AssignFrom,b2b-profile-id/text(),E
...WFD...,EDIAS2HTTPSend
...WFD...,HTTPAsyncSend
AssignFrom,SessionToken,C
...WFD...,AFTRouteFTPPUT
...WFD...,AFTRouteSFTPPUT
AssignLiteral,30,E
...WFD...,AS2SendAndProcessAsyncMDN
...WFD...,DirectAS2SendAndProcessAsyncMDN
AssignLiteral,HTTP,C
...WFD...,AS2SendASyncMDN
...WFD...,AS2SendAndProcessAsyncMDN
Custom,"count(//condition[contains(string(),'as2-http')]) > 0",E
...WFD...,HTTPAsyncSend
...WFD...,HTTPClientSend
```

Examples:

Get CodeList used in an assign to get TEXT1 based on Sender and Receiver Item:

```
<assign to="LocationID" from="get-codelist('N104', 'ZZVENDOR', '9212345',
'TEXT1')" />
```

Process Data:

```
<ProcessData>
  <LocationID>98765</LocationID>
</ProcessData>
```

Get CodeList used in an assign to get TEXT1 based only on Sender Item:

```
<assign to="LocationID" from="get-codelist('N104', 'ZZVENDOR', '',
'TEXT1')" />
```

Process Data:

```
<ProcessData>
  <LocationID>99999</LocationID>
</ProcessData>
```

Get CodeList used in an assign to get TEXT1 based only on Receiver Item:

```
<assign to="LocationID" from="get-codelist('N104', '', '9212345',
'TEXT1')" />
```

Process Data:

```
<ProcessData>
  <LocationID>98765</LocationID>
</ProcessData>
```

Get CodeList used in an assign to get TEXT1 based on neither Sender nor Receiver Item:

```
<assign to="LocationID" from="get-codelist('N104', '', '', 'TEXT1')" />
```

Process Data:

```
<ProcessData>
  <LocationID>99999</LocationID>
</ProcessData>
```

Get CodeList in a Translation Map to get Customer ID:

Extended Rule on #N104:

```
object getCodeList;
string[50] senderItem, receiverItem;

getCodeList = new ("com.boundlessintegrations.xpath.GetCodeList");
senderItem = #ISA05 + #ISA06;
receiverItem = #N103 + #N104;

#CUST_ID = getCodeList.performSearch("N104", senderItem, receiverItem,
"TEXT1");
```

Search WFD used in an assign to get all processes with default result and row:

```
<assign to="." from="search-wfd(&apos;count(//rule) = 0&apos;)" />
```

Process Data:

```
<ProcessData>
  <search>
    <wfd>Schedule_BPExpirator</wfd>
    <wfd>Schedule_BPRecovery</wfd>
    <wfd>Schedule_CheckExpireService</wfd>
    <wfd>Schedule_DBMonitorService</wfd>
    <wfd>Schedule_DocumentStatsArchive</wfd>
    <wfd>Schedule_MessagePurge</wfd>
    <wfd>Schedule_NonEdiInboundFSAdapter</wfd>
    <wfd>Schedule_NonEdiOutboundFSAdapter</wfd>
    <wfd>Schedule_PartialDocumentCleanUpService</wfd>
    <wfd>Schedule_PerfDataPurgeService</wfd>
    <wfd>Schedule_ProducedMsgPurgeService</wfd>
    <wfd>Schedule_RetentionProcessor</wfd>
    <wfd>Schedule_SAPTidCleaner</wfd>
    <wfd>Schedule_SWIFTNet7AdapterScheduler</wfd>
    <wfd>Schedule_Scheduled_AlertService</wfd>
  </search>
</ProcessData>
```

Search WFD used in an assign to get all pieces with defined result and row:

```
<assign to="." from=" search-wfd(&apos;count(//rule) = 0&apos;,
&apos;NoRules&apos;, &apos;WorkFlow&apos;)" />
```

Process Data:

```
<ProcessData>
  <NoRules>
    <WorkFlow>Schedule_BPExpirator</WorkFlow>
    <WorkFlow>Schedule_BPRecovery</WorkFlow>
    <WorkFlow>Schedule_CheckExpireService</WorkFlow>
    <WorkFlow>Schedule_DBMonitorService</WorkFlow>
    <WorkFlow>Schedule_DocumentStatsArchive</WorkFlow>
    <WorkFlow>Schedule_MessagePurge</WorkFlow>
    <WorkFlow>Schedule_NonEdiInboundFSAdapter</WorkFlow>
    <WorkFlow>Schedule_NonEdiOutboundFSAdapter</WorkFlow>
    <WorkFlow>Schedule_PartialDocumentCleanUpService</WorkFlow>
    <WorkFlow>Schedule_PerfDataPurgeService</WorkFlow>
    <WorkFlow>Schedule_ProducedMsgPurgeService</WorkFlow>
    <WorkFlow>Schedule_RetentionProcessor</WorkFlow>
    <WorkFlow>Schedule_SAPTidCleaner</WorkFlow>
    <WorkFlow>Schedule_SWIFTNet7AdapterScheduler</WorkFlow>
    <WorkFlow>Schedule_Scheduled_AlertService</WorkFlow>
  </search>
</NoRules>
```

Split String in a Translation Map to list workflow names:

Extended Rule on #NAME:

```
object searchWFD;
integer idx, wfCount;

searchWFD = new ("com.boundlessintegrations.xpath.SearchWorkFlowDef");

searchWFD.performSearch("count(//rule) = 0");
wfCount = searchWFD.numWFDs();

idx = 0;

while idx < wfCount do
begin
  idx = idx + 1;
  $groupWF[idx].#WORKFLOW_NAME = searchWFD.getWFD(idx);
end
```

Troubleshooting:

- B2Bi xpath function throws an exception.
  - Check system.log for details. Open an incident with Boundless Integrations if needed.
- All functions return a blank value.
  - Ensure license is up to date.

Support:

Contact Boundless Integrations for any support need. As part of your license, the following support is included at no additional charge:

- Bug fixes
- Installation problems (not including rework of any existing assets)
- Enhancements requests
- General "How Do I" questions

The following services (and any others not explicitly listed above) are available as contract support billed by hour plus any expenses:

- Rework existing business processes/maps/XSLTs
- Custom business processes/maps/XSLTs

Copyright Notice: