



Boundless Integrations

The Boundless Integrations Bootstrap Service provides the ability to bootstrap new processes in a variety of ways. This greatly reduces the need to have processes with a large number of steps or needing to carry a large Process Data.

Prerequisites:

- Install 3rd Party Vendor JAR: zip4j_1.3.2.jar. Available free under the Apache License, Version 2.0 (<http://www.lingala.net/zip4j/download.php>)
- 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)
- Make note of any hot fixes currently installed and make sure to have a copy of the jar file available

Installation:

- Download Bootstrap_1020402.0.0.jar: After purchasing the Bootstrap Service license, the jar file will be available for download
- Generate, download, and place into the properties directory of each node the BoundlessIntegrations.properties_bootstrap_ext.in file
- Stop B2Bi Node(s)
- Place in a location accessible to each node and run bin/installService(.sh or .cmd)
- Reapply latest hot fix (as installing new services will remove any hot fixes)
- If on a cluster environment, perform the previous two steps on all nodes before running bin/startCluster # (where # is the node number) starting with Node 1.
- Check properties/lang/en directory for Bootstrap_en.properties file. If not present, download and copy to that directory. If other languages are available, you may do the same copy to the appropriate language directory for the appropriate language properties file.
- Start B2Bi Node(s)
- Configure at least one Bootstrap Service within B2Bi (see below)

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_bootstrap_ext.in 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

Configure Bootstrap Service:

- Select Default Mode:
 - XML Document: Bootstrap based on an XML File
 - Zip Document: Bootstrap based on a Zip Document with XML rules
 - TAR Document: Bootstrap based on a TAR Document with XML rules
 - SQL Query: Bootstrap based on a SQL SELECT Query
 - Document Extract Map: Run DocExtract Service and bootstrap each split file

Workflow Parameters:

- mode:
 - XML: Bootstrap based on an XML File
 - Zip: Bootstrap based on a Zip Document with XML rules
 - TAR: Bootstrap based on a TAR Document with XML rules
 - SQL: Bootstrap based on a SQL SELECT Query
 - DocExtract: Run DocExtract Service and bootstrap each split file
- BootstrapBP:
 - Name of business process to start for each document
- delayMS:
 - Time in milliseconds to sleep between documents
- loadBalance:
 - Unbalanced: Do not manually load balance (B2Bi might)
 - Parent WF: Use execution settings of parent process
 - Child WF: Use execution settings of process to start
 - Execution Role: Use one or more execution roles (comma-delimited)
 - Node: Use one or more nodes (comma-delimited)
- balanceValue:
 - For Execution Role and Node options, the list of values
- userToken:
 - User account to use when executing child process
- password:
 - Password used for password-protected Zip Files in Zip mode
- filter:
 - XPath filter for XML mode to select XML nodes to bootstrap
- poolName:
 - Database pool in B2Bi for SQL mode query
- sqlText:
 - SELECT Statement for SQL mode
- DocExtractMapList:
 - List of maps for DocExtract Service (all other DocExtract parms available)



Boundless Integrations

- XML Mode Parameters:
 - PrimaryDocument (must be XML format)
 - BootstrapBP
 - delayMS
 - loadBalance
 - balanceValue
 - userToken
 - filter

- Zip Mode Parameters:
 - PrimaryDocument (must be an archive file which includes BootstrapRules.xml)
 - password
 - BootstrapBP
 - delayMS
 - loadBalance
 - balanceValue
 - userToken

- TAR Mode Parameters:
 - PrimaryDocument (must be an archive file which includes BootstrapRules.xml)
 - BootstrapBP
 - delayMS
 - loadBalance
 - balanceValue
 - userToken

- SQL Mode Parameters:
 - BootstrapBP
 - delayMS
 - loadBalance
 - balanceValue
 - userToken
 - poolName
 - sqlText

- DocExtract Mode Parameters:
 - PrimaryDocument (any format for DocExtract)
 - BootstrapBP
 - delayMS
 - loadBalance
 - balanceValue
 - userToken
 - DocExtractMapList (plus any other DocExtract Service parameters)



Boundless Integrations

In all cases, `message_to_child` is passed to the child process. For the Bootstrap Service, anything not part of the XML, Zip, SQL, or DocExtract default output needs to be passed using `message_to_child`.

Examples:

XML mode, example `orders.xml` file:

```
<Orders>
  <Order>
    <CustomerId>12345</CustomerId>
    <OrderDate>2014-09-27</OrderDate>
    <PurchaseOrder>999999</PurchaseOrder>
    <Line>
      <Item>ABC123</Item>
      <OrderQty>5</OrderQty>
    </Line>
    <Line>
      <Item>DEF456</Item>
      <OrderQty>7</OrderQty>
    </Line>
  </Order>
  <Order>
    <CustomerId>15263</CustomerId>
    <OrderDate>2014-09-28</OrderDate>
    <PurchaseOrder>888888</PurchaseOrder>
    <Line>
      <Item>DEF456</Item>
      <OrderQty>2</OrderQty>
    </Line>
  </Order>
  <Order>
    <CustomerId>12345</CustomerId>
    <OrderDate>2014-09-28</OrderDate>
    <PurchaseOrder>777777</PurchaseOrder>
    <Line>
      <Item>ABC123</Item>
      <OrderQty>3</OrderQty>
    </Line>
    <Line>
      <Item>DEF456</Item>
      <OrderQty>9</OrderQty>
    </Line>
    <Line>
      <Item>GHI789</Item>
      <OrderQty>12</OrderQty>
    </Line>
  </Order>
</Orders>
```



Boundless Integrations

XML mode, example schedule.xml file:

```
<Schedules>
  <Schedule interval="5">
    <Workflow>BNDINT_CheckCluster</Workflow>
    <Nodes>node1</Nodes>
    <userToken>admin</userToken>
  </Schedule>
  <Schedule interval="1">
    <Workflow>BNDINT_MailboxRules</Workflow>
    <Nodes>node1,node2</Nodes>
    <userToken>aft_user</userToken>
    <MailboxPattern>/AS2/Inbound/*</MailboxPattern>
    <NamePattern>*</NamePattern>
  </Schedule>
  <Schedule interval="15">
    <Workflow>BNDINT_MailboxRules</Workflow>
    <Nodes>node1,node2</Nodes>
    <userToken>aft_user</userToken>
    <MailboxPattern>/VAN/Inbound</MailboxPattern>
    <NamePattern>*</NamePattern>
  </Schedule>
  <Schedule interval="5">
    <Workflow>BNDINT_Metrics</Workflow>
    <Nodes>node1</Nodes>
    <userToken>admin</userToken>
    <MetricType>Memory</MetricType>
  </Schedule>
  <Schedule interval="5">
    <Workflow>BNDINT_Metrics</Workflow>
    <Nodes>node2</Nodes>
    <userToken>admin</userToken>
    <MetricType>Memory</MetricType>
  </Schedule>
  <Schedule interval="10">
    <Workflow>BNDINT_CheckOrphanDocuments</Workflow>
    <Nodes>node2</Nodes>
    <userToken>admin</userToken>
  </Schedule>
</Schedules>
```



Boundless Integrations

XML mode, all orders in orders.xml file

```
<operation name="Bootstrap Service">
  <participant name="BootstrapService"/>
  <output message="outmsg">
    <assign to="mode">XML</assign>
    <assign to="filter">/Orders/Order</assign>
    <assign to="BootstrapBP">Order_Load</assign>
    <assign to="loadBalance">Node</assign>
    <assign to="balanceValue">node1,node2</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

This will randomly select node1 or node2 as the starting node and then alternate between node1 and node2 for each order and start a new Order_Load process with the contents of the Order node in ProcessData. The orders.xml file is PrimaryDocument.

XML mode, orders in orders.xml file for CustomerId 12345 only

```
<operation name="Bootstrap Service">
  <participant name="BootstrapService"/>
  <output message="outmsg">
    <assign to="mode">XML</assign>
    <assign to="filter">/Orders/Order[CustomerId/text() = "12345"]</assign>
    <assign to="BootstrapBP">Order_Load</assign>
    <assign to="loadBalance">Child WF</assign>
    <assign to="delayMS">5000</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

This will pick the node(s) that Order_Load is configured to run on and start a new Order_Load process with the contents of the Order node in ProcessData for just the two Order nodes with a CustomerId of 12345. The 5,000 millisecond (5 second) delay is done between the two orders. The first order is immediately launched and then the delay is applied before starting the second order. There is no delay after the second order for the parent process to continue running. The orders.xml file is PrimaryDocument.



Boundless Integrations

XML mode, schedules in schedule.xml with an interval of 1 or 5:

```
<operation name="Bootstrap Service">
  <participant name="BootstrapService"/>
  <output message="outmsg">
    <assign to="mode">XML</assign>
    <assign to="filter">//Schedule[number(@interval) &lt;= 5]</assign>
    <assign to="BootstrapBP">Workflow/text()</assign>
    <assign to="loadBalance">Node</assign>
    <assign to="balanceValue">Nodes/text()</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

Since userToken was not specified as a parameter, the Bootstrap Service looks for a userToken node in the XML file and uses it if it's available.

The processes that are started:

- BNDINT_CheckCluster on node1 as admin
- BNDINT_MailboxRules on node1 or node2 (randomly) as aft_user
 - /AS2/Inbound/* is also passed as MailboxPattern
 - * is also passed as NamePattern
- BNDINT_Metrics on node1 as admin
 - Memory is also passed as MetricType
- BNDINT_Metrics on node2 as admin
 - Memory is also passed as MetricType

The interval value on the Schedule node is not passed to the child process as it is an attribute on the node being bootstrapped. However, any attributes on any node under the Schedule node will be passed to the child process. When specifying values within the XML file for the Bootstrap Service to use, most values are "sticky" in that the value from previous nodes will stay in effect until changed. The exception is the load balancing which will attempt to not change unless "Child WF" is specified and the workflow name is specified in the XML. Each time load balancing is changed, it randomly picks the first node (if multiple nodes are specified). This might make the Bootstrap Service appear to not load balance as a random selection could pick the same node multiple times in a row. The schedule.xml file is PrimaryDocument.



Boundless Integrations

Zip or TAR mode example BootstrapRules.xml:

```
<Bootstrap>
  <File>
    <BootstrapBP>App_Outbound_Handler</BootstrapBP>
    <userToken>aft_user</userToken>
    <FileName>Invoices.csv</FileName>
    <DocExtractMapList>DocExtract810App</DocExtractMapList>
  </File>
  <File>
    <BootstrapBP>App_Outbound_Handler</BootstrapBP>
    <userToken>aft_user</userToken>
    <FileName>Orders.csv</FileName>
    <DocExtractMapList>DocExtract850App</DocExtractMapList>
  </File>
  <File>
    <BootstrapBP>App_Outbound_Handler</BootstrapBP>
    <userToken>aft_user</userToken>
    <FileName>Shipments.csv</FileName>
    <DocExtractMapList>DocExtract856App</DocExtractMapList>
  </File>
  <File>
    <BootstrapBP>Outbound_Web_Service</BootstrapBP>
    <loadBalance>ExecRole</loadBalance>
    <balanceValue>Communication</balanceValue>
    <userToken>admin</userToken>
    <FileName>WebCustomers.xml</FileName>
    <xsltName>Customers2WebSevice</xsltName>
    <wsdlName>ToWeb</wsdlName>
    <operation>LoadCustomer</operation>
  </File>
  <File>
    <BootstrapBP>Outbound_Web_Service</BootstrapBP>
    <delayMS>30000</delayMS>
    <userToken>admin</userToken>
    <FileName>WebParts.xml</FileName>
    <xsltName>Parts2WebSevice</xsltName>
    <wsdlName>ToWeb</wsdlName>
    <operation>LoadParts</operation>
  </File>
</Bootstrap>
```


Boundless Integrations

Zip mode:

```
<operation name="Bootstrap Service">
  <participant name="BootstrapService"/>
  <output message="outmsg">
    <assign to="mode">Zip</assign>
    <assign to="password">P@ssw0rd</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

NOTE: Password is only needed if the archive file is password protected. TAR files cannot be password protected. If that TAR file is compressed, it must be uncompressed first.

The BootstrapRules.xml file is required to be included in the archive file with a root node of Bootstrap and target node of File. The FileName node is the only required node under each File node provided BootstrapBP is specified on the Bootstrap Service as a literal value. This mode allows multiple individual files to be transmitted as a single archive with the necessary information for the Bootstrap Service to start a process for each file. Any additional nodes under the File node is also passed to the process. The example BootstrapRules.xml file includes additional nodes. NOTE: The delayMS is only specified on the last File node. This causes there to be a delay before the last file is processed. If delayMS is specified on the first File node, it is ignored. The "sticky" settings apply to BootstrapRules.xml like any other XML file processed in XML mode. The archive file is PrimaryDocument.

SQL mode, perform cache refresh on all nodes:

```
<operation name="Bootstrap Service">
  <participant name="BootstrapService"/>
  <output message="outmsg">
    <assign to="mode">SQL</assign>
    <assign to="poolName">oraclePool</assign>
    <assign to="sqlText">SELECT NODE_NAME FROM OPS_NODE_INFO</assign>
    <assign to="BootstrapBP">BNDINT_CacheRefresh</assign>
    <assign to="loadBalance">Node</assign>
    <assign to="balanceValue">NODE_NAME/text()</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

Boundless Integrations

Selects all records in OPS_NODE_INFO and starts BNDINT_CacheRefresh on the node specified by NODE_NAME. NOTE: The node can be specified as either NODE_NAME/text() or just NODE_NAME. However, it is not a normal XPath expression (i.e. the expression string(NODE_NAME) will not work as desired).

In this example, all records are pulled from OPS_NODE_INFO. If trying to replicate this example, it is recommended to also check the node status and node type (only ASI nodes process workflows).

SQL mode, process from a database list table of customers:

```
<operation name="Bootstrap Service">
  <participant name="BootstrapService"/>
  <output message="outmsg">
    <assign to="mode">SQL</assign>
    <assign to="poolName">ERP</assign>
    <assign to="sqlText">SELECT * FROM CUSTOMER_LIST</assign>
    <assign to="BootstrapBP">BNDINT_CheckCustomerConnection</assign>
    <assign to="delayMS">500</assign>
    <assign to="loadBalance">Node</assign>
    <assign to="balanceValue">node1,node2</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

Selects all records in CUSTOMER_LIST and starts BNDINT_CheckCustomerConnection

There is a half second delay between each customer and alternates between node1 and node2. All fields on CUSTOMER_LIST are placed in ProcessData for each process. The CUSTOMER_LIST table is in the database identified by the ERP database pool.



Boundless Integrations

DocExtract mode:

```
<operation name="Bootstrap Service">
  <participant name="BootstrapService"/>
  <output message="outmsg">
    <assign to="mode">DocExtract</assign>
    <assign to="BootstrapBP">Outbound_810_Handler</assign>
    <assign to="loadBalance">Node</assign>
    <assign to="balanceValue">node1,node2</assign>
    <assign to="DocExtractMapList">DocExtract810</assign>
    <assign to="BatchLikeDocuments">YES</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

Executes a DocExtract Service on PrimaryDocument using the DocExtract810 map and batches like documents. The Bootstrap Service then bootstraps each split document to Outbound_810_Handler alternating between node1 and node2 with no delay.

The DocExtract option allows for the DocExtract and ForEachDocument Services to be combined with the result in starting a new process for each split document. This accomplishes two things: reduces the size of ProcessData by having each process only contain the split document it needs and reduces the number of steps in the parent process. Both of these savings will increase B2Bi performance significantly. If ProcessData is large, each step slows down; and, more steps add significantly to the size of the database.

With the use of any DocExtract Service parameters, the DocExtract option should be able to fit into existing processes. If those processes are not starting new processes for each document, it will require the processing steps to be separated into a new process that can be called by the Bootstrap Service.

The Bootstrap Service does not support inline or sync process invokes. If one of these types of invocations are required, the Bootstrap Service is not a good fit.



Boundless Integrations

Troubleshooting:

- Bootstrap service throws an exception.
 - Check system.log for details. Open an incident with Boundless Integrations if needed.
- No processes are started.
 - Check workflow to ensure all parameters were passed.
 - Check system.log to see if the license is invalid. If not, see License Update session to get a valid license. NOTE: If your maintenance is up to date, you can regenerate a valid license file for the entire 13 month period.
- Bootstrap service is not load balancing.
 - Ensure all desired nodes are available at the time the workflow was running. An inactive node is not considered for load balancing and will be excluded automatically. If all requested nodes for load balancing are inactive (including all nodes in an execution role), the Bootstrap service reverts to unbalanced mode.

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.