



Boundless Integrations

The Boundless Integrations String XPath Functions provide access to additional xpath functions that deal with string inputs. 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_string.jar: After purchasing the String 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_stringxpath_license_ext file and the xpathFunctions.properties_boundlessintegrations_string_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_stringxpath_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.



Boundless Integrations

Split String XPath Function:

The Split String XPath Function will split a string based on a given delimiter. It honors quotation marks (single or double) when parsing. 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 split the string, get the number of pieces, and getting a specific piece.

Assign/XSLT syntax:

- split-string(string, delimiter [, result [, row]])
 - result defaults to "split"
 - row defaults to "piece"

Translation Map methods:

- Object Name: com.boundlessintegrations.xpath.SplitString
 - No constructor parameters
- performSplit(String unparsedString, String delimiter)
 - Void function (returns no value)
 - Parses unparsedString using delimiter and stores within object
- numPieces()
 - Returns a Number value representing the number of pieces created
- getPiece(Number pieceNum)
 - Returns a String value representing the requested piece
 - Returns a blank value if pieceNum is out of bounds
 - 1-based input (e.g. getPiece(1) is the first piece)

Replace String XPath Function:

The Replace String XPath Function will replace some text within a string with new text. It always returns a single String value regardless of use.

Assign/XSLT syntax:

- replace-string(string, replaceText, withText)

Translation Map methods:

- Object Name: com.boundlessintegrations.xpath.ReplaceString
 - No constructor parameters
- performReplace(String originalString, String replaceText, String withText)
 - Returns a String value representing the new value

Boundless Integrations

Convert String XPath Function:

The Convert String XPath Function will convert a string using various modes. It always returns a single String value regardless of use.

Assign/XSLT syntax:

- `convert-string(string, mode)`

Translation Map methods:

- Object Name: `com.boundlessintegrations.xpath.ConvertString`
 - No constructor parameters
- `performConvert(String originalString, String mode)`
 - Returns a String value representing the new value
 - The mode can be:
 - Upper: Converts originalString to upper case
 - Lower: Converts originalString to lower case
 - Proper: Converts originalString to proper case (for proper names)

Examples:

Split String used in an assign to get all pieces with default result and row:

```
<assign to="." from="split-string('1|2|3|4','')"/>
```

Process Data:

```
<ProcessData>
  <split>
    <piece>1</piece>
    <piece>2|3</piece>
    <piece>4</piece>
  </split>
</ProcessData>
```

Split String used in an assign to get all pieces with defined result and row:

```
<assign to="." from="split-string('Adam's Garage','James 'Lucky' Loan Shark','City Taxi','Customer','Name')"/>
```

Boundless Integrations

Process Data:

```
<ProcessData>
  <Customer>
    <Name>Adam's Garage</Name>
    <Name>James "Lucky" Loan Shark</Name>
    <Name>City Taxi</Name>
  </Customer>
</ProcessData>
```

Split String used to get file name from an absolute path:

```
<assign to="file" from="split-string('/d01/share/web/orders.xml',
'/')/piece[last()]/text()" />
```

Process Data:

```
<ProcessData>
  <file>orders.xml</file>
</ProcessData>
```

Split String in a Translation Map to split a name using one extended rule:

Extended Rule on #NAME:

```
object splitString;
integer splitCount;

splitString = new ("com.boundlessintegrations.xpath.SplitString");

splitString.performSplit(#NAME, " ");
splitCount = splitString.numPieces();

if splitCount >= 1 then #FIRST_NAME = splitString.getPiece(1);

if splitCount = 2 then
begin
  #LAST_NAME = splitString.getPiece(2);
end

if splitCount = 3 then
begin
  #MIDDLE_NAME = splitString.getPiece(2);
  #LAST_NAME = splitString.getPiece(3);
end
```

Boundless Integrations

Split String in a Translation Map to split a name using multiple extended rules:

Pre-Session Extended Rule:

```
object splitString;
integer splitCount;

splitString = new ("com.boundlessintegrations.xpath.SplitString");
splitCount = 0;
```

Extended Rule on #NAME:

```
splitString.performSplit(#NAME, " ");
splitCount = splitString.numPieces();
```

Extended Rule on #FIRST_NAME:

```
if splitCount >= 1 then #FIRST_NAME = splitString.getPiece(1);
```

Extended Rule on #MIDDLE_NAME:

```
if splitCount = 3 then #MIDDLE_NAME = splitString.getPiece(2);
```

Extended Rule on #LAST_NAME:

```
if splitCount >= 2 then #LAST_NAME = splitString.getPiece(splitCount);
```

NOTE: The above only works if #NAME, #FIRST_NAME, #MIDDLE_NAME, and #LAST_NAME are in the same loop.

Replace String used in an assign to set sqlText:

```
<assign to="sqlText" from="replace-string(&apos;SELECT * FROM MBX_MAILBOX
WHERE PATH LIKE '?'&apos;;, &apos;?&apos;;, &apos;/AS2/%&apos;)" />
```

Process Data:

```
<ProcessData>
  <sqlText>SELECT * FROM MBX_MAILBOX WHERE PATH LIKE '/AS2/'</sqlText>
</ProcessData>
```

Replace String used in a Translation Map to set a name:

```
object replaceString;
replaceString = new ("com.boundlessintegrations.xpath.ReplaceString");
#GREETING = replaceString.performReplace(#GREETING, "%NAME%", "John");
```



Boundless Integrations

Convert String used in an assign to convert to lower case:

```
<assign to="file" from="convert-string(&apos;ORDERS.XML&apos;;,
&apos;Lower&apos;)" />
```

Process Data:

```
<ProcessData>
  <file>orders.xml</file>
</ProcessData>
```

Convert String used in an assign to convert to upper case:

```
<assign to="custId" from="convert-string(&apos;abc123&apos;;,
&apos;Upper&apos;)" />
```

Process Data:

```
<ProcessData>
  <custId>ABC123</custId>
</ProcessData>
```

Convert String used in an assign to convert to proper case:

```
<assign to="custName" from="convert-string(&apos;james mcdonnell&apos;;,
&apos;Proper&apos;)" />
```

Process Data:

```
<ProcessData>
  <custId>James McDonnell</custId>
</ProcessData>
```

Convert String used in a Translation Map to change to lower case:

```
object convertString;
convertString = new ("com.boundlessintegrations.xpath.ConvertString");
#ITEM = convertString.performConvert(#ITEM, "Lower");
```

Convert String used in a Translation Map to change to upper case:

```
object convertString;
convertString = new ("com.boundlessintegrations.xpath.ConvertString");
#LINE_TYPE = convertString.performConvert(#LINE_TYPE, "Upper");
```



Boundless Integrations

Convert String used in a Translation Map to change to proper case:

```
object convertString;  
convertString = new ("com.boundlessintegrations.xpath.ConvertString");  
#NAME = convertString.performConvert(#NAME, "Proper");
```

Troubleshooting:

- String 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:

© Copyright 2014 Boundless Integrations, LLC. All rights reserved.