# VersaLex
# Integration Web Service

November 2013

Cleo

Move View Act™


email sales@cleo.com

phone 815.282.7600

fax 815.654.8294

web cleo.com

## Disclaimer

# Table of Contents

# VersaLex Integration Web Service

The general purpose VersaLex web service can be used to initiate trading partner mailbox sends and receives through any VersaLex protocol. Similar to the Java API IMailboxController interface, the web service provides for send(), receive() and sendAndReceive() operations. A delete() operation is also included for confirming receipt of a payload file after a receive().

The VersaLex web service URL is:

http(s)://VersaLexComputer:http(s)Port/services/versalexws

For example, if connecting to the web service from the same computer and using default VersaLex HTTP 5080, the web service URL would be:

http://localhost:5080/services/versalexws

# Operations

## send

String = **send**(String password, String[] mailbox, String xml) <- when **Use arrays** is checked
String = **send**(String password, String host, String mailbox, String xml) <- when **Use arrays** is unchecked

Sends outgoing payload using the specified mailbox

**Parameters (with arrays):**
password – the web service password
mailbox – the VersaLex host and mailbox alias
xml - xml containing the file(s) to be sent

**Parameters (without arrays):**
password – the web service password
host -- the VersaLex host alias
mailbox – the VersaLex mailbox alias
xml - xml containing the file(s) to be sent

**Returns:**
xml containing the result of the send

# receive

String = **receive**(String password, String[] mailbox)

Receives incoming payload using the specified mailbox

**Note:** When **WSDL 1.2 compliant** is checked, this variant is not available. Use receive(password, mailbox[], xml) (when **Use arrays** is checked) or receive(password, host, mailbox, xml) (when **Use arrays** is unchecked) instead.

**Parameters:**
password – the web service password
mailbox – the VersaLex host and mailbox alias

**Returns:**
xml containing the result of the receive, including the file(s) received

---

String = **receive**(String password, String[] mailbox, String xml) <- when **Use arrays** is checked
String = **receive**(String password, String host, String mailbox, String xml) <- when **Use arrays** is unchecked

Receives incoming payload using the specified mailbox

**Parameters (with arrays):**
password – the web service password
mailbox – the VersaLex host and mailbox alias
xml - xml containing optional receive settings

**Parameters (without arrays):**
password – the web service password
host – the VersaLex host alias
mailbox – the VersaLex mailbox alias
xml - xml containing optional receive settings

**Returns:**
xml containing the result of the receive, including the file(s) received

## sendAndReceive

String = **sendAndReceive** (String password, String[] mailbox, String xml) <- when **Use arrays** checked
String = **sendAndReceive** (String password, String host, String mailbox, String xml) <- when **Use arrays** unchecked

Sends outgoing payload and receives incoming payload using the specified mailbox

**Parameters (with arrays):**
password – the web service password
mailbox – the VersaLex host and mailbox alias
xml - xml containing the file(s) to be sent and optionally any receive settings

**Parameters (without arrays):**
password – the web service password
host – the VersaLex host alias
mailbox – the VersaLex mailbox alias
xml - xml containing the file(s) to be sent and optionally any receive settings

**Returns:**
xml containing the result of the send and receive, including the file(s) received

---

## delete

String = **delete**(String password, String xml)

Deletes previously received payload

Parameters:
password – the web service password
xml - xml containing the file path(s) to be deleted

Returns:
xml containing the result of the delete

*The web service password is configuration in VersaLex by going to Local Listener Web Service.

# XML Layout

The general layout of the XML string is the same for both the xml string parameter and the returned xml string:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Webservice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="webservice.xsd">
  <!-- Action name is either "send", "receive", or "delete"  -->
  <!-- Optional base attribute overrides use of default send or receive action  -->
  <Action name="send" base="mysend">

    <!-- Optional override of base send or receive commands  -->
    <!-- Command syntax dictated by protocol  -->
    <Command>...</Command>

    <!-- Optional send or receive parameters -->
    <!-- Available parameters are dictated by protocol  -->
    <!-- Put.Destination and Get.Source can be used to set the PUT command destination
or GET command source  -->
    <Parameters>
      <Subject>...</Subject>
      <Put.Destination>...</Put.Destination>
    </Parameters>

    <!-- Outgoing or incoming file  -->
    <File name="test.edi">
      <!-- Optional file size in bytes  -->
      <Size>...</Size>
      <!-- File path  -->
      <Path>...</Path>
      <!-- Base64 encoded file content if not too large to pass thru web service (<1mb)  -
->
      <Content>...</Content>
      <!-- Optional outgoing file Content-type  -->
      <Content-type>...</Content-type>
      <!-- Optional outgoing file ExternalID (for transfer logging)  -->
      <ExternalID>...</ExternalID>
      <!-- Optional incoming file MessageID  -->
      <MessageID>...</MessageID>
    </File>

    <!-- Copy of result from logs/VersaLex.xml -->
    <Result text="Success">...</Result>
  </Action>
</Webservice>
```

* The webservice.xsd schema can be found in the VersaLex webserver\WEB-INF\schemas directory.

* Possible <Result> text= attribute values include “Success”, “Warning”, “Error”, “Exception”, or “Interrupted”.
For the **send()** operation,

- Only <Action name=”send”> elements in the xml are processed.

For the **receive()** operation,

- An <Action name=”receive”> element in the xml parameter is not required, and if not present, the receive() operation will insert one in the returned xml. However, there are two optional receive attributes whose use does require an <Action name=”receive”> element in the xml parameter:
collectOnly=”true” indicates that only already received files should be returned. For peer-to-peer protocols like AS2, collectOnly is already implicitly true.
collect=”false” indicates that only newly received files should be returned. For peer-to-peer protocols like AS2, collect is not applicable.
filePathOnly=”true” indicates that base64 content should not be included in the returned <File> elements.

- AS2, ebMS, and SMTP support multipart messages (i.e. more than one file bundled together in an outgoing or incoming message). If the VersaLex Local Listener Advanced property “Do Not Create Inbox Subdirectories For Multiple Payload Files” is left off, then each <File> returned for a received multipart message will include a <MessageID> element containing the same ID value. Each <MessageID> element will also have the following attributes:
index= which starts at 0 and increments with each multipart file
parts= which indicates the number of parts

- The returned <Result> element for an <Action name=“receive”> will have a moreFiles=”true|false” attribute. If more files are available than could be returned (because content size exceeded 1MB or total number of files received exceeded 1K), the value of the attribute will be “true”.

- If multiple files were received from the trading partner as part of this action invocation, since only one <Result> element is returned, it reflects the result of the last file received, whether successful or unsuccessful.

For the **delete()** operation,

- Only <Action name=”delete”> elements in the xml are processed.

- The delete() operation should always be used in tandem with a receive(); otherwise the same payload can be received more than once.

# AS2 send example

String = **send**(String password, String[] mailbox, String xml)

Parameters:

password = ********
mailbox = {"Looptest AS2", "myMailbox"}
xml =

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Webservice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="webservice.xsd">
  <Action name="send">
    <Parameters>
      <Subject>test</Subject>
    </Parameters>
    <File name="test.edi">
      <Size>1533</Size>
      <Content>
SVNBKjAwKiAgICAgICAgICAqMDAqICAgICAgICAgICpaWipFREkgU2VuZGVyICAgIC
AqWloqRURJ
...
ClNFKjIyKjAwMDA0MDAwMQ0KR0UqMSowMDAwMDAwMDQNCklFQSoxKjAwMDAw
MDAwMQ0K
      </Content>
      <Content-type>application/edi-x12</Content-type>
    </File>
  </Action>
</Webservice>
```

Returns:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Webservice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="webservice.xsd">
  <Action name="send">
    <File name="test.edi">
    <Result text="Success" mdn="AS2\mdn\received\CLEOAS2-
20091027_135903800@Looptest_Looptest-T.mdn" messageID="CLEOAS2-
20091027_135903800@Looptest_Looptest-T" source="*stream*\test.edi"
direction="Local-&gt;Host" number="1 of 1" fileSize="1533" transferID="AS2-
20091027_085903785-T" bytes="3594" seconds="0.01">Sent and Received Message
Integrity Check codes match</Result>
  </Action>
</Webservice>
```

# AS2 receive/delete example

String = **receive**(String password, String[] mailbox)

Parameters:

password = ********
mailbox = {“Looptest AS2”, “myMailbox”}

Returns:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Webservice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="webservice.xsd">
  <Action name="receive">
    <File name="test.edi">
      <Size>1533</Size>
      <Path>inbox\</Path>
      <Content>
SVNBKjAwKiAgICAgICAgICAqMDAqICAgICAgICAgICpaWipFREkgU2VuZGVyICAgICAqWloqRURJ
...
ClNFKjIyKjAwMDA0MDAwMQ0KR0UqMSowMDAwMDAwMDQNCklFQSoxKjAwMDAwMDAwMQ0K
      </Content>
    </File>
    <Result text="Success" />
  </Action>
</Webservice>
```

The xml string passed to delete() is simply a modification of the xml string returned by receive():

- The <Action name=”receive”> attributes changed to name=“delete”

- The <Size>, <Content>, and <Result> elements stripped out

String = **delete**(String password, String xml)

Parameters:

password = ********

xml =

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Webservice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="webservice.xsd">
  <Action name="delete">
    <File name="test.edi">
      <Path>inbox\</Path>
    </File>
  </Action>
</Webservice>
```

Returns:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<Webservice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="webservice.xsd">
  <Action name="delete">
    <File name="test.edi">
      <Path>inbox\</Path>
    </File>
    <Result text="Success" />
  </Action>
</Webservice>
```