

# WmEDISamples

Built-In Services

Version 9.12

November 2017

This document applies to *webMethods Module for EDI 9.12* and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2017 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

*ESTD-EDI-BIS-912-20171123*

## CONTENTS

<b>1</b>	<b>WmEDISamples Package</b>	<b>6</b>
<b>2</b>	<b>EDIFFSchema</b>	<b>7</b>
2.1	EDIFFSchema.FlatFile:CSVSchema	7
2.2	EDIFFSchema.FlatFile:Dictionary	7
2.3	EDIFFSchema.FlatFile:Example1	7
2.4	EDIFFSchema.FlatFile:Example2	7
2.5	EDIFFSchema.UNEDIFACT.V98A:Dictionary	7
2.6	EDIFFSchema.UNEDIFACT.V98A:TORDERS	7
2.7	EDIFFSchema.X12.V4010:Dictionary	8
2.8	EDIFFSchema.X12.V4010:T810	8
2.9	EDIFFSchema.X12.V4010:T850	8
2.10	EDIFFSchema.X12.V4010:T997	8
<b>3</b>	<b>Tutorial.EDIttoXML</b>	<b>9</b>
3.1	Tutorial.EDIttoXML:EDIttoXMLPOMap	9
3.2	Tutorial.EDIttoXML:dateParse	9
3.3	Tutorial.EDIttoXML:processEDI850_4010	10
<b>4</b>	<b>Tutorial.EDIttoXML:segmentMapping</b>	<b>11</b>
<b>5</b>	<b>Tutorial.Flatfile</b>	<b>12</b>
5.1	Tutorial.FlatFile:parseCsvInputS	12
5.2	Tutorial.FlatFile:parseFlatFile	12
5.3	Tutorial.FlatFile:parseFlatFile2	12
<b>6</b>	<b>Tutorial.Records</b>	<b>14</b>
6.1	Tutorial.Records:POSource	14
6.2	Tutorial.Records:POTarget	14
6.3	Tutorial.Records:rec_FF	14
6.4	Tutorial.Records:rec_FF2	14
6.5	Tutorial.Records:rec_XCEL	14
<b>7</b>	<b>Tutorial.XMLtoEDI</b>	<b>15</b>
7.1	Tutorial.XMLtoEDI:XMLPOtoEDIMap	15
7.2	Tutorial.XMLtoEDI:XMLdateConversionToDateTime	15
7.3	Tutorial.XMLtoEDI:appendSegment	16
7.4	Tutorial.XMLtoEDI:conditionalTruncate	16
7.5	Tutorial.XMLtoEDI:processXMLSource	16

7.6	Tutorial.XMLtoEDI:replaceControlChar .....	17
<b>8</b>	<b>Tutorial.XMLtoEDI.segmentMapping .....</b>	<b>18</b>
8.1	Tutorial.XMLtoEDI.segmentMapping:XMLToBEG .....	18
8.2	Tutorial.XMLtoEDI.segmentMapping:XMLToDTM .....	18
8.3	Tutorial.XMLtoEDI.segmentMapping:XMLToFOB .....	18
8.4	Tutorial.XMLtoEDI.segmentMapping:XMLToGS .....	18
8.5	Tutorial.XMLtoEDI.segmentMapping:XMLToISA .....	18
8.6	Tutorial.XMLtoEDI.segmentMapping:XMLToN1 .....	18
8.7	Tutorial.XMLtoEDI.segmentMapping:XMLToPO1_CTT .....	18
8.8	Tutorial.XMLtoEDI.segmentMapping:XMLToREF .....	19
8.9	Tutorial.XMLtoEDI.segmentMapping:XMLToTD5 .....	19
8.10	Tutorial.XMLtoEDI.segmentMapping:XMLToTXI .....	19
<b>9</b>	<b>Tutorial.properties .....</b>	<b>21</b>
9.1	Tutorial.properties:getProperties .....	21
<b>10</b>	<b>Sample Flow Services .....</b>	<b>22</b>
10.1	sampleServices:Iterator810 .....	22
10.2	sampleServices:UNEDIFACTtoValues .....	23
10.3	sampleServices:X12toValues .....	23
<b>11</b>	<b>Sample Business Process .....</b>	<b>25</b>
11.1	wm.b2b.editn.sample.setup:setupSample .....	25
11.2	wm.b2b.editn.sample.setup:setupEDIBizDocTypes .....	26
11.3	wm.b2b.editn.sample.setup:setupXMLBizDocTypes .....	26
11.4	wm.b2b.editn.sample.setup:setupProfile .....	27
11.5	wm.b2b.editn.sample.receiver:Process850 .....	27
11.6	wm.b2b.editn.sample.receiver:Send997 .....	28
11.7	wm.b2b.editn.sample.receiver:Generate997 .....	28
11.8	wm.b2b.editn.sample.receiver:Send855 .....	28
11.9	wm.b2b.editn.sample.receiver:ProcessEnvelope .....	29
11.10	wm.b2b.editn.sample.util:getLastDocuments .....	29
11.11	wm.b2b.editn.sample.util:enable .....	30
11.12	wm.b2b.editn.sample.util:return997 .....	30
11.13	wm.b2b.editn.sample.util:getBizDocFromEvent .....	31
11.14	wm.b2b.editn.sample.util:checkSetup .....	31
11.15	wm.b2b.editn.sample.util:receive .....	32
11.16	wm.b2b.editn.sample.util:deliver .....	32
11.17	wm.b2b.editn.sample.sender:Process997 .....	32

---

11.18	wm.b2b.editn.sample.sender:Log850 .....	33
11.19	wm.b2b.editn.sample.sender:SendEnvelope .....	33
11.20	wm.b2b.editn.sample.sender:GenerateEnvelope .....	33
11.21	wm.b2b.editn.sample.sender:Process855 .....	34
11.22	wm.b2b.editn.sample.sender:start .....	34
<b>12</b>	<b>Wm.EDIINT.sample .....</b>	<b>35</b>
12.1	Wm.EDIINT.sample:send .....	35
<b>13</b>	<b>Wm.EDIforTN EANCOM .....</b>	<b>38</b>
13.1	wm.b2b.editn.eancom:generateAndProcessAUTACK .....	38

## 1 WmEDISamples Package

The WmEDISamples package is available for download under *Tips, Tricks, and Code* in the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. The samples in this package are certified and tested by Software AG.

## 2 EDIFFSchema

The EDIFFSchema folder contains flat file schemas and dictionaries that other services in the WmEDISamples package can use to validate EDI documents and sample flat file schemas.

### 2.1 EDIFFSchema.FlatFile:CSVSchema

Sample flat file schema for a comma separated values (CSV) flat file.

For more information about flat file schemas and dictionaries, see *Flat File Schema Developer's Guide*.

### 2.2 EDIFFSchema.FlatFile:Dictionary

Sample flat file schema dictionary.

For more information about flat file schemas and dictionaries, see *Flat File Schema Developer's Guide*.

### 2.3 EDIFFSchema.FlatFile:Example1

Sample flat file schema.

For more information about flat file schemas and dictionaries, see *Flat File Schema Developer's Guide*.

### 2.4 EDIFFSchema.FlatFile:Example2

Sample flat file schema.

For more information about flat file schemas and dictionaries, see *Flat File Schema Developer's Guide*.

### 2.5 EDIFFSchema.UNEDIFACT.V98A:Dictionary

Sample flat file schema dictionary.

For more information about flat file schemas and dictionaries, see *Flat File Schema Developer's Guide*.

### 2.6 EDIFFSchema.UNEDIFACT.V98A:TORDERS

Flat file schema to parse and validate a UN/EDIFACT 98A ORDERS document.

## 2.7 EDIFFSchema.X12.V4010:Dictionary

Sample flat file schema dictionary.

For more information about flat file schemas and dictionaries, see *Flat File Schema Developer's Guide*.

## 2.8 EDIFFSchema.X12.V4010:T810

Flat file schema to parse and validate the ANSI X12, version 4010, and version 810 documents.

## 2.9 EDIFFSchema.X12.V4010:T850

Flat file schema to parse and validate the ANSI X12, version 4010, and version 850 documents.

## 2.10 EDIFFSchema.X12.V4010:T997

Flat file schema to parse and validate the ANSI X12, version 4010, and version 997 documents.



### 3 Tutorial.EDtoXML

The services in this folder illustrate how to map an EDI document to an XML document.

#### 3.1 Tutorial.EDtoXML:EDtoXMLPOMap

Illustrates how to convert an EDI document into a different format. (For example, an XML PO).

##### Input Parameters:

Parameter	Description
<i>EDValues</i>	Document Source EDI transaction set.

##### Output Parameters:

Parameter	Description
<i>POTarget</i>	Document Target XML document.
<i>mappingStatus</i>	String Status of the mapping result.

##### Usage Notes:

This is the main ANSI X12-to-XML target mapping service. This service uses the mapping services in the Tutorial.EDtoXML:segmentMapping folder described in the following section.

##### See Also:

Tutorial.Records:POTarget

#### 3.2 Tutorial.EDtoXML:dateParse

Parses out the date from YYMMDD or YYYYMMDD to the pipeline variables.

##### Input Parameters:

Parameter	Description
<i>date</i>	String Date input in format "YYMMDD" or "YYYYMMDD".

##### Output Parameters:

Parameter	Description
<i>newYear</i>	String Four digit year.
<i>newMonth</i>	String Two digit month.
<i>newDay</i>	String Two digit day.

Parameter	Description
<i>newHour</i>	String Current time in hour.
<i>newMinute</i>	String Current time in minute.
<i>newSecond</i>	String Current time in second.

**Usage Notes:**

Output variable *status* is not used.

### 3.3 Tutorial.EDItoXML:processEDI850\_4010

Illustrates how to convert an X12 4010 850 document to an XML string.

**Input Parameters:**

Parameter	Description
<i>edidata</i>	String Sample X12 4010 850 document.

**Output Parameters:**

Parameter	Description
<i>xmldata</i>	String Output XML string.

**Usage Notes:**

The `sampleServices:X12toValues` service is used to generate an `IData` object from the EDI document. The `pub.xml:documentToXMLString` service is used to generate an XML String.

## 4 Tutorial.EDItoXML:segmentMapping

The services in this folder are used to map the ANSI X12 segments to the XML targets. The Tutorial.EDItoXML:EDItoXMLPOMap service uses these services.

## 5 Tutorial.Flatfile

The services in this folder illustrate how to convert the two main types of flat files (character-delimited and positional) to an IData object using a template and a flat file as input.

### 5.1 Tutorial.FlatFile:parseCsvInputS

Illustrates how to convert a character-delimited flat file to an IData object using a template and the flat file as input.

**Input Parameters:**

None.

**Output Parameters:**

None.

**See Also:**

Tutorial.Records:rec\_XCEL

### 5.2 Tutorial.FlatFile:parseFlatFile

Illustrates how to convert a positional flat file to an IData object using a template and the flat file as input. The flat file is created using Microsoft Excel.

**Input Parameters:**

None.

**Output Parameters:**

None.

**See Also:**

Tutorial.Records:rec\_FF

### 5.3 Tutorial.FlatFile:parseFlatFile2

Illustrates how to convert a positional flat file to an IData object using a template and a flat file as input. The flat file is created using Microsoft Excel.

**Input Parameters:**

None.

**Output Parameters:**

None.

**See Also:**

Tutorial.Records:rec\_FF2

## 6 Tutorial.Records

The IS document types in this folder are used for mapping purposes in the following three folders: Tutorial.EDItToXML, Tutorial.XMLtoEDI, and Tutorial.Flatfile.

### 6.1 Tutorial.Records:POSource

Reflects the source XML document that the services in the Tutorial.XMLtoEDI folder use.

### 6.2 Tutorial.Records:POTarget

Reflects the target XML document that the services in the Tutorial.EDItToXML folder use.

### 6.3 Tutorial.Records:rec\_FF

The IS document type that the flat file sample in Tutorial.FlatFile:parseFlatFile uses. It also represents the output of the service.

### 6.4 Tutorial.Records:rec\_FF2

The IS document type that the flat file sample in Tutorial.FlatFile:parseFlatFile2 uses. It also represents the output of the service.

### 6.5 Tutorial.Records:rec\_XCEL

The IS document type that the flat file sample in Tutorial.FlatFile:parseCsvInput uses. It also represents the output of the service.

## 7 Tutorial.XMLtoEDI

The services in this folder illustrate the mapping and converting of an XML document into an EDI document.

### 7.1 Tutorial.XMLtoEDI:XMLPOtoEDIMap

Maps an XML PO to an ANSI X12 4010 850 EDI document.

This service invokes many mapping services that are available in the Tutorial.XMLtoEDI.segmentMapping folder.

#### Input Parameters:

Parameter	Description
<i>POSource</i>	Document PO that uses the format defined in the Tutorial.Records:POSource IS document type. This represents the XML PO document structure.

#### Output Parameters:

Parameter	Description				
<i>EDIT850_4010</i>	Document X12 4010 850 EDI document that uses the format defined in the EDIrecords:X12_4010_850 IS document type.				
	<table> <tr> <th>Variable in EDIT850_4010</th><th>Description</th></tr> <tr> <td><i>ST</i></td><td>Document EDI segment from which the IDATA object starts.</td></tr> </table>	Variable in EDIT850_4010	Description	<i>ST</i>	Document EDI segment from which the IDATA object starts.
Variable in EDIT850_4010	Description				
<i>ST</i>	Document EDI segment from which the IDATA object starts.				

### 7.2 Tutorial.XMLtoEDI:XMLdateConversionToDateTime

Converts the date and time format based on the input specifications.

#### Input Parameters:

Parameter	Description
<i>XMLDate</i>	String Input date in the YYYYMMDD hh:mm:ss format.
<i>yearDigit</i>	String (optional) Number of digits for year in the output. Default is set to four.

#### Output Parameters:

Parameter	Description
<i>date</i>	String Date.
<i>time</i>	String Time.

### 7.3 Tutorial.XMLtoEDI:appendSegment

Appends an IData object that represents a segment to an array of IData objects.

Input Parameters:

Parameter	Description
<i>record</i>	Document Segment to add.
<i>recordList</i>	Document List Array of IData objects to which the segment will be appended.

Output Parameters:

Parameter	Description
<i>recordList</i>	Document List Array of IData objects to which the segment is appended.

### 7.4 Tutorial.XMLtoEDI:conditionalTruncate

Truncates an input String to a specified length.

Input Parameters:

Parameter	Description
<i>inString</i>	String Input string.
<i>truncateLength</i>	String Length to truncate.

Output Parameters:

Parameter	Description
<i>outString</i>	String Resulting truncated string.

### 7.5 Tutorial.XMLtoEDI:processXMLSource

Converts an input XML document into an IData object and maps the IData object into a string that represents an EDI document.

Input Parameters:

Parameter	Description
<i>xmlData</i>	String (optional) Input XML document.

Output Parameters:

Parameter	Description
<i>EDIDocument</i>	String (optional) ANSI X12 850 document.



## 7.6 Tutorial.XMLtoEDI:replaceControlChar

Replaces the control characters, such as carriage return, line feed, tab, and so on, with the specified string.

### Input Parameters:

Parameter	Description
<i>inString</i>	String Input string.
<i>replaceString</i>	String Character string used to replace the control characters.

### Output Parameters:

Parameter	Description
<i>outString</i>	String Resulting string.

## 8 Tutorial.XMLtoEDI.segmentMapping

This folder contains segment mappings that map from an XML source to an EDI segment. You can use these as examples for your own mapping needs.

### 8.1 Tutorial.XMLtoEDI.segmentMapping:XMLToBEG

Maps an XML document equivalent to a BEG segment to an IData object that represents a BEG segment. This service uses the Tutorial.XMLtoEDI:XMLdateConversionToDateTime service.

### 8.2 Tutorial.XMLtoEDI.segmentMapping:XMLToDTM

Maps an XML document equivalent to a DTM segment to an IData object that represents a DTM segment. This service uses the Tutorial.XMLtoEDI:XMLdateConversionToDateTime and Tutorial.XMLtoEDI:appendSegment services.

### 8.3 Tutorial.XMLtoEDI.segmentMapping:XMLToFOB

Maps an XML document equivalent to a FOB segment to an IData object that represents a FOB segment. This service uses the Tutorial.XMLtoEDI:appendSegment service to append a FOB IData object to an array of FOB IData objects.

### 8.4 Tutorial.XMLtoEDI.segmentMapping:XMLToGS

Maps an XML document equivalent to a GS segment to an IData object that represents a GS segment.

### 8.5 Tutorial.XMLtoEDI.segmentMapping:XMLToISA

Maps an XML document equivalent to an ISA segment to an IData object that represents an ISA segment.

### 8.6 Tutorial.XMLtoEDI.segmentMapping:XMLToN1

Maps an XML document equivalent to a N1 segment to an IData object that represents a N1 segment.

### 8.7 Tutorial.XMLtoEDI.segmentMapping:XMLToPO1\_CTT

Maps an input XML source line item to the ANSI X12 850 PO1 and CTT segments.

**Input Parameters:**

Parameter	Description
<i>LineItem</i>	Document List XML source line items.

**Output Parameters:**

Parameter	Description
<i>POTotal</i>	String (optional) Maps to AMT02 in the AMT segment under the CTT segment.
<i>PO1</i>	Document List PO1 loop for the ANSI X12 850 transaction set.
<i>CTT</i>	Document List CTT segment for the ANSI X12 850 transaction set.

### 8.8 Tutorial.XMLtoEDI.segmentMapping:XMLToREF

Creates a REF segment loop for the ANSI X12 850 transaction sets.

Input Parameters:

Parameter	Description
<i>CustomerPONumber</i>	String Purchase order number.
<i>AccountingCode</i>	Document List Accounting code.

Output Parameters:

Parameter	Description
<i>REF</i>	Document List REF segment loop for the ANSI X12 850 transaction set.

### 8.9 Tutorial.XMLtoEDI.segmentMapping:XMLToTD5

Creates a TDS5 segment loop for the ANSI X12 850 transaction sets.

Input Parameters:

Parameter	Description
<i>ShipMethodCode</i>	String Ship method code.
<i>ShipmentPriorityCode</i>	String Shipment priority code.

Output Parameters:

Parameter	Description
<i>TD5</i>	Document List TD5 segment loop for the ANSI X12 850 transaction set.

### 8.10 Tutorial.XMLtoEDI.segmentMapping:XMLToTXI

Creates a TXI segment loop for the ANSI X12 850 transaction sets.

Input Parameters:

Parameter	Description
<i>TaxExemptCode</i>	<b>String</b> Tax exempt code.
<i>TaxExemptNumber</i>	<b>String</b> Tax exempt number.

**Output Parameters:**

Parameter	Description
<i>TXI</i>	<b>Document List</b> TXI segment loop for the ANSI X12 850 transaction set.

## 9 Tutorial.properties

This folder contains a Java service that obtains the trading partner-specific information. This information is used by the mapping services in the Tutorial.XMLtoEDI.segmentMapping folder to create the Interchange Header/Trailer.

### 9.1 Tutorial.properties:getProperties

Retrieves properties that are defined in the partner's configuration file.

#### Input Parameters:

Parameter	Description
<i>B2B_config_file</i>	String Configuration file name.
<i>properties</i>	String List Property names and values to retrieve.

#### Output Parameters:

Parameter	Description
<i>values</i>	String List Retrieved property names and values.

## 10 Sample Flow Services

The following sample flow services illustrate how you can use different built-in services and features of Module for EDI to process your documents:

### 10.1 sampleServices:Iterator810

Demonstrates the use of the *iterator* variable in the `wm.b2b.edi:convertToValues` service to parse large input data. Every time the `wm.b2b.edi:convertToValues` service is invoked, the *iterator* variable is set to `true`.

The format of the `EDIFFSchema.X12.V4010:Sample810_Large` flat file schema indicates how the service processes the sample 810 document. This flat file schema defines three top-level segment structures: ST, IT1, and TDS.

The first time the `Iterator810` service invokes the `wm.b2b.edi:convertToValues` service, the ST segment structure is processed. It reads the input data from the sample 810 document until it encounters the next top-level segment structure, which is, the first IT1 segment structure. Next, the `Iterator810` service invokes the `wm.b2b.edi:convertToValues` service for each IT1 segment structure until it encounters the TDS segment structure. After processing the TDS segment structure, the `Iterator810` service exits the REPEAT loop if there is no more input data. The `Iterator810` service determines that there is no more input data when *ediObject*, an output parameter from the `wm.b2b.edi:convertToValues` service, is null.

Because the `wm.b2b.edi:convertToValues` service always returns its output in *EDIVvalues*, each pass of the service invocation should save the output or map it somewhere else. In this example, the main iteration maps each segment under ISA. In the iteration steps under the ST segment, an empty sequence operation is used. This can be replaced by the services that process one item in the invoice. The services have access to the current items and all the information in ISA, GS, ST, and other segments under ST. This service does not validate or check the structure of any segments.

#### Input Parameters:

Parameter	Description
<i>edidata</i>	<b>String</b> EDI 810 document of ANSI X12 version 4010. If it is not specified, then the default sample document is used.

#### Output Parameters:

Parameter	Description
<i>Values</i>	<b>Document List</b> 810 document structure that is defined in the <code>EDIFFSchema.X12.V4010:Sample810_Large</code> flat file schema.
<i>EDIVvalues</i>	<b>Document Single</b> invoice item. This is displayed only after the sub-level iterator call.

## 10.2 sampleServices:UNEDIFACTtoValues

Illustrates an example flow service for an inbound UN/EDIFACT document.

### Input Parameters:

Parameter	Description								
<i>edidata</i>	<b>String</b> UN/EDIFACT standard document.								
<i>complianceCheck</i>	<b>String</b> (optional) Whether the compliance check is required. Specify either <code>true</code> or <code>false</code> . <table> <tr> <th>Value of complianceCheck</th><th>Meaning</th></tr> <tr> <td><code>True</code></td><td>Compliance check is required.</td></tr> <tr> <td><code>False</code></td><td>Compliance check is <i>not</i> required.</td></tr> </table>	Value of complianceCheck	Meaning	<code>True</code>	Compliance check is required.	<code>False</code>	Compliance check is <i>not</i> required.		
Value of complianceCheck	Meaning								
<code>True</code>	Compliance check is required.								
<code>False</code>	Compliance check is <i>not</i> required.								
<i>Validate</i>	<b>String</b> (optional) <table> <tr> <th>Value of validate</th><th>Meaning</th></tr> <tr> <td><code>True</code></td><td>Validation check is required.</td></tr> <tr> <td><code>False</code></td><td>Validation check is <i>not</i> required.</td></tr> </table>	Value of validate	Meaning	<code>True</code>	Validation check is required.	<code>False</code>	Validation check is <i>not</i> required.		
Value of validate	Meaning								
<code>True</code>	Validation check is required.								
<code>False</code>	Validation check is <i>not</i> required.								
<i>returnErrors</i>	<b>String</b> (optional) How you want this service to return error messages when <i>validate</i> is set to <code>true</code> . Specify one of the following: <table> <tr> <th>Value of validate</th><th>Meaning</th></tr> <tr> <td><code>asArray</code></td><td>Return validation errors with the <i>edidata</i> in an array called <i>errors</i>. This is the default.</td></tr> <tr> <td><code>inResults</code></td><td>Return validation errors in the <i>EDIVvalues</i> object.</td></tr> <tr> <td><code>Both</code></td><td>Return validation errors in both <i>errors</i> and <i>EDIVvalues</i>.</td></tr> </table>	Value of validate	Meaning	<code>asArray</code>	Return validation errors with the <i>edidata</i> in an array called <i>errors</i> . This is the default.	<code>inResults</code>	Return validation errors in the <i>EDIVvalues</i> object.	<code>Both</code>	Return validation errors in both <i>errors</i> and <i>EDIVvalues</i> .
Value of validate	Meaning								
<code>asArray</code>	Return validation errors with the <i>edidata</i> in an array called <i>errors</i> . This is the default.								
<code>inResults</code>	Return validation errors in the <i>EDIVvalues</i> object.								
<code>Both</code>	Return validation errors in both <i>errors</i> and <i>EDIVvalues</i> .								

### Output Parameters:

Parameter	Description
<i>Values</i>	<b>Document</b> Output of the UN/EDIFACT document in IData object format.

## 10.3 sampleServices:X12toValues

Illustrates an example flow service for an inbound ANSI X12 document.

### Input Parameters:

Parameter	Description
<i>edidata</i>	<b>String</b> EDI document.

Parameter	Description								
<i>docProcessingSvc</i>	<b>String</b> Service to process an ANSI X12 transaction.								
<i>complianceCheck</i>	<b>String</b> (optional) Whether the compliance check is required. Specify either <code>true</code> or <code>false</code> . <table> <tr> <th>Value of complianceCheck</th><th>Meaning</th></tr> <tr> <td><code>true</code></td><td>Compliance check is required.</td></tr> <tr> <td><code>false</code></td><td>Compliance check is <i>not</i> required.</td></tr> </table>	Value of complianceCheck	Meaning	<code>true</code>	Compliance check is required.	<code>false</code>	Compliance check is <i>not</i> required.		
Value of complianceCheck	Meaning								
<code>true</code>	Compliance check is required.								
<code>false</code>	Compliance check is <i>not</i> required.								
<i>validate</i>	<b>String</b> (optional) <table> <tr> <th>Value of validate</th><th>Meaning</th></tr> <tr> <td><code>true</code></td><td>Validation check is required.</td></tr> <tr> <td><code>false</code></td><td>Validation check is <i>not</i> required.</td></tr> </table>	Value of validate	Meaning	<code>true</code>	Validation check is required.	<code>false</code>	Validation check is <i>not</i> required.		
Value of validate	Meaning								
<code>true</code>	Validation check is required.								
<code>false</code>	Validation check is <i>not</i> required.								
<i>returnErrors</i>	<b>String</b> (optional) How you want this service to return error messages when <i>validate</i> is set to <code>true</code> . Specify one of the following: <table> <tr> <th>Value of validate</th><th>Meaning</th></tr> <tr> <td><code>asArray</code></td><td>Return validation errors with the <i>edidata</i> in an array called <i>errors</i>. This is the default.</td></tr> <tr> <td><code>inResults</code></td><td>Return validation errors in the <i>EDIVvalues</i> object.</td></tr> <tr> <td><code>both</code></td><td>Return validation errors in both <i>errors</i> and <i>EDIVvalues</i>.</td></tr> </table>	Value of validate	Meaning	<code>asArray</code>	Return validation errors with the <i>edidata</i> in an array called <i>errors</i> . This is the default.	<code>inResults</code>	Return validation errors in the <i>EDIVvalues</i> object.	<code>both</code>	Return validation errors in both <i>errors</i> and <i>EDIVvalues</i> .
Value of validate	Meaning								
<code>asArray</code>	Return validation errors with the <i>edidata</i> in an array called <i>errors</i> . This is the default.								
<code>inResults</code>	Return validation errors in the <i>EDIVvalues</i> object.								
<code>both</code>	Return validation errors in both <i>errors</i> and <i>EDIVvalues</i> .								

**Output Parameters:**

Parameter	Description
<i>Values</i>	Document Output of the ANSI X12 document in IData object format.



## 11 Sample Business Process

The WmEDISamples package provides a sample that illustrates the use of the EDI documents in a business process. Before you read the overview of the sample, you should set it up and run it.

To set up a sample business process:

---

**Note:** If you also want to run the sample provided for webMethods EDIINT Module (which is, also included in the WmEDISamples package), then set up the EDIINT Module sample *before* setting up this sample business process.

---

1. Download the WmEDISamples package under *Tips, Tricks, and Code* in the Software AG TECH-community website at <http://techcommunity.softwareag.com>.

---

**IMPORTANT:** You should delete the WmEDISamples package before you move into production.

---

2. Open the Server Administrator.
3. In the navigation panel: **Packages > Management**, click **WmEDISamples** home icon.  
The Server Administrator displays the webMethods Module for EDI Sample in a new browser window.
4. In the navigation panel of webMethods Module for EDI Sample page, click **Overview**.
5. To continue, follow the instructions in the Overview page.

The following sections describe the services that are used in the execution and setup of the sample:

### 11.1 [wm.b2b.editn.sample.setup:setupSample](#)

Creates the partner profiles and TN document types for the sample, if they do not exist.

**Input Parameters:**

Parameter	Description
<i>SenderCorporateName</i>	<b>String</b> Corporate name of the sender's profile.
<i>SenderDuns</i>	<b>String</b> Sender's D-U-N-S number.
<i>SenderHost</i>	<b>String</b> Machine name of the sender.
<i>SenderPort</i>	<b>String</b> Machine name of the sender.
<i>SenderUserName</i>	<b>String</b> User name used to connect to the sender's machine.
<i>SenderPassword</i>	<b>String</b> Password used to connect to the sender's machine.
<i>ReceiverCorporateName</i>	<b>String</b> Corporate name of the receiver's profile.
<i>ReceiverDuns</i>	<b>String</b> Receiver's D-U-N-S number.
<i>ReceiverHost</i>	<b>String</b> Machine name of the receiver.

Parameter	Description
<i>ReceiverPort</i>	<b>String</b> Port on which the receiver will view the data.
<i>ReceiverUsername</i>	<b>String</b> User name used to connect to the receiver's machine.
<i>ReceiverPassword</i>	<b>String</b> Password used to connect to the receiver's machine.

**Output Parameters:**

None.

**Usage Notes:**

Only the sample web pages should invoke this service. If you also want to run the sample provided for webMethods EDIINT Module in the WmEDISamples package, set up the EDIINT Module sample *before* setting up the WmEDIforTN sample.

## 11.2 [wm.b2b.editn.sample.setup:setupEDIBizDocTypes](#)

Installs the TN document types that are required for the EDI documents used in the sample.

**Input Parameters:**

None.

**Output Parameters:**

None.

**Usage Notes:**

Only the sample web pages should invoke this service.

## 11.3 [wm.b2b.editn.sample.setup:setupXMLBizDocTypes](#)

Installs the TN document types that are needed for the XML documents used in the sample.

**Input Parameters:**

None.

**Output Parameters:**

None.

**Usage Notes:**

Only the sample web pages should invoke this service.

## 11.4 [wm.b2b.editn.sample.setup:setupProfile](#)

Sets up the profile for partners in the sample.

### Input Parameters:

Parameter	Description
<i>CorporationName</i>	<b>String</b> Corporate name for the profile.
<i>DUNS</i>	<b>String</b> External ID for the profile.
<i>Host</i>	<b>String</b> Machine name.
<i>port</i>	<b>String</b> Port for this profile.
<i>url</i>	<b>String</b> URL to connect to.
<i>User</i>	<b>String</b> User name for this profile.
<i>Password</i>	<b>String</b> Password for the profile.

### Output Parameters:

None.

### Usage Notes:

Only the sample web pages should invoke this service.

## 11.5 [wm.b2b.editn.sample.receiver:Process850](#)

Receives EDI X12 4010 850 and generates an 855 in response.

### Input Parameters:

None.

### Output Parameters:

Parameter	Description
<i>edidata</i>	<b>String</b> EDI 855 document.

### Usage Notes:

The 'EDI Seller 850-855' business process invokes this service.

### 11.6 [wm.b2b.editn.sample.receiver:Send997](#)

Transmits a 997 document to the trading partner of the receiver.

#### Input Parameters:

Parameter	Description
997	String Text of an EDI 977 document.

#### Output Parameters:

None.

#### Usage Notes:

The 'Receiver EDI Envelope' business process invokes this service.

### 11.7 [wm.b2b.editn.sample.receiver:Generate997](#)

Creates a 997 document in response to an X12 Envelope.

#### Input Parameters:

Parameter	Description
<i>X12Env</i>	Document X12 envelope.
<i>Edidata</i>	String X12 Envelope used to generate the 997 functional acknowledgment.

#### Output Parameters:

Parameter	Description
997	String 977 document.

#### Usage Notes:

The 'Receiver EDI Envelope' business process invokes this service.

### 11.8 [wm.b2b.editn.sample.receiver:Send855](#)

Sends an EDI 855 document to the trading partner of the sender.

#### Input Parameters:

Parameter	Description
-----------	-------------

Parameter	Description
<i>edidata</i>	<b>String</b> Text of an EDI 855 document.

**Output Parameters:**

None.

**Usage Notes:**

The 'EDI Seller 850-855' business process invokes this service.

### 11.9 [wm.b2b.editn.sample.receiver:ProcessEnvelope](#)

Represents where an envelope processing would occur in a business process.

**Input Parameters:**

None.

**Output Parameters:**

None.

**Usage Notes:**

The 'Receiver EDI Envelope' business process invokes this service.

### 11.10 [wm.b2b.editn.sample.util:getLastDocuments](#)

Retrieves the BizDocEnvelopes for events that occurred in a business process.

**Input Parameters:**

Parameter	Description
<i>parameters</i>	<b>Document</b> Provided by the process run time facility of the Integration Server. The process run time facility manages the execution of business processes.
<i>documents</i>	<b>Document</b> Provided by the process run time facility of the Integration Server.
<i>getContent</i>	<b>String</b> Whether the document content that is saved in the BizDocEnvelope should be retrieved.
<i>lastEvents</i>	<b>String List</b> Names of the most recent event(s) that occurred in the business process.

**Output Parameters:**

Parameter	Description
<i>recordCount</i>	String Number of events returned.
<i>events</i>	Document List (optional) BizDocEnvelopes for all the document events.

### 11.11 [wm.b2b.editn.sample.util:enable](#)

This service places a variable in the pipeline that alters the way in which the conversation IDs are constructed for the EDI documents.

To run both sides of a conversation on a single instance of the Integration Server, the construction of the conversation IDs is enforced as the `ReceiverID-SenderID-InstanceID`.

#### Input Parameters:

None.

#### Output Parameters:

None.

#### Usage Notes:

This service should be used only for demonstration purposes. The conversation ID is only constructed this way if the variable is present in the pipeline when the service `wm.tn:receive` is invoked.

### 11.12 [wm.b2b.editn.sample.util:return997](#)

Determines whether a 997 should be sent in response to a document.

#### Input Parameters:

Parameter	Description
<i>parameters</i>	<b>Document</b> Provided by the process run time facility of the Integration Server. The process run time facility manages the execution of business processes.
<i>documents</i>	<b>Document</b> Provided by the process run time facility of the Integration Server.

#### Output Parameters:

Parameter	Description
<i>X12Env</i>	<b>Document</b> BizDocEnvelope of the previous business process event that occurred.
<i>send997</i>	<b>String</b> Whether a 997 should be sent. If a 997 should be sent, then specify <code>true</code> , otherwise specify <code>false</code> .

### 11.13 [wm.b2b.editn.sample.util:getBizDocFromEvent](#)

Retrieves the BizDocEnvelopes for the events that occurred in a business process.

#### Input Parameters:

Parameter	Description
<i>parameters</i>	<b>Document</b> Provided by the process run time facility of the Integration Server. The process run time facility manages the execution of business processes.
<i>documents</i>	<b>Document</b> Provided by the process run time facility of the Integration Server.
<i>getContent</i>	<b>String</b> Whether the document content that is saved in the BizDocEnvelope should be retrieved.
<i>lastEvents</i>	<b>String List</b> Names of the most recent event(s) that occurred in the business process.

#### Output Parameters:

Parameter	Description
<i>bizEnv</i>	<b>Document</b> The BizDocEnvelope.

### 11.14 [wm.b2b.editn.sample.util:checkSetup](#)

Checks the system to determine whether the sample is configured properly or otherwise.

#### Input Parameters:

None.

#### Output Parameters:

Parameter	Description
<i>setupProperly</i>	<b>String</b> Whether all the components of the sample are configured properly. If configured properly, then specify <code>true</code> , otherwise specify <code>false</code> .
<i>profilesOkay</i>	<b>String</b> Whether the trading partner profiles for the sample are present. If the profiles are present, then specify <code>true</code> , otherwise specify <code>false</code> .
<i>docsOkay</i>	<b>String</b> Whether the TN document types for the sample are present. If the TN document types are present, then specify <code>true</code> , otherwise specify <code>false</code> .
<i>scriptsOkay</i>	<b>String</b> Whether the process models for the sample are present. If the process models are present, then specify <code>true</code> , otherwise specify <code>false</code> .

### 11.15 [wm.b2b.editn.sample.util:receive](#)

Alternate receive method that first calls `wm.b2b.editn.sample.util:enable` and then the `wm.tn:receive` service.

This is to enforce the conversation IDs in such a way that both sides of the conversation are executed on a single instance of the Integration Server.

#### Input Parameters:

None.

#### Output Parameters:

None.

#### Usage Notes:

Only the sample web pages should invoke this service.

### 11.16 [wm.b2b.editn.sample.util:deliver](#)

Deliver BizDocEnvelopes from one trading partner to another. The default method is to use the Secondary HTTP delivery method that is specified in the receiver's trading partner profile.

#### Input Parameters:

Parameter	Description
<i>bizdoc</i>	Document BizDocEnvelope to deliver.

#### Output Parameters:

None.

#### Usage Notes:

Only the sample web pages should invoke this service.

### 11.17 [wm.b2b.editn.sample.sender:Process997](#)

This is a placeholder service that represents the processing of a 997 performed by the sender.

#### Input Parameters:

None.

#### Output Parameters:



None.

**Usage Notes:**

The 'Send EDI Envelope' business process invokes this service.

**11.18 [wm.b2b.editn.sample.sender:Log850](#)**

This is a placeholder service that represents the logging of an outbound 850 message by the sender.

**Input Parameters:**

None.

**Output Parameters:**

None.

**Usage Notes:**

The 'EDI Buyer 850-855' business process invokes this service.

**11.19 [wm.b2b.editn.sample.sender:SendEnvelope](#)**

Sends an X12 envelope containing an EDI 850 document to a trading partner.

**Input Parameters:**

None.

**Output Parameters:**

None.

**Usage Notes:**

The 'Send EDI Envelope' business process invokes this service.

**11.20 [wm.b2b.editn.sample.sender:GenerateEnvelope](#)**

Generates an X12 envelope containing an EDI 850 document.

**Input Parameters:**

None.

**Output Parameters:**

None.

**Usage Notes:**

The 'EDI Buyer 850-855' business process invokes this service.

### 11.21 [wm.b2b.editn.sample.sender:Process855](#)

This is placeholder service that represents the business logic that is executed when an 855 document is received.

**Input Parameters:**

None.

**Output Parameters:**

None.

**Usage Notes:**

The 'EDI Buyer 850-855' business process.

### 11.22 [wm.b2b.editn.sample.sender:start](#)

Starts the sample. Creates an XML Purchase Order document that is submitted to Trading Networks.

**Input Parameters:**

Parameter	Description
<i>SenderID</i>	<b>String</b> D-U-N-S number of the trading partner that sends the document.
<i>ReceiverID</i>	<b>String</b> D-U-N-S number of the trading partner to whom the document is sent.
<i>DocumentID</i>	<b>String</b> Unique identifier for the purchase order.
<i>ControlNumber</i>	<b>String</b> Unique identifier for the group segment that contains the purchase order.

**Output Parameters:**

Parameter	Description
<i>SenderTNID</i>	<b>String</b> Trading Networks internal ID for the sender.
<i>ReceiverTNID</i>	<b>String</b> Trading Networks internal ID for the receiver.

**Usage Notes:**

Only the sample web pages should invoke this service.

## 12 Wm.EDIINT.sample

You can run the following sample send service within Software AG Designer to exchange sample EDIINT messages:

### 12.1 Wm.EDIINT.sample:send

Reads the input file as an `InputStream` and invokes the `wm.EDIINT.send` service.

#### Input Parameters:

Parameter	Description																		
<i>dataFileName</i>	<b>String</b> Name and location of the file to send.																		
<i>dataContentType</i>	<b>String</b> Content type of the file to send. For example, application/edi-x12.																		
<i>type</i>	<b>String</b> S/MIME type that you want to use for the outbound EDIINT message. Specify one of the following: <table><tr><th>Specify the Value</th><th>Type</th><th>Description</th></tr><tr><td>plain</td><td></td><td>When you want the <code>wm.EDIINT.sample:send</code> service to neither sign nor encrypt the outbound EDIINT message.</td></tr><tr><td>signed</td><td></td><td>When you want the <code>wm.EDIINT.sample:send</code> service to sign the outbound EDIINT message.</td></tr><tr><td>encrypted</td><td></td><td>When you want the <code>wm.EDIINT.sample:send</code> service to encrypt the outbound EDIINT message.</td></tr><tr><td>signedAndEncrypted</td><td></td><td>When you want the <code>wm.EDIINT.sample:send</code> service to sign and encrypt the outbound EDIINT message.</td></tr><tr><td>getFromProfile</td><td></td><td>When you want the <code>wm.EDIINT.sample:send</code> service to determine whether to sign and/or encrypt the outbound EDIINT message using the value of the S/MIME Type extended field from the receiver's profile.</td></tr></table>	Specify the Value	Type	Description	plain		When you want the <code>wm.EDIINT.sample:send</code> service to neither sign nor encrypt the outbound EDIINT message.	signed		When you want the <code>wm.EDIINT.sample:send</code> service to sign the outbound EDIINT message.	encrypted		When you want the <code>wm.EDIINT.sample:send</code> service to encrypt the outbound EDIINT message.	signedAndEncrypted		When you want the <code>wm.EDIINT.sample:send</code> service to sign and encrypt the outbound EDIINT message.	getFromProfile		When you want the <code>wm.EDIINT.sample:send</code> service to determine whether to sign and/or encrypt the outbound EDIINT message using the value of the S/MIME Type extended field from the receiver's profile.
Specify the Value	Type	Description																	
plain		When you want the <code>wm.EDIINT.sample:send</code> service to neither sign nor encrypt the outbound EDIINT message.																	
signed		When you want the <code>wm.EDIINT.sample:send</code> service to sign the outbound EDIINT message.																	
encrypted		When you want the <code>wm.EDIINT.sample:send</code> service to encrypt the outbound EDIINT message.																	
signedAndEncrypted		When you want the <code>wm.EDIINT.sample:send</code> service to sign and encrypt the outbound EDIINT message.																	
getFromProfile		When you want the <code>wm.EDIINT.sample:send</code> service to determine whether to sign and/or encrypt the outbound EDIINT message using the value of the S/MIME Type extended field from the receiver's profile.																	
<i>deliveryMethod</i>	<b>String</b> Delivery method you want to use to send the EDIINT document. The <code>wm.EDIINT:send</code> service obtains the Trading Networks profile for the receiver that is specified by the <code>receiverID</code> input variable and delivers the EDIINT document to the receiver's system using the information specified in the delivery method. Specify a delivery method that is defined in the receiver's Trading Networks profile. <ul style="list-style-type: none"><li>For EDIINT AS1, specify one of the following:<ul style="list-style-type: none"><li>PrimarySMTP (corresponds to Trading Networks Primary E-mail delivery method)</li><li>SecondarySMTP (corresponds to Trading Networks Second-</li></ul></li></ul>																		

Parameter	Description								
	<p>ary E-mail delivery method)</p> <ul style="list-style-type: none"> <li>For EDIINT AS2, specify one of the following: <ul style="list-style-type: none"> <li>PrimaryHTTP</li> <li>SecondaryHTTP</li> <li>PrimaryHTTPS</li> <li>SecondaryHTTPS</li> </ul> </li> <li>For EDIINT AS3, specify the following: <ul style="list-style-type: none"> <li>AS3</li> </ul> </li> </ul>								
<i>returnMDN</i>	<p><b>String</b> Whether you want the receiver to return an MDN. Specify one of the following:</p> <table> <tr> <th>Value of requestMDN</th><th>Meaning</th></tr> <tr> <td>none</td><td>Do <i>not</i> request a return MDN.</td></tr> <tr> <td>synchronousMDN</td><td>Request a return synchronous MDN.</td></tr> <tr> <td>asynchronousMDN</td><td>Request a return asynchronous MDN.</td></tr> </table> <p><b>Note:</b> If you specify PrimarySMTP, SecondarySMTP, PrimaryFTPS, or SecondaryFTPS as <i>deliveryMethod</i>, then you will only receive an asynchronous MDN.</p>	Value of requestMDN	Meaning	none	Do <i>not</i> request a return MDN.	synchronousMDN	Request a return synchronous MDN.	asynchronousMDN	Request a return asynchronous MDN.
Value of requestMDN	Meaning								
none	Do <i>not</i> request a return MDN.								
synchronousMDN	Request a return synchronous MDN.								
asynchronousMDN	Request a return asynchronous MDN.								
<i>requestSignedReceipt</i>	<p><b>String</b> Whether you want the MDN to be signed.</p> <p><b>Note:</b> <i>requestSignedReceipt</i> is ignored when <i>requestMDN</i> is false.</p> <table> <tr> <th>Value of re-questSignedReceipt</th><th>Meaning</th></tr> <tr> <td>true</td><td>Request a signed MDN.</td></tr> <tr> <td>false</td><td>Request a plain (not signed) MDN.</td></tr> </table>	Value of re-questSignedReceipt	Meaning	true	Request a signed MDN.	false	Request a plain (not signed) MDN.		
Value of re-questSignedReceipt	Meaning								
true	Request a signed MDN.								
false	Request a plain (not signed) MDN.								
<i>senderID</i>	<p><b>Document</b> Identification of the sender of the EDIINT message.</p> <table> <tr> <th>Variable in senderID</th><th>Description</th></tr> <tr> <td>id</td><td>Sender's external ID, that is, the identification for the sender in the message.</td></tr> <tr> <td>idTypeDesc</td><td>(Optional) External ID type for the sender ID you specified in <i>id</i>. This is an external ID type as defined in Trading Networks.</td></tr> </table> <p><b>Note:</b> By default, the service uses the appropriate external ID type based on the value you specify for the <i>deliveryMethod</i> parameter. For example, if you specify AS3 for the <i>deliveryMethod</i> parameter, then the service uses the EDIINT AS3 external ID type. Specify a value for <i>idTypeDesc</i> only if you want to override this default.</p>	Variable in senderID	Description	id	Sender's external ID, that is, the identification for the sender in the message.	idTypeDesc	(Optional) External ID type for the sender ID you specified in <i>id</i> . This is an external ID type as defined in Trading Networks.		
Variable in senderID	Description								
id	Sender's external ID, that is, the identification for the sender in the message.								
idTypeDesc	(Optional) External ID type for the sender ID you specified in <i>id</i> . This is an external ID type as defined in Trading Networks.								

Parameter	Description	
<i>receiverID</i>	Document Identification of the receiver of the EDIINT message.	
	Variable in senderID	Description
	<i>id</i>	The receiver's external ID, that is, the identification for the receiver in the message.
	<i>idTypeDesc</i>	(Optional) External ID type for the receiver ID you specified in <i>id</i> . This is an external ID type as defined in Trading Networks.
		<hr/> <b>Note:</b> By default, the service uses the appropriate external ID type based on the value you specify for the <i>deliveryMethod</i> parameter. For example, if you specify AS3 for the <i>deliveryMethod</i> parameter, then the service uses the EDIINT AS3 external ID type. Specify a value for <i>idTypeDesc</i> only if you want to override this default. <hr/>
<i>conversationID</i>	<p><b>String</b> (optional) Conversation ID for the outbound EDIINT message. The <i>conversationID</i> parameter is an identifier that links all the documents that are part of the same business process (also called a conversation). That is, all the documents in the same business process should have the same <i>conversationID</i>. As described in <i>webMethods Trading Networks User's Guide</i>, Trading Networks can extract <i>conversationIDs</i> from EDI documents and use them to pass the documents to webMethods Process Run Time (PRT) after Trading Networks performs the actions identified by a processing rule.</p> <p>In this field, you should specify the same conversation ID as that of the payload that you send. The EDIINT Module automatically assigns the same conversation ID that is assigned here to a return MDN.</p> <hr/> <p><b>Note:</b> Leave this field empty unless you own a license for webMethods Process Run Time (PRT). <i>requestSignedReceipt</i> is ignored when <i>requestMDN</i> is <i>false</i>.</p> <hr/>	

#### Output Parameters:

None.

## 13 Wm.EDIforTN EANCOM

The service in this folder is used for processing EDI documents through webMethods Trading Networks.

### 13.1 `wm.b2b.editn.eancom:generateAndProcessAUTACK`

This service generates and processes AUTACK messages for the EANCOM sample messages.