webMethods

webMethods EDI Module

Built-In Services Reference

VERSION 6.5.2

webMethods, Inc.
South Tower
3877 Fairfax Ridge Road
Fairfax, VA 22030
USA
703.460.2500
http://www.webmethods.com

Document ID: EDI-BIS-RF-652-20060612

Contents

About This Guide

The *webMethods EDI Module Built-In Services Reference* describes the built-in services provided with the webMethods EDI Module (EDI Module). The services that are documented in this guide are provided in the WmEDI and WmEDIforTN packages that are installed with the webMethods EDI Module.

Services are also installed with a standard installation of the webMethods Integration Server and webMethods Trading Networks (Trading Networks). You will find documentation for the built-in services provided with the webMethods Integration Server in the *webMethods Integration Server Built-In Services Reference Guide* and documentation for services provided with Trading Networks in the *webMethods Trading Networks Built-In Services Reference*.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Italic	Identifies variable information that you must supply or change based on your specific situation or environment. Identifies terms the first time they are defined in text. Also identifies service input and output variables.
Narrow font	Identifies storage locations for services on the webMethods Integration Server using the convention *folder.subfolder:service*.
`Typewriter font`	Identifies characters and values that you must type exactly or messages that the system displays on the console.
UPPERCASE	Identifies keyboard keys. Keys that you must press simultaneously are joined with the "+" symbol.
\	Directory paths use the "\" directory delimiter unless the subject is UNIX-specific.
[ ]	Optional keywords or values are enclosed in [ ]. Do not type the [ ] symbols in your own code.

Additional Information

The webMethods Advantage Web site at http://advantage.webmethods.com provides you with important sources of information about webMethods components:

- Sample services. webMethods provides sample services and documentation in the WmEDIsample package, which is located in the webMethods Knowledge Base. The sample services in this package have been certified, meaning that they have been tested by webMethods.
- Troubleshooting Information. webMethods provides troubleshooting information for many webMethods components in the webMethods Knowledge Base.
- Documentation Feedback. To provide documentation feedback to webMethods, go to the Documentation Feedback Form on the webMethods Bookshelf.
- Additional Documentation. All webMethods documentation is available on the webMethods Bookshelf.

webMethods

CHAPTER 1

WmEDI Package

wm.b2b.edi

The services in this folder (wm.b2b.edi) are the core services that you use when converting between EDI documents and IS documents (IData objects), and when creating flat file schemas.

wm.b2b.edi:convertToString

Converts an IS document (IData object) to a String, based on a flat file schema that you specify.

The difference between this service and the pub.flatFile:convertToString service is that it handles EDI documents. For EDI documents, it will optionally fill in the counters and control numbers if they are empty. For example, if `SE01` is null, it will fill in the segment count. If you want to control the counters or control numbers, modify the IS document (IData object) prior to invoking this service to convert it to a String.

Note: To convert TRADACOMS documents, use the wm.b2b.edi.tradacoms:convertToString service instead of this service.

Input Parameters

values	**Document** The IS document (IData object) object that you want to convert to a String.
EDItemplate	**Document** (optional) This input variable is provided only for backwards compatibility. It should contain a single name/value pair that is a String named *templateName* with the value set to the namespace name of the flat file schema to use for the conversion. This is the output from the wm.b2b.edi.templateMgr:getTemplate service. Do *not* specify an EDI template. You must specify *ffSchema*, *EDItemplate*, or *nsRecord*. webMethods will deprecate *nsRecord* and *EDItemplate* in the later versions, so it is recommended that you use *ffSchema*.
ffSchema	**String** (optional) The fully-qualified namespace name of the flat file schema to use to convert the specified IS document (IData object) (in *values*) to a String. You must specify *ffSchema*, *EDItemplate*, or *nsRecord*. webMethods will deprecate *nsRecord* and *EDItemplate* in the later versions, so it is recommended that you use *ffSchema*.

spacePad

String How you want the resulting String to be justified. Specify one of the following:

Value of *spacePad*	Meaning
`left`	Left justify.
`right`	Right justify.
`none`	No justification. This is the default.

Important! If you are upgrading from webMethods Integration Server version 4.6, to enable left or right justification you must add the following line to the *webMethods6**IntegrationServer*\packages\WmFlatFile\config\ff.cnf file:

```
spacePadJustifies=false
```

Then, reload the WmFlatFile package so that this configuration setting will take effect. For more information about `spacePadJustifies`, see the *Flat File Schema Developer's Guide*.

noEmptyTrailingFields

String Whether to remove empty trailing fields from records. The convertToString service only uses this variable for records that have delimited fields. Specify `true` or `false`.

Value of *noemptyTrailingFields*	Meaning
`true`	The convertToString service removes empty trailing fields from the output. For example, a record with empty trailing fields might look like the following: `AAA*01*02!` (where `!` is the segment terminator). This is the default.
`false`	The convertToString service does *not* remove empty trailing fields. Instead it uses the field separator to denote an empty field. For example, a record with empty trailing field might look like the following: `AAA*01*02*******!` (where * is the field separator and `!` is the segment terminator).

nsRecord

String (optional) The fully-qualified name of the IS document type on which the resulting String will be based. If you specify *nsRecord*, the convertToString service ignores the *ffSchema* variable.

You must specify *ffSchema*, *EDItemplate*, or *nsRecord*. webMethods will deprecate *nsRecord* and *EDItemplate* in the later versions, so it is recommended that you use *ffSchema*.

Segment_terminator

String (optional) The segment terminator character that you want the convertToString service to append to the end of each record in the output String.

Field_separator String (optional) The field separator that you want the convertToString service to insert between each field for each segment in the output String.

Subfield_separator String (optional) The subfield separator that you want the convertToString service to use for composite elements.

FormatInfo Document (optional) Values you want the convertToString service to pass unmodified to all format services it invokes.

releaseCharacter String (optional) The character you want the convertToString service to use as an escape character. If one of the characters that you specify for *Segment_terminator*, *Field_separator*, or *Subfield_separator* appears in field or subfield, the convertToString service will prefix the character with this escape character before writing it to the output String.

outputFileName String (optional) The name of the file to which you want the String output written. If you do not specify *outputFileName* the output is not written to a file.

encoding String The type of encoding used to write data to the output file. The default encoding is UTF-8.

startAt String Allows the convertToString service to start at a specific record in the flat file schema used to create the output string. Specify the path to the element where you want to start composing the output string.

countSegments String Whether to count the number of segments written to the output file.

`true`	The convertToString service counts the number of segments written to the output file and returns that number in the output parameter *segmentCount*. This is the default.
`false`	The convertToString service does *not* count the number of segments written to the output file.

sortInput String (optional) Whether you want the service to sort the input records to match the flat file schema specified in *ffSchema*. When set to `true` (the default), this flag is useful in either of the following cases:

- If the data in *values* is not in the same order as defined by *ffSchema*.
- If EDI transactions contain two segments with the same name at the same level, but with distinctly different structures. For more information, see "Usage Notes" on page 11.

`true`	You want the service to sort the input records. This is the default.

	Important! If you select to sort the input records, note that: ■ The service will run slower. ■ All undefined records will be sorted *after* the defined records. ■ The order of the undefined records appear in the final document is random. ■ If there are multiple records at the same level with the same name, the order they appear in the final document is random.
`false`	You do not want the service to sort the input records.

Output Parameters

string	String The output String that represents the data specified in the input variable, *values*.
errorArray	String List Error messages describing the errors that the convertToString service encountered during conversion. If the convertToString service did not encounter errors, *errorArray* is null.
segmentCount	String The number of records written; only returned when *countSegments* is `true`.

Usage Notes

- You can specify the terminator or separator as a character (e.g., *), as unicode (e.g., `\u001c`), as a hex character (e.g., `0x15`), as an octo character (e.g., `027`), or as a decimal character (e.g., `21`).
- When you use the convertToString service to convert an IS document (IData object) to an EDI ANSI X12 String, this service automatically:
  - Generates a control number for each header and trailer if the control number is null.
  - Calculates and replaces segment counts, group counts, and document counts to ensure that their values are accurate (only if the count is `blank` or `null`).
- You can also use *sortInput* to handle EDI transactions that contain two segments with the same name at the same level, but with distinctly different structures. For example, assume that the flat file schema for UNEDIFACT 97A INVOIC shown below contains two TAX segments:

```
UNH
.
TAX (in header)
LIN
TAX (in details)
.
UNT
```

The first TAX segment is optional. If only the second TAX segment is present in the file, the output of the wm.b2b.edi:convertToValues service will differ, depending on the value of the *sortInput* flag.

- If *sortInput* is set to true, then the convertToString service assumes that the input IData is out of sequence. The convertToString service will sort the input record so that the TAX segment is in the header of the output document.
- If *sortInput* is set to false (the default), then the TAX segment will appear in the output document after the LIN segment.

Example

See the Tutorial.XMLtoEDI:processXMLSource service in the WmEDIsamples package, which is located in the Knowledge Base on the webMethods Advantage Web site at http://advantage.webmethods.com.

wm.b2b.edi:convertToValues

Converts an InputStream or String (e.g., an EDI transaction set document) to an IS document (IData object) based on the input flat file schema.

Note: To convert TRADACOMS documents, use the wm.b2b.edi.tradacoms:convertToValues service instead of this service.

Input Parameters

edidata	String or InputStream The data you want to convert to an IData object.
ediObject	Object (optional) An object that encapsulates and keeps track of the input data segments during processing. It is used only when the *iterator* variable has been set to `true`.
encoding	String (optional) The encoding of the data passed in to *edidata*.
EDItemplate	Document (optional) This input variable is provided only for backwards compatibility. It should contain a single name/value pair that is a String named *templateName* with the value set to the namespace name of the flat file schema to use for the conversion. This is the output from the wm.b2b.edi.templateMgr:getTemplate service. Do *not* specify an EDI template. You must specify *EDIffSchema* or *EDItemplate*. webMethods will deprecate *EDItemplate* in the later versions, so it is recommended that you use *EDIffSchema*.
EDIFFSchema	String (optional) The fully-qualified name of the flat file schema object used to parse the *edidata* object. You must specify *EDIffSchema* or *EDItemplate*. webMethods will deprecate *EDItemplate* in the later versions, so it is recommended that you use *EDIffSchema*.

delimiters

Document (optional) Delimiters to use to parse the input data. If no delimiters are specified, the convertToValues service uses the corresponding delimiter defined for the flat file schema.

Note: If you specify a value for one variable of *delimiters*, you must specify values for *all* the variables. If you specify delimiters, the delimiters in the flat file schema will not be used.

Variables in *delimiters*	Description
record	String The segment terminator used in the input data.
field	String The field separator used in the input data.
subfield	String The subfield separator used in the input data.
release	String The release character used in the input data.
FormatInfo	Document Values you want the convertToString service to pass unmodified to all format services it invokes.

iterator

String (optional) Whether you want to process segments one at a time or process all input data at one time. Specify `true` or `false`.

Value of *iterator*	Meaning
`true`	The convertToValues service starts processing segment structures with a top-level record as defined by the flat file schema. The service returns to the caller when it encounters another top-level record in the input data. The next time the service is invoked, it begins processing the input data where it left off.
`false`	The convertToValues service processes all input data at one time. This is the default.

encoding

String (optional) The encoding of the InputStream passed in. The default encoding is UTF-8.

nullable

String (optional) Whether to create an IS document (IData object) if all fields are null. Specify `true` or false.

Value of *nullable*	Meaning
`true`	Do not create an IS document (IData object) if all the fields are null. This is the default.
`false`	Always create an IS document even though all the fields are null.

skipWhiteSpace String (optional) Whether to ignore white space from the beginning of records. Specify `true` or false.

Value of *skipWhiteSpace*	Meaning
`true`	Ignore white spaces at the beginning of records. This is the default.
`false`	Use records as they are. Specify `false` when the data contains positional data records.

keepResults String (optional) Whether you want the convertToValues service to return an IData object or to just validate the structure of the data in *edidata*. Specify `true` or false.

Value of *keepResults*	Meaning
`true`	Return an IData object in the output variable, *EDIValues*. This is the default.
`false`	Do *not* return an IData object in the output variable, *EDIValues*. Use this option when validating the structure of the *edidata* against the specified flat file schema.

validate String (optional) Whether you want the convertToValues service to return error messages describing how *edidata* differs from the specified flat file schema. Specify `true` or false.

Value of *validate*	Meaning
`true`	Return errors describing how the given *edidata* violates the constraints described in the flat file schema.
`false`	Do *not* return error messages describing how the *edidata* differs from the specified flat file schema. This is the default.

flag

Document (optional) Flags that you can set to govern convertToValues options.

Variables in *flag*	Description
addRecordCount	String Whether you want the service to add an additional field (`@record-count`) to each parsed record in the resulting IData object (*EDIValues*). The `@record-count` field is used to identify the record number of each parsed record.

Value of *addRecordCount*	Meaning
`true`	The `@record-count` field is added to each parsed record. This field contains the number of the parsed record. The first parsed record is 1, the second is 2, etc. If there are records that are undefined data, the count of the next defined record will reflect the undefined data. For example, if the `@record-count` field for a record is 2 and that record contains 5 undefined records, the `@record-count` field for the next defined record will be 8.
`false`	The `@record-count` field is *not* added to each parsed record. This is the default.

detailedErrors	String Whether you want detailed conditional validation error information. This flag is only used when *validate* is `true`.

Value of *detailedErrors*	Meaning
`true`	When a conditional validation error occurs, the output *errors* variable will contain detail information about all the conditions that were violated. For more information, see information about validation errors in the *Flat File Schema Developer's Guide*.
`false`	When a conditional validation error occurs, the service does *not* provide detail error information. Conditional validators report only whether a condition failed validation with no additional information about the conditions that were violated. This is the default.

returnErrors String (optional) How you want the convertToValues service to return error messages when *validate* is set to `true`. Specify one of the following.

Value of *validate*	Meaning
`asArray`	Return validation errors with the *edidata* in an array called *errors*. This is the default.
`inResults`	Return validation errors in the *EDIValues* object.
`both`	Return validation errors in both *errors* and *EDIValues*.

maxErrors String (optional) Maximum number of errors that you want returned when *validate* is set to `true`. When the flat file parser encounters more than the maximum number of errors within a record, the parser stops parsing and returns the parsed data and errors processed up until that point.

Output Parameters

EDIValues Document The *edidata* input data in IS document (IData object) format.

ediObject Object (optional) An object that encapsulates and keeps track of the input data segments during processing. It is used only when the *iterator* variable has been set to `true`. When all input data has been processed, the object becomes null. When the *ediObject* variable is `null`, you should exit out of the LOOP to discontinue processing. For an example of using the section about processing a document segment by segment in Chapter 3, "Receiving and Processing Inound EDI Documents" of the *webMethods EDI Module User's Guide*.

isValid String Whether the data in *edidata* is valid.

Value of *isValid*	Meaning
`true`	The *validate* input variable was set to `true` and no errors were found.
`false`	The *validate* input variable was set to `true` and errors were found, or the *validate* input variable was set to `false`.

errors Document List (optional) The validation errors, if any, that were found in *edidata*. Validation errors are returned in *errors* only if *validate* is set to `true` -AND- *returnErrors* is set to `asArray` or `both`. The list includes the path of the errors.

Usage Notes

- This service always returns the output IData object that contains the converted data in the *EDIValues* output parameter. When the input parameter *iterator* is set to `true`, the value of *EDIValues* is overwritten with the IData object for the record structure most recently converted. As a result, each time you invoke this service to convert a segment of the input document you should save the output or map it somewhere else.

- By default, each *recordWithNoID* record appears as a child of the record above it, in an array. Alternatively, you can set a flag to mimic the handling of *recordWithNoID*s that was implemented in version 4.6 of the Integration Server. That is, all *recordWithNoID* records appeared as children of the root. In addition, when the wm.b2b.edi:convertToValues service returned only one *recordWithNoID* record, it returned it as a single record, *not* as an array.

  If you would rather use this kind of *recordWithNoID* handling, set the following flag to true in the *Integration Server_directory*\packages\WmFlatFile\config\FlatFile.cnf file:

  ```
  recWithNoIDLike46=true
  ```

Example

See the sampleServices:Iterator810 service WmEDIsamples package, which is located in the Knowledge Base on the webMethods Advantage Web site at http://advantage.webmethods.com.

wm.b2b.edi:createIDOCtemplate

Creates a flat file schema for an IDOC by querying the SAP system for the IDOC name.

Input Parameters

serverName	String The name of the SAP server, e.g., `WMSAP1`.
packageName	String The name of the Integration Server package in which to create the flat file schema.
targetFolder	String The name of the IS folder in which to create the flat file schema.
IDOCname	String The name of the IDOC on the SAP server, e.g., `ORDERS02`.
IDOCversion	String The version of the IDOC.
table	String Number used to control the segment name generation. `1`: An IDOC segment will be generated as `E1EDK14`. `2`: The same segment will be generated as `E2EDK14`.
CIMtype	String (optional) If this is unknown, do not specify.
recordDelimiter	String The delimiter character that separates each record. The default is the new line character.
targetDictionary	String The name of the flat file dictionary that will hold the record, field, and composite definitions for this flat file schema.
sourceDictionaries	String List (optional) Names of flat file dictionaries in which to search for definitions of records, fields, and composites. If a definition already exists, refer to it instead of creating a new entry in the *targetDictionary*.

overwrite String Whether you want the createIDOCtemplate service to overwrite entries in the target dictionary if they already exist. Specify `false` or `true`.

Value of *overwrite*	Meaning
false	Do *not* overwrite flat file dictionary entries in the target dictionary if the entries already exist. This is the default.
true	Overwrite existing dictionary entries with values specified by this IDOC. If you specify true for *overwrite*, the createIDOCtemplate service ignores the *sourceDictionaries* variable.

Output Parameters

None

wm.b2b.edi:createTemplateFromSEF

This service will be deprecated in a future release. You should use wm.b2b.edi:SEFParse.

Creates a flat file schema object from a SEF file. The flat file schema is saved to the Integration Server namespace. The flat file schema describes the EDI document structure and validation criteria.

Input Parameters

transactionName String The EDI transaction set name (e.g., 850) for which you want to create a flat file schema.

SEFfileName String The full path and file name of the SEF file on your local file system. If the SEF file is located on a Web server, use wm.b2b.edi:SEFParse.

includeEnvelope String (optional) Whether you want the createTemplateFromSEF service to create envelope segments in the output flat file schema objects. Specify `false` or `true`.

Value of *includeEnvelope*	Meaning
false	The createTemplateFromSEF service does *not* create the envelope segments (i.e., ISA/IEA, GS/GE, UNB/UNZ, UNG/UNE) in the output flat file schema object. This is the default.
true	The createTemplateFromSEF service *does* create the envelope segments in the output flat file schema object.

isBigDocTemplate String Whether you will use the generated flat file schema to parse documents that are considered large. Specify `false` or `true`. For more information about large document handling, see the *webMethods EDI Module User's Guide*.

Value of *isBigDocTemplate*	Meaning
false	You will *not* use the generated flat file schema to parse large documents. This is the default.
true	The generated flat file schema will *not* have any nest structures. You can use this flat file schema with iterator.

targetSchema String The fully-qualified name that you want to assign the flat file schema that you are creating. Specify the name using the following naming convention:

`EDIFFSchema.standard.Vversion:Tname`, where:

- `standard` represents the EDI standard (e.g., `X12`).
- `version` represents the EDI standard version (e.g., `4010`).
- `name` represents the EDI transaction (e.g., `850`).

For example: `EDIFFSchema.X12.V4010:T850`

targetPackage String The name of the Integration Server package in which to create the flat file schema.

targetDictionary String The name of the flat file dictionary to hold the record, field, and composite definitions for this flat file schema.

overwriteDictionary String Whether you want the createTemplateFromSEF service to overwrite entries in the target dictionary if they already exist. Specify `false` or `true`.

Value of *overwriteDictionary*	Meaning
false	Do *not* overwrite flat file dictionary entries in the target dictionary if the entries already exist. This is the default.
true	Overwrite existing dictionary entries with values specified by this SEF file. If you specify true for *overwriteDictionary*, the createTemplateFromSEF service ignores the *sourceDictionaries* variable.

sourceDictionaries String List (optional) Names of flat file dictionaries in which to search for definitions of records, fields, and composites. If a definition already exists, refer to it instead of creating a new entry in the *targetDictionary*.

Output Parameters

None

wm.b2b.edi:createW3CXMLSchema

Creates a W3C XML schema from a SEF file on your local file system. (Because this service might be deprecated in the next version, you should begin using flat file schemas.)

Input Parameters

SEFfileName String The full path and file name of the SEF file on your local file system.

includeEnvelope String (optional) Whether you want the createW3CXMLSchema service to create envelope segments in the output W3C XML schema. Specify `false` or `true`.

Value of *includeEnvelope*	Meaning
false	The createW3CXMLSchema service does *not* create the envelope segments (i.e., ISA/IEA, GS/GE, UNB/UNZ, UNG/UNE) in the output W3C XML schema. This is the default.
true	The createW3CXMLSchema service *does* create the envelope segments in the output W3C XML schema.

transactionSet String The EDI transaction set name (e.g., 850) for which you want to create a flat file schema.

schemaFileName String The fully-qualified name that you want to assign the W3C XML schema that you are creating.

Output Parameters

XMLschema String The resulting W3C XML schema displayed as a String. The actual schema is saved to a file, and the IS document type creation uses the saved file. This output String is displayed at the end of the service execution for informational purposes only.

errors String List Error messages that the createW3CXMLSchema service encountered while creating the schema from the SEF file, if any.

wm.b2b.edi:envelopeProcess

For ANSI X12 and UN/EDIFACT documents only; not for use with TRADACOMS documents. Processes the envelopes in an inbound EDI document. Accepts an inbound EDI document, converts the envelope header segments (ISA/IEA, GS/GE, UNB/UNZ, UNG/UNE) to IData objects, and leaves the body of each constituent EDI document unprocessed.

If the document is not considered large, the document body remains beneath the transaction set header as an "unDefData" (undefined data) String. If the document is considered large, a pipeline with an "_RID_" (reservation ID) pointer is created, referring to the document in temporary storage. For more information about large document handling, see the *webMethods EDI Module User's Guide*.

While processing the envelope, the service optionally can validate the envelope against the predefined flat file schema for ANSI X12 or UN/EDIFACT documents. It also can perform compliance checks against the interchanges if specified.

Input Parameters

edidata String or InputStream The EDI document input to process. The data type (String or InputStream) is determined by the content handler associated with the inbound document. Input should not include manual line breaks.

validate String Whether you want to validate the envelopes against a predefined flat file schema.

Value of *validate*	Meaning
true	Validate the envelope against the predefined flat file schema. This service will report all the errors it finds through the *errorArray* variable. This is the default.
false	Do *not* validate the envelope.

complianceCheck String Whether you want to perform a compliance check against the interchange.

Value of *complianceCheck*	Meaning
true	Perform a compliance check against the entire interchange. The processEnvelope service stops executing after encountering the first error. This is the default.
false	Do *not* perform the compliance check.

userEnvelopeFF Schema	String (optional) A flat file schema that overrides the predefined flat file schema that *validate* uses. If the value is invalid, the output parameters *errorArray* and *lastError* will contain error information.

Note: To specify a flat file schema for the EDI envelope, make a copy of the EDI flat file schema from the wm.b2b.edi.EDIFFSchema folder and modify the validation criteria of the copy. If you have changed the envelope structure, the EDI Module might not correct the compliance variation.

Output Parameters

values	Document The resulting IData object with envelope segments expanded. The transaction set contents will remain as unparsed Strings in the pipeline or held in temporary storage.
standard	String The standard to which your EDI document adheres, e.g., `X12` or `UNEDIFACT`.
hasError	String Whether the validation or compliance check resulted in error.

Value of *hasError*	Meaning
false	If the *validate* is `true`, validation errors can be retrieved from *errorArray*. Otherwise, it indicates errors from compliance check.
true	No errors.

errorArray	Document List Array of error messages.

Usage Notes

- Besides serving as an envelope validation and compliance check service, this is the first in a series of conversion services you should invoke within a larger flow to turn each EDI transaction set into an autonomous, pure IS document (IData object). After it is in pure IData format, a transaction set can be individually validated, mapped, manipulated, etc.
- If you are working with non-EDI flat files, do not use this service for processing. Instead, see the *Flat File Schema Developer's Guide*.
- If the schema structure specified does not reflect the EDI envelope structure, the validation will fail.

wm.b2b.edi:SEFParse

Creates a flat file schema from a SEF file (Standard Exchange Format from Foresight Corporation) and saves the flat file schema in the Integration Server namespace. This flat file schema describes the EDI document structure and contains validation criteria.

Input Parameters

url
String (optional) The full URL of the SEF file from which you want to create a flat file schema. For example, if the SEF file is located in the `/pub` directory of the *WmEDIProduction* package on the Rubicon server, you type:
`http://rubicon:5555/WmEDIProduction/4010.sef`

If the SEF file is on a local file system, use the input variable, *SEFfilename*, instead of *url* and *method*.

method
String (optional) The HTTP method; specify `GET`. If you specify *url*, you must specify *method*.

auth
Document (optional) Authorization information that the HTTP service will submit if the resource specified in *url* is protected. The format of *auth* is:

Variable in *auth*	Description
type	String Type of authentication you want the HTTP service to use when it submits this request. Currently, only basic authentication is supported. If you are accessing a protected resource, set *auth* to `Basic`.
user	String The user name that the SEFParse service will submit when requesting a protected resource.
pass	String The password associated with *user*.

data Document (optional) Data that you want the HTTP service to submit with the HTTP request. Specify your data in *one* of the following elements:

Variable in *data*	Description
args	Document An IS document (IData object) that contains the *name/value* pairs that you want the HTTP service to submit to the resource in *url*. Create one element for each *name/value* pair, where *name* is the element's name and *value* is the value of the element. Note that when you use *args*, the HTTP service will: ■ Automatically url-encode the *name/value* pairs. You do not need to url-encode the values. ■ Automatically insert the "&" character between pairs. You do not need to include it. ■ Automatically prefix the entire query string with the "?". You do not need to include this character.
string	String A string of text that you want the HTTP service to submit to the resource in *url*. If you use *string* to submit data, make sure that you specify the string *exactly* as you want it presented in the HTTP request. Also make sure you url-encode the contents of *string*.
bytes	String Table Data that the HTTP service will use to construct a query string to submit to the resource specified in *url*. Note that the input variable, *bytes,* is similar to *args*, but *bytes* allows you to submit unnamed values in a query string, not just name/value pairs. Specify each value you want to submit in a separate row in the String Table. When you specify the String, ensure: ■ The contents of column 0 represent the name portion of the pair (leave this column null to submit an unnamed value). ■ The contents of column 1 represent the value portion of the pair.

Note that when you use *bytes*, the HTTP service will:

- Automatically url-encode the name/value pair. You do not need to url-encode the values.
- Automatically insert the "&" character between the pairs (or unnamed values) that it constructs. You do not need to include it.
- Automatically prefix the entire query string with the "?" character if it submits the data in table via the GET method. You do not need to include this character.

headers

Document (optional) Fields that you want to explicitly override in the HTTP request header that the HTTP service issues.

Specify one element in the *headers* IS document (IData object) for each header field that you want to set, where:

- The element's name represents the name of the header field.
- The element's value represents the value of that header field.

If you do not set *headers*, the HTTP service will use its default header values.

EDIDocName

String The EDI transaction set name (e.g., 850) or TRADACOMS file name (e.g., INVOIC), for which you want to create a flat file schema. Specify only a single value.

SEFfileName

String (optional) The full path and file name of the SEF file on your local file system. If the SEF file is located on a Web server, use the *url* and *method* variables instead of *SEFfileName*.

includeEnvelope

String (optional) Whether you want the SEFParse service to create envelope segments in the output flat file schema objects. Specify `false` or `true`.

Value of *includeEnvelope*	Meaning
false	The SEFParse service does *not* create the envelope segments (i.e., ISA/IEA, GS/GE, UNB/UNZ, UNG/UNE, STX/END, BAT/EOB, MHD/MTR) in the output flat file schema object. This is the default.
true	The SEFParse service *does* create the envelope segments in the output flat file schema object.

isBigDocTemplate String Whether you will use the generated flat file schema to parse documents that are considered large. Specify `false` or `true`. For more information about large document handling, see the *webMethods EDI Module User's Guide*.

Value of *isBigDocTemplate*	Meaning
false	You will *not* use the generated flat file schema to parse large documents. This is the default.
true	You *will* use the generated flat file schema to parse large documents.

version String (optional) The version of the EDI standard. For TRADACOMS, specify the version of the TRADACOMS File document type.

targetSchema String The fully qualified name that you want to assign the flat file schema that you are creating. For all supported EDI standards except TRADACOMS, use the following naming convention for *targetSchema* if you plan to use Trading Networks:

`EDIFFSchema.`*`standard`*`.V`*`version`*`:T`*`transaction`*, where:

- *`standard`* represents the EDI standard (e.g., `X12`).
- *`version`* represents the EDI standard version (e.g., `4010`).
- *`transaction`* represents the EDI transaction (e.g., `850`).

For example: `EDIFFSchema.X12.V4010:T850`

For the TRADACOMS EDI standard, the wm.b2b.edi:SEFParse service creates a temporary flat file schema. Use the following naming convention for *targetSchema*:

`EDIFFSchema.Tradacoms.V`*`version`*`.T`*`name`*`:TEMP_SCHEMA`, where:

- *version* represents the version of the TRADACOMS File document type (for example, `v2`)
- *name* represents the name of the TRADACOMS File document type (for example, `ORDHDR`)

This temporary flat file schema contains all the messages contained in the TRADACOMS file. Because the flat file parser (the wm.b2b.edi.tradacoms:convertToValues service) cannot always properly parse this temporary flat file schema, you should then execute the wm.b2b.edi.tradacoms.ui:modifyTradacomsSchema service to split the flat file schema into one flat file schema per MHD segment in the TRADACOMS file. This temporary flat file schema will be deleted upon successful execution of the wm.b2b.edi.tradacoms.ui:modifyTradacomsSchema service.

Note: You *must* also specify a value for the *targetDictionary* parameter.

If you use the migration utilities to move templates from webMethods version 4.*x* to flat file schemas in 6.0.x, the utility creates the flat file schemas using the naming convention described above. For more information about migrating templates, see the section about flat file schemas in Chapter 1, "Before You Can Process EDI Documents" of the *webMethods EDI Module User's Guide*.

targetPackage String The name of the Integration Server package in which to create the flat file schema.

targetDictionary String The name of the flat file dictionary to hold the record, field, and composite definitions for this flat file schema.

overwriteDictionary String Whether you want the SEFParse service to overwrite entries in the target dictionary if they already exist. Specify `false` or `true`.

Value of *overwriteDictionary*	Meaning
false	Do *not* overwrite flat file dictionary entries in the target dictionary if the entries already exist. This is the default.
true	Overwrite existing dictionary entries with values specified by this SEF file. If you specify true for *overwriteDictionary*, the SEFParse service ignores the *sourceDictionaries* variable.

sourceDictionaries String List (optional) Names of flat file dictionaries in which to search for definitions of records, fields, and composites. If a definition already exists, refer to it instead of creating a new entry in the *targetDictionary*.

Output Parameters

None

Usage Notes

- To view the flat file schema that you create using the SEFParse service in the webMethods Developer, you must refresh your connection to the Developer.
- If a EDI document contains multiple consecutive HL segments, this service will create a flat file schema that contains a single HL record. That record will be a superset of all the HL segment definitions in the original SEF file.

Example

See the wm.b2b.edi:createTemplateFromSEF service.

wm.b2b.edi.migration

Use these services to migrate EDI Module version 4.*x* templates to EDI version 6.*x* flat file schemas.

For more information about migrating templates, see the section about flat file schemas in Chapter 1, "Before You Can Process EDI Documents" of the *webMethods EDI Module User's Guide*.

wm.b2b.edi.migration:migrateTemplate

Creates a flat file schema from the specified template.

Input Parameters

oldTemplate Document Current name of the template that you want to migrate.

templateName String Target name of the template you want to migrate.

documentName String Name for the new flat file schema you want to create.

targetFolder String Folder in which to create the flat file schema.

targetPackage String Package in which to create the flat file schema.

targetDictionary String The name of the flat file dictionary that will hold the record, field, and composite definitions for this schema.

sourceDictionaries String List (optional) Names of flat file dictionaries in which to search for definitions of records, fields, and composites. If a definition already exists, refer to it instead of creating a new entry in the *targetDictionary*.

overwrite String (optional) Whether you want the migrateTemplate service to overwrite entries in the target dictionary if they already exist. Specify `false` or `true`.

Value of *overwrite*	Meaning
false	Do *not* overwrite flat file dictionary entries in the target dictionary if the entries already exist. This is the default.
true	Overwrite existing dictionary entries with values specified by this template.

Output Parameters

warnings String List List of warning messages that might reflect migration errors.

wm.b2b.edi.templateMgr

Use the services in this folder to help you save and manage EDI template objects.

wm.b2b.edi.templateMgr:getProperties

Returns a list of the system properties. You can narrow the properties using the *type* input variable.

Input Parameters

type — String The component for which you want to retrieve properties. You can use this input to narrow down the output list. Specify one of the following:

Value of *type*	Meaning
EDI	Return the properties for the EDI Module.
tn	Return the properties for Trading Networks.
server	Return the properties for the Integration Server.
all	Return the properties for the EDI Module, Trading Networks, and the Integration Server.

Output Parameters

propertyValue — String List The list of properties.

Usage Notes

The properties are returned in the pipeline with the property name and value.

wm.b2b.edi.templateMgr:getTemplate

This service will be deprecated in a future release.

Constructs a namespace name for a flat file schema based on a specified formula.

Input Parameters

name — String The name of the template that you previously saved, e.g., `850`.

packageName — String (optional) Ignored.

standard — String (optional) The name of the EDI standard for which the template is used, e.g., `ANSI`, `UNEDIFACT`, `UCS`, `VICS`, `ODETTE`, or `EANCOM`.

version — String (optional) The standard version of the transaction set for which the template is used, e.g., `4010` for ANSI X12, or `98A` for UN/EDIFACT.

Output Parameters

templateObject	Document If the flat file schema was created following the standard described in the following "Usage Notes," this variable contains a String called *templateName* that gives the namespace name of the corresponding flat file schema.

Usage Notes

The location of the schema (the naming convention) is determined as follows:

- `EDIFFstandard` represents the EDI standard (e.g., `X12`).
- `version` represents the EDI standard version (e.g., `4010`).
- `name` represents the EDI transaction (e.g., `850`).

For example: `EDIFFSchema.X12.V4010:T850`

If the migration utilities are used to move templates from webMethods version 4.*x* to 6.*x*, they will have been created using this naming convention.

wm.b2b.edi.templateMgr:init (For Internal Use Only)

A startup Java service that is automatically invoked when the WmEDI package starts. This service registers the EDI content handler and initializes the IS document (IData) template object in memory. For internal use only.

wm.b2b.edi.tradacoms

The services in this folder are the core services that you use when converting between TRADACOMS EDI documents and IS documents (IData objects), and when creating flat file schemas.

wm.b2b.edi.tradacoms:convertToString

Converts an IS document (IData object) to a String based on a flat file schema that you specify.

The difference between this service and the pub.flatFile:convertToString service is that this service handles TRADACOMS EDI files.

The service will optionally fill in the counters and control numbers if they are empty. If you want to control the counters or control numbers, modify the IS document (IData object) prior to invoking this service to convert it to a String.

In addition, the service will automatically provide values for the following fields if they are not present:

Segment	Field	Value provided
END	NMST	The number of messages or batches in the transmission.
EOB	NMST	The number of MHD segments in the batch.
MHD	MSRF (for each MHD segment)	A consecutive count of MHD segments in the transmission or batch, starting from 1 and incrementing by 1 for each MHD.
MTR	NOSG	The number of segments from the last MHD to this MTR segment, including both the MHD and MTR segments.

Input Parameters

values Document The IS document (IData object) object that you want to convert to a String.

tradacomsFFSchema String The fully-qualified namespace name of the flat file schema to use to convert the specified IS document (IData object) (in *values*) to a String.

spacePad String How you want the resulting String to be justified. Specify one of the following:

Value of *spacePad*	Meaning
`left`	Left justify.
`right`	Right justify.
`none`	No justification. This is the default.

noEmptyTrailingFields String Whether to remove empty trailing fields from records. The convertToString service only uses this variable for records that have delimited fields. Specify `true` or `false`.

Value of *noemptyTrailingFields*	Meaning
`true`	The convertToString service removes empty trailing fields from the output. For example, a record with empty trailing fields might look like the following: `AAA*01*02!` (where `!` is the segment terminator). This is the default.
`false`	The convertToString service does *not* remove empty trailing fields. Instead it uses the field separator to denote an empty field. For example, a record with empty trailing field might look like the following: `AAA*01*02********!` (where * is the field separator and `!` is the segment terminator).

FormatInfo Document (optional) Values you want the convertToString service to pass unmodified to all format services it invokes.

outputFileName String (optional) The name of the file to which you want the String output written. If you do not specify *outputFileName* the output is not written to a file.

encoding String The type of encoding used to write data to the output file. The default encoding is UTF-8.

startAt String Allows the convertToString service to start at a specific record in the flat file schema used to create the output string. Specify the path to the element where you want to start composing the output string.

countSegments String Whether to count the number of segments written to the output file.

`true`	The convertToString service counts the number of segments written to the output file and returns that number in the output parameter *segmentCount*. This is the default.
`false`	The convertToString service does *not* count the number of segments written to the output file.

Output Parameters

string String The output String that represents the data specified in the input variable, *Values*.

errorArray String List Error messages describing the errors that the convertToString service encountered during conversion. If the convertToString service did not encounter errors, *errorArray* is null.

segmentCount String The number of records written; only returned when *countSegments* is `true`.

wm.b2b.edi.tradacoms:convertToValues

Converts an InputStream or String (i.e., a TRADACOMS file) to an IS document (IData object) based on the input flat file schemas.

Input Parameters

tradacomsData String or InputStream The TRADACOMS data you want to convert to an IData object.

ediObject Object (optional) An object that encapsulates and keeps track of the input data segments during processing. It is used only when the *iterator* variable has been set to `true`.

encoding String (optional) The encoding of the data passed in to *tradacomsData*.

tradacomsFFSchema String (optional) The fully-qualified name of the flat file schema object used to parse the *tradacomsData* object.

iterator String (optional) Whether you want to process segments one at a time or process all input data at one time. Specify `true` or `false`.

Value of *iterator*	Meaning
`true`	The convertToValues service starts processing segment structures with a top-level record as defined by the flat file schema. The service returns to the caller when it encounters another top-level record in the input data. The next time the service is invoked, it begins processing the input data where it left off.
`false`	The convertToValues service processes all input data at one time. This is the default.

nullable String (optional) Whether to create an IS document (IData object) if all fields are null. Specify `true` or false.

Value of *nullable*	Meaning
`true`	Do not create an IS document (IData object) if all the fields are null. This is the default.
`false`	Always create an IS document even though all the fields are null.

skipWhiteSpace String (optional) Whether to ignore white space from the beginning of records. Specify `true` or false.

Value of *skipWhiteSpace*	Meaning
`true`	Ignore white spaces at the beginning of records. This is the default.
`false`	Use the records as they are. Specify `false` when the data contains positional data records.

keepResults String (optional) Whether you want the convertToValues service to return an IData object or to just validate the structure of the data in *tradacomsData*. Specify `true` or false.

Value of *keepResults*	Meaning
`true`	Return an IData object in the output variable, *EDIValues*. This is the default.
`false`	Do *not* return an IData object in the output variable, *EDIValues*. Use this option when validating the structure of the *tradacomsData* against the specified flat file schema.

validate String (optional) Whether you want the convertToValues service to return error messages describing how *tradacomsData* differs from the specified flat file schema. Specify `true` or false.

Value of *validate*	Meaning
`true`	Return errors describing how the given *tradacomsData* violates the constraints described in the flat file schema.
`false`	Do *not* return error messages describing how the *tradacomsData* differs from the specified flat file schema. This is the default.

flag

Document (optional) Flags that you can set to govern convertToValues options.

Variables in *flag*	Description
addRecordCount	String Whether you want the service to add an additional field (`@record-count`) to each parsed record in the resulting IData object (*EDIValues*). The `@record-count` field is used to identify the record number of each parsed record.

Value of *addRecordCount*	Meaning
`true`	The `@record-count` field is added to each parsed record. This field contains the number of the parsed record. The first parsed record is 1, the second is 2, etc. If there are records that are undefined data, the count of the next defined record will reflect the undefined data. For example, if the `@record-count` field for a record is 2 and that record contains 5 undefined records, the `@record-count` field for the next defined record will be 8.
`false`	The `@record-count` field is *not* added to each parsed record. This is the default.

Variables in *flag*	Description
detailedErrors	String Whether you want detailed conditional validation error information. This flag is only used when *validate* is `true`.

Value of *detailedErrors*	Meaning
`true`	When a conditional validation error occurs, the output *errors* variable will contain detail information about all the conditions that were violated. For more information, see information about validation errors in the *Flat File Schema Developer's Guide*.
`false`	When a conditional validation error occurs, the service does *not* provide detail error information. Conditional validators report only whether a condition failed validation with no additional information about the conditions that were violated. This is the default.

returnErrors String (optional) How you want the convertToValues service to return error messages when *validate* is set to `true`. Specify one of the following.

Value of *validate*	Meaning
`asArray`	Return validation errors with the *tradacomsData* in an array called *errors*. This is the default.
`inResults`	Return validation errors in the *tradacomsValues* object.
`both`	Return validation errors in both *errors* and *tradacomsValues*.

maxErrors String (optional) Maximum number of errors that you want returned when *validate* is set to `true`. When the flat file parser encounters more than the maximum number of errors within a record, the parser stops parsing and returns the parsed data and errors processed up until that point.

Output Parameters

tradacomsValues Document The *tradacomsData* input data in IS document (IData object) format.

ediObject Object (optional) An object that encapsulates and keeps track of the input data segments during processing. It is used only when the *iterator* variable has been set to `true`. When all input data has been processed, the object becomes null. When the *ediObject* variable is `null`, you should exit out of the LOOP to discontinue processing. For an example of using the section about processing a document segment by segment in Chapter 3, "Receiving and Processing Inound EDI Documents" of the *webMethods EDI Module User's Guide*.

isValid String Whether the data in *tradacomsData* is valid.

Value of *isValid*	Meaning
`true`	The *validate* input variable was set to `true` and no errors were found.
`false`	The *validate* input variable was set to `true` and errors were found, or the *validate* input variable was set to `false`.

errors Document List (optional) The validation errors, if any, that were found in *tradacomsData*. Validation errors are returned in *errors* only if *validate* is set to `true` -AND- *returnErrors* is set to `asArray` or `both`. The list includes the path of the errors.

Usage Notes

- This service always returns the output IData object that contains the converted data in the *tradacomsValues* output parameter. When the input parameter *iterator* is set to `true`, the value of *tradacomsValues* is overwritten with the IData object for the record structure most recently converted. As a result, each time you invoke this service to convert a segment of the input document you should save the output or map it somewhere else.

- By default, each *recordWithNoID* record appears as a child of the record above it, in an array. Alternatively, you can set a flag to mimic the handling of *recordWithNoID*s that was implemented in version 4.6 of the Integration Server. That is, all *recordWithNoID* records appeared as children of the root. In addition, when the wm.b2b.edi:convertToValues service returned only one *recordWithNoID* record, it returned it as a single record, *not* as an array.

  If you would rather use this kind of *recordWithNoID* handling, set the following flag to true in the *Integration Server_directory*\packages\WmFlatFile\config\FlatFile.cnf file:

  ```
  recWithNoIDLike46=true
  ```

wm.b2b.edi.tradacoms.compose

Use the services in this folder to create TRADACOMS documents that exceed the Large Document threshold. For information about the Large Document threshold, see Chapter 7, "Handling Large Documents", in the *webMethods EDI Module User's Guide*.

wm.b2b.edi.tradacoms.compose:addToTradacomsTransmission

Use this service to add message segments to the *tradacomsTransmission* object that the wm.b2b.edi.tradacoms.compose:startTradacomsTransmission service returned.

Input Parameters

tradacomsTransmission	Object The *tradacomsTransmission* object that the wm.b2b.edi.tradacoms.compose:startTradacomsTransmission service returned.
values	Document Message segments that will be written to the *tradacomsTransmission* object. You can specify one or more header messages, detail messages, and trailer messages. In addition you may specify a VAT trailer, if appropriate. Note: You are responsible for entering segments in the sequence that the TRADACOMS standard expects. This service does *not* sort messages into the proper sequence.
tradacomsFFSchema	String The flat file schema used to convert *values* to a String.
formatInfo	Document (optional) Values you want the addToTradacomsTransmission service to pass unmodified to all format services it invokes.
startAt	String Optional. Allows the addToTradacomsTransmission service to start at a specific record in the flat file schema used to create the output string. Specify the path to the element where you want to start composing the output string.

Output Parameters

tradacomsTransmission	Object Used to write transmission information to disk or held in memory. If the document exceeds the Large Document threshold, or if the *outputFileName* is specified, it will be written to disk. Otherwise, the completed document will be held in memory. For information about the Large Document threshold, see Chapter 7, "Handling Large Documents" in the *webMethods EDI Module User's Guide*.

wm.b2b.edi.tradacoms.compose:endTradacomsTransmission

Use this service to create an END segment for a transmission. This service will also:

- Add an EOB (End of Batch) segment to close the batch document if the wm.b2b.edi.tradacoms.compose:startTradacomsBatch service was invoked for this transmission
- Write an RSGRST (reconciliation) message if you set the *includeRSGRSG* parameter to true in the wm.b2b.edi.tradacoms.compose:startTradacomsTransmission service.

Input Parameters

END	Document
END01	String (optional) The number of messages in the transmission.
formatInfo	Document Format information used to create an STX segment. This value should match the *formatInfo* value that you specified in the wm.b2b.edi.tradacoms.compose:startTradacomsTransmission service that you used to create the transmission.
tradacomsTransmission	Object The *tradacomsTransmission* object that the wm.b2b.edi.tradacoms.compose:startTradacomsTransmission service returned.

Output Parameters

InputStream	Object The input stream containing the entire document that has been written to disk or held in memory.

wm.b2b.edi.tradacoms.compose:startTradacomsBatch

Use this service to create a batch (BAT) segment for a transmission. This service will also add an EOB (End of Batch) segment to close any previous batch if the wm.b2b.edi.tradacoms.compose:startTradacomsBatch service has been called previously on the input *tradacomsTransmission* object.

Input Parameters

BAT	Document
BAT01	String The recipient's transmission reference.
formatInfo	Document Format information used to create an STX segment. This value should match the *formatInfo* value that you specified in the wm.b2b.edi.tradacoms.compose:startTradacomsTransmission service that you used to create the transmission.
tradacomsTransmission	Object The *tradacomsTransmission* object that the wm.b2b.edi.tradacoms.compose:startTradacomsTransmission service returned.

Output Parameters

tradacomsTransmission Object Used to write transmission information to disk or held in memory. If the document exceeds the Large Document threshold, or if the *outputFileName* is specified, it will be written to disk. Otherwise, the completed document will be held in memory.

wm.b2b.edi.tradacoms.compose:startTradacomsTransmission

Use this service to create an STX segment for a transmission. This service returns a TRADACOMS transmission object.

Input Parameters

STX Document Contains the following information needed to create an STX segment:

STX01 Document

- STDS01 Document Syntax rules identifier.
- STDS02 String Version

STX02 Document

- FROM01 String Identification of transmission sender *Code*.
- FROM02 String Identification of transmission sender *Name*.

STX03 Document

- UNTO01 String Identification of transmission recipient *Code*.
- UNTO02 String Identification of transmission recipient *Name*.

STX04 Document

- TRDT01 String Date of transmission.
- TRDT02 String Time of transmission.

STX05 String Sender's transmission reference.

STX06 String Recipient's transmission reference.

STX07 String Application reference.

STX08 String Transmission priority code.

outputFileName String (optional) The file to which to write the document to disk. If this field is blank, the document is held in memory.

formatInfo Document Format information used to create an STX segment.

encoding String The encoding used to write the document to disk.

includeRSGRSG Whether the service creates an RSGRSG message as the last message in the output document.

Value of *includeRSGRSG*	Meaning
`false`	Do not create an RSGRSG message. This is the default.
`true`	Create a transmission with an RSGRSG message as the last message in the output document.

Output Parameters

tradacomsTransmission Object Used to write transmission information to disk or to memory. If the document exceeds the Large Document threshold, or if the *outputFileName* is specified, it will be written to disk. Otherwise, the completed document will be held in memory. For information about the Large Document threshold, see Chapter 7, "Handling Large Documents", in the *webMethods EDI Module User's Guide*.

wm.b2b.edi.tradacoms.doc

Use the services in this folder to extract information about TRADACOMS files.

wm.b2b.edi.tradacoms.doc:getContentPart

Returns a Trading Networks *bizDocEnvelope* content part object that represents the requested content part.

Input Parameters

bizDocEnvelope — Object The BizDocEnvelope that represents the TRADACOMS file.

Note: When you invoke the wm.tn.doc:view service to retrieve the *bizdoc* document from the Trading Networks database, make sure that you retrieve the contents of the *bizdoc* by setting that service's *getContent* input parameter to true. For more information about the wm.tn.doc:view service, see the *webMethods Trading Networks Built-In Services Reference*.

partType — String The content part of the message to be returned. Specify one of the following:

Value of *partType*	Meaning
`Header`	The message of the file that contains standard header information, for example, MHD type of INFVIL or GENHDR.
`Detail`	The message of the file that contains the body of the message, for example, MHD type of INVOIC or GENRAL. If this input value is specified, then the *detailIndex* input parameter is required.
`VAT`	The message of the file that contains VAT information, for example, VATTLR.
`Trailer`	The message of the file that contains trailer information, for example, INVTLR or GENTLR.
`Transmission Header`	The STX segment and BAT segment (if present) for the transmission that contains this file.
`Transmission Trailer`	The END segment, EOB segment (if present), and reconciliation message (if present) for the transmission that contains this file.

detailIndex — String (Optional) The number of the detail message to extract from *bizDocEnvelope*. If the value of *partType* is not `Detail`, then *detailIndex* is ignored.

Output Parameters

contentPart	Object A Trading Networks *bizDocEnvelope* containing a representation of the specified content part.

Usage Notes

- This service throws an exception if:
  - *bizDocEnvelope* is null
  - *partType* is null or is not an allowed value
  - The value of *partType* is `Detail` and one of the following is true:

    No detail index is specified

    The detail index is not a number greater than or equal to zero and less than the number of detail messages
  - An error occurs while reading the document from the database

wm.b2b.edi.tradacoms.doc:getDocumentPartInfo

Returns information about the detail messages in a TRADACOMS file that is contained in a Trading Networks *bizDocEnvelope*.

Input Parameters

bizDocEnvelope	Object The bizDocEnvelope that represents the TRADACOMS file.

Output Parameters

numberOfDetail Messages	String The number of detail messages contained in the TRADACOMS file.
hasVAT	String Whether the file contains VAT information.

Value of *hasVAT*	Meaning
`true`	The file contains VAT information.
`false`	The file does not contain VAT information.

Usage Notes

This service throws an exception if *bizDocEnvelope* is null or is not a TRADACOMS file.

wm.b2b.edi.tradacoms.doc:getDocumentStream

Returns the document input stream of a TRADACOMS file that is contained in a *bizDocEnvelope*.

Input Parameters

bizDocEnvelope	Object The BizDocEnvelope that represents the TRADACOMS file.

Output Parameters

inputStream	Object An input stream containing the entire document that has been written to disk or held in memory. The input stream will contain all content parts, in the original sequence.
encoding	String The encoding that can be used to convert the input stream to a string.

Usage Notes

This service throws an exception if *bizDocEnvelope* is null or is not a TRADACOMS file.

wm.b2b.edi.tradacoms.doc:getFFSchemaNames

Returns the names of the flat file schemas that can be used to parse the parts of a TRADACOMS file.

Input Parameters

bizDocEnvelope	Object The BizDocEnvelope that represents the TRADACOMS file.

Output Parameters

headerFFSchema	String The namespace name of the flat file schema that can be used to parse the header message of the TRADACOMS file contained in *BizDocEnvelope*.
detailFFSchema	String The namespace name of the flat file schema that can be used to parse the detail message of the TRADACOMS file contained in *BizDocEnvelope*.
vatFFSchema	String The namespace name of the flat file schema that can be used to parse the VAT message (if present) of the TRADACOMS file contained in *BizDocEnvelope*.
trailerFFSchema	String The namespace name of the flat file schema that can be used to parse the trailer message of the TRADACOMS file contained in *BizDocEnvelope*.

wm.b2b.edi.tradacoms.doc:isFileEnvelope

Determines whether a BizDocEnvelope contains a TRADACOMS File document.

Input Parameters

bizDocEnvelope Object The BizDocEnvelope that represents the TRADACOMS file.

Output Parameters

isFileEnvelope String Whether *bizDocEnvelope* contains a TRADACOMS file.

Value of *isFileEnvelope*	Meaning
`true`	*bizDocEnvelope* contains a TRADACOMS file.
`false`	*bizDocEnvelope* does not contain a TRADACOMS file.

wm.b2b.edi.tradacoms.ui

Use the service in this folder to split a TRADACOMS flat file schema into multiple flat file schemas: one flat file schema per message in the File.

wm.b2b.edi.tradacoms.ui:modifyTradacomsSchema

Splits a flat file schema that the wm.b2b.edi:SEFParse service created for a TRADACOMS file into multiple flat file schemas: one flat file schema per message in the file.

The wm.b2b.edi:SEFParse service creates a temporary flat file schema in the following location:

`EDIFFSchema.Tradacoms.V`*version*`.T`*name*`:TEMP_SCHEMA`

This temporary flat file schema contains all the messages contained in the TRADACOMS file. Because the flat file parser (the wm.b2b.edi.tradacoms:convertToValues service) cannot always properly parse this temporary flat file schema, you should execute the modifyTradacomsSchema service to split the flat file schema into one flat file schema per MHD segment in the TRADACOMS file.

Input Parameters

schemaName	String The name of the temporary flat file schema created by the wm.b2b.edi:SEFParse service. This temporary flat file schema will be deleted upon successful execution of this service.
standard	String The EDI standard; value must be `Tradacoms`.
version	String The version of the TRADACOMS file that is being created.
docType	String The name of the message header for the particular TRADACOMS file. For example, the value for the ORDERS file, you should specify ORDHDR.

Output Parameters

None

wm.b2b.edi.util

Use the services in this folder to help process document objects.

wm.b2b.edi.util:addGroupEnvelope

For outbound EDI documents, adds a group envelope (GS and GE segments) according to the ANSI X12, UCS, or VICS standards.

Note: For UN/EDIFACT EDI documents, use the wm.b2b.edi.util:addGroupEnvelopeEDIFACT service instead.

Input Parameters

documents **String List** Documents to which to add a group envelope.

IDcode **String** The Functional ID Code of the EDI document according to the EDI standard.

senderQual **String** EDI ID qualifier for the sender ID. It is used with *sender* to obtain the Trading Networks internal ID.

sender **String** The sender to identify in the group envelope. For example, if you specify 01 for *senderQual* (indicating a D-U-N-S number), specify the value of the D-U-N-S number for *sender*.

receiverQual **String** EDI ID qualifier for the receiver ID. It is used with *receiver* to obtain the Trading Networks internal ID.

receiver **String** The receiver to identify in the group envelope. For example, if you specify 01 for *receiverQual* (indicating a D-U-N-S number), specify the value of the D-U-N-S number for *receiver*.

ctlFromTable **String** (optional) Whether you want the service to obtain the control number for the group from the EDIControlNumber table, which is an EDI Module-specific table in the Trading Networks database. Specify `true` or `false`.

Value of *ctlFromTable*	Meaning
true	Obtain the control number from the EDIControlNumber table.
false	Do *not* obtain the control number from the EDIControlNumber table. Use the value specified in the *grpCtlNumber* variable.

grpCtlNumber **String** (optional) The group control number of the EDI document.

agencyCode **String** (optional) Responsible agency code: `T` (default) or `X`.

verRelCode **String** EDI standard version and release code, e.g., `4010`, `3040`, etc.

delimiters Document (optional) Delimiters used in the outbound EDI document.

Variable in *delimiters*	Description
record	String The segment terminator for the EDI document, e.g., `\u000a` to use the new line character. The default is the new line character.
field	String The field separator for each EDI segment e.g., `!`. The default is the * character.
subfield	String The separator for composite elements, e.g., ^. The default is the : character.

Output Parameters

outDocument String Contains the outbound EDI string.

Usage Notes

- Only use this service for ANSI X12, UCS, or VICS standards.
- All ANSI X12 documents *must* have a group envelope. You can add a group envelope *either* by:
  - Using this service and setting the *addGroup* input variable of the wm.b2b.edi.util:addICEnvelope service to `false`. Use this method to control the values used in the group envelope because you can specify them using the input variables of the addGroupEnvelope service.
  - Using wm.b2b.edi.util:addICEnvelope service and setting the *addGroup* input variable to `true`. Use this method if you do not need to control the values used in the group envelope, that is if you can accept the defaults.

  Choose only one of these methods. Failing to add a group envelope or creating a group envelope twice will result in an invalid document.
- The *ctlFromTable* input variable can be used to control the group control number. If the control number is empty and *ctlFromTable* is set to `true`, the group control number for the group type and version is retrieved from the EDIControlNumber table. If *ctlFromTable* is set to `true`, make sure that *senderQual* and *receiverQual* variables are specified.

wm.b2b.edi.util:addGroupEnvelopeEDIFACT

For outbound EDI documents, adds a group envelope (UNG and UNE segments) according to the UN/EDIFACT standard and its sub-standards ODETTE and EANCOM.

Note: For ANSI X12 documents, use the wm.b2b.edi.util:addGroupEnvelope service instead.

Input Parameters

documents String List Documents to which to add a group envelope.

IDcode String The Functional ID Code of the EDI document according to the EDI standard.

syntaxVersion String Syntax version of the envelope level.

senderID String (optional when *syntaxVersion* is greater than 3) The sender to identify in the group envelope. For example, if you specify 01 for *senderQual* (indicating a D-U-N-S number), specify the value of the D-U-N-S number for *sender*.

senderQual String (optional) EDI ID qualifier for the sender ID.

receiverID String (optional when *syntaxVersion* is greater than 3) The receiver to identify in the group envelope. For example, if you specify 01 for *receiverQual* (indicating a D-U-N-S number), specify the value of the D-U-N-S number for *receiver*.

receiverQual String (optional) EDI ID qualifier for the receiver ID.

grpCtlNumber String (optional) The group control number of the EDI document. If a group control number is *not* specified, the service obtains the control number from the EDIControlNumber table. That is, the service behaves as if the *ctlFromTable* variable is set to true.

ctlFromTable String (optional; this variable is used only when the *grpCtlNumber* variable is not specified) Whether you want the service to obtain the control number for the group from the EDIControlNumber table, which is an EDI Module-specific table in the Trading Networks database. Specify `true` or `false`.

Value of *ctlFromTable*	Meaning
true	Obtain the control number from the EDIControlNumber table.
false	Do *not* obtain the control number from the EDIControlNumber table. Use a random generated number as the value.

prodMode String (optional) The production mode associated with control number: `Production` (the default) or `Test`.

agencyCode String (optional) Responsible agency code: `UN` (default) or `AA`.

verCode String (optional) EDI standard version code, e.g., `D`, `S`, etc.

relCode String (optional) EDI message standard release code, e.g., `96A`, `97B`, etc.

aACode String (optional) EDI message standard assigned code, e.g., `OD`, `EN`, etc.

password String (optional) The recipient transmission reference password.

delimiters Document (optional) Delimiters used in the outbound EDI document.

Variable in *delimiters*	Description
record	String The segment terminator for the EDI document, e.g., `\u000a` to use the new line character. The default is the new line character.
field	String The field separator for each EDI segment e.g., `!`. The default is the * character.
subfield	String The separator for composite elements, e.g., ^. The default is the : character.
release	String The release character for composite elements, e.g., ^. The default is the `?` character.
decimal	String The release character for composite elements, e.g., `.` (the period character, which is the default).

Output Parameters

outDocument String Contains the outbound EDI string.

wm.b2b.edi.util:addICEnvelope

For an outbound EDI document, adds an Interchange (IC) envelope (ISA and IEA segments) and a group envelope if specified, according to the EDI ANSI X12, UCS, or VICS standards.

If you use the wm.b2b.edi.util:addGroupEnvelope service to add group envelopes, set the *addGroup* variable in the wm.b2b.edi.util:addICEnvelope service to `false`.

Note: To add an IC envelope to a UN/EDIFACT document, see the wm.b2b.edi.util:addICEnvelopeEDIFACT service.

Input Parameters

documents	**String List** EDI documents to which to add an interchange envelope. The EDI document might be wrapped with group envelopes.
authQual	**String** Authorization qualifier for the interchange envelope.
authInfo	**String** Authorization information for the interchange envelope.
securityQual	**String** Security qualifier for the interchange envelope.
securityInfo	**String** Security information for the interchange envelope.
senderQual	**String** EDI ID qualifier for the sender ID. It is used with *sender* to obtain the Trading Networks internal ID.
sender	**String** The sender to identify in the interchange envelope. For example, if you specify 01 for *senderQual* (indicating a D-U-N-S number), specify the value of the D-U-N-S number for *sender*.
receiverQual	**String** EDI ID qualifier for the receiver ID. It is used with *receiver* to obtain the Trading Networks internal ID.
receiver	**String** The receiver to identify in the interchange envelope. For example, if you specify 01 for *receiverQual* (indicating a D-U-N-S number), specify the value of the D-U-N-S number for *sender*.
ctlVersion	**String** Version of the EDI standard used, with a `00` prefix, e.g., for version 4010, specify `004010`.
ctlNumber	**String** (optional) The interchange control number of the EDI document.

ctlFromTable

String (optional; this variable is used only when the *ctlNumber* variable is not specified) Whether you want the service to obtain the control number for the interchange from the EDIControlNumber table, which is an EDI Module-specific table in the Trading Networks database. Specify `true` or `false`.

Value of *ctlFromTable*	Meaning
true	Obtain the control number from the EDIControlNumber table.
false	Do *not* obtain the control number from the EDIControlNumber table. Use the value specified in the *ctlNumber* variable.

ctlNumberWleadingZero

String (ANSI X12 only)Whether to add leading zeros to the interchange control number to make it a nine-digit number. Specify `true` or `false`.

Value of *ctlNumberWleadingZero*	Meaning
false	Do *not* add leading zeros to the interchange control number. This is the default.
true	Add leading zeros to the interchange control number to make it a nine digit number, e.g., `12` becomes `000000012`.

ackRequested

String (optional) Whether you want to request an acknowledgement for this interchange. Specify `0` or `1`.

Value of ackRequested	Meaning
0	Do not request an acknowledgement for this interchange. This is the default.
1	Requests an acknowledgement for this interchange.

testIndicator

String (optional) Whether to indicate production or test mode.

Value of testIndicator	Meaning
P	Production. This is the default.
T	Test.

repSeparator

String (optional) A separator for the repeated occurrences of a simple data element or a composite data structure. Length: 1.

Note: The *repSeparator* must be a different character than the *record*, *field*, or *subfield* delimiters.

delimiters Document (optional) Delimiters used in the outbound EDI document.

Variable in *delimiters*	Description
record	String The segment terminator for the EDI document, e.g., `\u000a` to use the new line character. The default is the new line character.
field	String The field separator for each EDI segment e.g., `!`. The default is the * character.
subfield	String The separator for composite elements, e.g., ^. The default is the : character.

addGroup String (optional) Whether you want the addICEnvelope service to add group envelopes in addition to the interchange envelope. Specify `true` or `false`.

Note: You must add a group envelope *either* by setting this variable to `true`, or setting this variable to `false` and using the addGroupEnvelope service prior to using the addICEnvelope service. Failing to add a group envelope or creating a group envelope twice will result in an invalid document.

Value of addGroup	Meaning
true	Add group envelopes (GS and GE segments) to the document *prior* to adding the interchange envelope. Note that ANSI X12 documents *must* have a group segment. Set this variable to `true` if you did *not* use the addGroupEnvelope service and you do *not* need maximum control over group envelope variables.
false	Do not add group envelopes prior to creating the Interchange envelope. Set this variable to `false` if you would prefer to use the addGroupEnvelope service prior to using this service. The addGroupEnvelope service provides more control over the variables in the group envelopes.

groupInfo Document (optional) Information about the group segments added by *addGroup*.

Variable in *groupInfo*	Description
IDcode	String The Functional ID Code of the EDI document according to the EDI standard.
senderQual	String EDI ID qualifier for the sender ID. It is used with *sender* to obtain the Trading Networks internal ID. Note: When you do not specify values for the group-level *senderQual*, *sender*, *receiverQual*, and *receiver* fields, the service uses the values specified in the interchange-level *senderQual*, *sender*, *receiverQual*, and *receiver* fields.
sender	String The sender to identify in the group envelope. For example, if you specify 01 for *senderQual* (indicating a D-U-N-S number), specify the value of the D-U-N-S number for *sender*.
receiver	String The receiver to identify in the group envelope. For example, if you specify 01 for *receiverQual* (indicating a D-U-N-S number), specify the value of the D-U-N-S number for *receiver*.
receiverQual	String EDI ID qualifier for the receiver ID. It is used with *receiver* to obtain the Trading Networks internal ID.
ctlFromTable	String (optional) Whether you want the service to obtain the control number for the group from the EDIControlNumber table, which is an EDI Module-specific table in the Trading Networks database. Specify `true` or `false`. Value of *ctlFromTable* — Meaning true — Obtain the control number from the EDIControlNumber table. false — Do *not* obtain the control number from the EDIControlNumber table. Use the value specified in the *grpCtlNumber* variable.
grpCtl Number	String (optional) The group control number of the EDI document.

agencyCode	String (optional) Responsible agency code: `T` (default) or `X`.
verRelCode	String EDI standard version and release code, e.g., `4010`, `3040`, etc.

Output Parameters

outDocument	String Contains the outbound EDI string. This variable can be validated by providing it as input to the wm.b2b.edi:envelopeProcess service.

Usage Notes

Only use this service for ANSI X12, UCS, or VICS standards.

Example

For an example of processing an outbound EDI document, see the Tutorial.XMLtoEDI:processXMLSource service in the WmEDIsamples package, which is located in the Knowledge Base on the webMethods Advantage Web site at http://advantage.webmethods.com.

wm.b2b.edi.util:addICEnvelopeEDIFACT

For outbound EDI documents, adds an IC envelope (UNB and UNZ) according to the UN/EDIFACT standard.

Input Parameters

documents	String List EDI documents to which to add an IC envelope.
syntaxId	String Syntax identifier.
syntaxVersion	String Syntax version number.
senderId	String The sender ID of the interchange.
senderQual	String (optional) The sender EDI ID qualifier code.
reverseRoute	String (optional) The sender internal identification.
receiverId	String (optional) The receiver ID of the interchange.
receiverQual	String (optional) The receiver EDI ID qualifier code.
routingAddress	String (optional) The receiver internal identification.
ctlFromTable	String (optional) Whether you want the service to obtain the control number for the interchange from the EDIControlNumber table, which is an EDI Module-specific table in the Trading Networks database. Specify `true` or `false`.

Value of *ctlFromTable*	Meaning
true	Obtain the control number from the EDIControlNumber table.
false	Do *not* obtain the control number from the EDIControlNumber table. Use the value specified in the *ICreference* variable.

ICreference	String (optional) The IC control number of the EDI document.
password	String (optional) The recipient reference password.
passwordQual	String (optional) The recipient reference password qualifier.
applReference	String (optional) The application reference.
priority	String (optional) The processing priority code.

ackRequested String (optional) Whether you want to request an acknowledgement for this interchange. Specify `0` or `1`.

Value of *ackRequested*	Meaning
0	Do not request an acknowledgement for this interchange. This is the default.
1	Request an acknowledgement for this interchange.

agreementId String (optional) The interchange agreement identifier.

testIndicator String (optional) Whether to indicate production or test mode.

Value of *testIndicator*	Meaning
P	Production. This is the default.
T	Test.

UNARequired String Whether you want the service to create a UNA segment in front of the resulting output. Specify `true` or `false`.

Value of *UNARequired*	Meaning
true	Create the UNA segment.
false	Do not create the UNA segment.

delimiters Document (optional) Delimiters used in the outbound EDI document.

Variable in *delimiters*	Description
record	String The segment terminator for the EDI document, e.g., `\u000a` to use the new line character. The default is the new line character.
field	String The field separator for each EDI segment e.g., `!`. The default is the * character.
subfield	String The separator for composite elements, e.g., ^. The default is the `:` character.
release	String The release character for composite elements, e.g., ^. The default is the `?` character.
decimal	String The decimal character for composite elements, e.g., ^. If *UNARequired* is `true`, the decimal delimiter is used. The default is the `.` character. You may enter only one character. If *UNARequired* is `false`, *decimal* is ignored.

addGroup String (optional) Whether you want the service to add group header and trailer (UNG and UNE) segments in the interchange. Specify `true` or `false`.

Note: You can add a group envelope *either* by setting this variable to `true`, or setting this variable to `false` and using the addGroupEnvelope service prior to using the addICEnvelopeEDIFACT service. Creating a group envelope twice will result in an invalid document.

Value of *addGroup*	Meaning
true	Add the UNG and UNE segments to the document *prior* to adding the interchange envelope.
false	Do not add the UNG and UNE segments.

groupInfo Document (optional) Information about the group segments added by *addGroup*.

Variable in *groupInfo*	Description
IDcode	String (optional) The Functional ID Code of the EDI document according to the EDI standard.
syntax Version	String Syntax version of the envelope level.
senderID	String (optional when the value of *syntaxVersion* is greater than 3) The sender to identify in the group envelope. For example, if you specify 01 for *senderQual* (indicating a D-U-N-S number), specify the value of the D-U-N-S number for *sender*. Note: When the value of *syntaxVersion* is 1 and you do not specify values for the group-level *senderID*, *senderQual*, *receiverID*, and *receiverQual* fields, the service uses the values specified in the interchange-level *senderID*, *senderQual*, *receiverID*, and *receiverQual* fields. When the value of *syntaxVersion* is 4 and you do not specify values for the group-level *senderID*, *senderQual*, *receiverID*, and *receiverQual* fields, the service leaves these group-level fields blank.
senderQual	String (optional) EDI ID qualifier for the sender ID.

receiverID	String (optional when the value of *syntaxVersion* is greater than 3) The receiver to identify in the group envelope. For example, if you specify 01 for *receiverQual* (indicating a D-U-N-S number), specify the value of the D-U-N-S number for *receiver*.
receiverQual	String (optional) EDI ID qualifier for the receiver ID.
grpCtl Number	String (optional) The group control number of the EDI document.
ctlFromTable	String (optional; this variable is used only when the *grpCtlNumber* variable is not specified) Whether you want the service to obtain the control number for the group from the EDIControlNumber table, which is an EDI Module-specific table in the Trading Networks database. Specify `true` to obtain the control number from the EDIControlNumber table, or `false` to use a random generated number as the value.
prodMode	String (optional) The production mode associated with control number: `Production` (the default) or `Test`.
agencyCode	String (optional) Responsible agency code: `UN` (default) or `AA`.
verCode	String (optional) EDI standard version code, e.g., `D`, `S`, etc.
relCode	String (optional) EDI message standard release code, e.g., `96A`, `97B`, etc.
aACode	String (optional) EDI message standard assigned code, e.g., `OD`, `EN`, etc.
password	String (optional) The recipient transmission reference password.

Output Parameters

outDocument	String Contains the outbound EDI string. This variable can be validated by providing it as input to the wm.b2b.edi:envelopeProcess service.

Usage Notes

Only use this service for the UN/EDIFACT standard.

wm.b2b.edi.util:concatStringArray

Creates a String from the input parameters.

The String starts with the value of the input variable *prepend*, followed the value of each element in the input variable *stringArray*, and end with the value of the input variable *append*. For example, if the input values were:

- *stringArray:*
  - String1
  - String2
- *prepend*: String to prepend
- *append*: String to append

The final string would be:

```
String to prependString1String2String to append
```

Input Parameters

stringArray	String List An array of Strings to be concatenated.
prepend	String String value to prepend on to the final result.
append	String String value to append on to the final result.

Output Parameters

concated	String The concatenated String.

wm.b2b.edi.util:controlNumber

Creates a control number.

The control number is a nine-digit number, based on a two-digit day, two-digit hours, two-digit minutes, and a random three-digit system count (100-999).

Input Parameters

None

Output Parameters

ctlNumber	String (optional) description.

wm.b2b.edi.util:convertToValues

This is a utility service that maintains compatibility between EDI 4.6 and 6.*x*.

Usage Notes

Never invoke this service directly.

wm.b2b.edi.util:EDIconcat

Return an InputStream pointer from either the “unDefData” or the “_RID_” elements.

These elements are part of the output of an EDI object that has gone through the wm.b2b.edi:convertToValues service.

Input Parameters

header	String (optional) String data to which the “unDefData” or the “_RID_” data will be appended.
trailer	String (optional) String data to append at the end of the result.
unDefData	String (optional) Input data with type String.
reservation	Object (optional) ID used to retrieve the data from temporary storage (i.e., tspace).

Output Parameters

output	String or InputStream Resulting data.

Usage Notes

If you are working with non-EDI flat files, you do not use this service for processing. Instead, see the *Flat File Schema Developer's Guide*.

For more information about how to use this service, see the section about processing EDI documents segment by segment in Chapter 3, "Receiving and Processing Inound EDI Documents" of the *webMethods EDI Module User's Guide*.

This service is analogous to the wm.b2b.edi.util:getEDIstring service for traditional EDI documents, except that the getEDIstring service creates a complete EDI document from the transaction set header (e.g., ST/SE) and the “unDefData” or the “_RID_” elements representing the document content. Instead of looking for a standard transaction set header to construct the document, the EDIconcat service looks only for an “unDefData” or the “_RID_” element. To construct a full document, use the header and trailer variables to insert the converted “unDefData” or the “_RID_” segment into the document's larger context.

Example

The sampleServices:Iterator810 service uses the EDIconcat service during processing. This service is included in the WmEDIsamples package, which is located in the Knowledge Base on the webMethods Advantage Web site at http://advantage.webmethods.com.

wm.b2b.edi.util:generateFA

Creates a functional acknowledgement (FA) that acknowledges all levels of an input EDI document.

This service takes an EDI document as input, performs validation and compliance check, and creates an FA as output.

Input Parameters

edidata — String or InputStream (optional) An unparsed EDI document for which you want to generate an FA. Specify a value for either *edidata* or *ICValues*.

ICValues — Document (optional) A parsed EDI document (an IData object) for which you want to generate an FA. This must include the error results. This is typically the output from the wm.b2b.edi:envelopeProcess service. Specify a value for either *edidata* or *ICValues*.

EDI_delimiters — Document (optional) Delimiters used in the outbound FA. If you do not specify *EDI_delimiters*, the service uses the delimiters from the document it is acknowledging.

Variable in *EDI_delimiters*	Description
record	String The segment terminator for the FA, e.g., +. The default is the ' character.
field	String The field separator for each EDI segment e.g., !. The default is the + character.
subfield	String The separator for composite elements, e.g., ^. The default is the : character.
release	String Character used to ignore a record, field, or subfield delimiter in a field. If a release character occurs in a field or subfield before the delimiter, it will be prefixed with this character in the output.

FARequest

String How to handle potentially conflicting information in the EDI interchange header of the inbound document. For ANSI X12, the inbound document specification is in ISA14. For UN/EDIFACT, the inbound document specification is in UNB09. Specify one of the following:

Value of *FArequest*	Meaning
`yes`	The FA is generated regardless of FA specification in the input EDI interchange header.
`no`	If `no` *and* the input EDI interchange header specifies that an FA is not required, the FA is not generated.
`depend_on_input`	The FA is generated based on the FA specification in the input EDI interchange header.

FALevel

String The level at which to acknowledge. If any errors occur at the level you select, the FA will list those errors at the level selected. Specify one of the following:

Value of *FALevel*	Description
`Default`	Acknowledge at the envelope level (group for ANSI X12 and interchange for UN/EDIFACT).
`TransactionSet`	Acknowledge at the transaction set level (ANSI X12 only).
`Segment`	Acknowledge at the segment level (ANSI X12 only).
`Element`	Acknowledge at the element level (ANSI X12 only).

generateControlNumber

String How you want the service to obtain the control number it uses in the interchange and group headers of the generated FA. Specify one of the following:

Value of *generateControlNumber*	Meaning
`FromInboundDocument`	Use the control number from the corresponding header in the EDI document that is being acknowledged. For example, if generating an FA for a group in an ANSI X12 document, use the control number from the group header of that group for the control number of the FA.
`Random`	The service randomly generates a control number.

`FromControlNumberTable`	Obtain the next control number from the EDIControlNumber table, and then increment the value of the next control number in the table entry, so it reflects the new next control number. **Note:** You can only specify `FromControlNumberTable` if you are using the EDI Module with Trading Networks.

addGroups

String Whether you want to add group segments to the ANSI X12 or UN/EDIFACT FA (e.g., a 997 or a UN/EDIFACT CONTRL). Specify `true` or `false`.

Value of *addGroups*	Meaning
true	Add a group to the FA.
false	Do not add a group to the FA.

addICEnvelopes

String Whether you want to add an interchange envelope to the FA. Specify `true` or `false`.

Value of *addICEnvelopes*	Meaning
true	Add an interchange to the FA. This is the default. For ANSI 12, if you add an envelope, you will automatically get the group.
false	Do not add an interchange to the FA.

syntaxErrorStatus

String How you want the service to report the syntax error status for a transaction, group, or UN/EDIFACT interchange. The syntax error status indicates whether there are syntax errors, for example, missing mandatory elements, violation of syntax rules, invalid field lengths, code list violations, or segment repeat counts exceeded.

The service uses the syntax error status along with the logical error status and child transaction rejected status (if applicable) to determine the FA status for a transaction, group, or UN/EDIFACT interchange.

Specify one of the following to indicate how you want the service to report syntax errors:

Value of *syntaxErrorStatus*	Meaning
`Rejected`	The syntax error status is reported as "Rejected" if syntax errors are encountered. Specify `Rejected` if you want to reject elements that have syntax errors.

`Accepted, But Errors Were Noted`	The syntax error status is reported as "Accepted, But Errors Were Noted" if syntax errors are encountered. Specify `Accepted, But Errors Were Noted` if you want to know whether there are syntax errors, but do not want to reject an element because of them.
`Accepted`	The syntax error status is *always* reported as "Accepted" regardless of any syntax errors that might be encountered. Specify `Accepted` if you do not want to check for syntax errors.

logicalErrorStatus String How you want the service to report the logical error status for a transaction, group, or UN/EDIFACT interchange. The logical error status indicates whether there are logical errors, for example:

- The control number in a header does not match the control number in the corresponding trailer, or
- The segment count in a trailer does not have an accurate group, transaction, or segment count.

The service uses the logical error status along with the syntax error status and child transaction rejected status (if applicable) to determine the FA status for a transaction, group, or UN/EDIFACT interchange.

Specify one of the following to indicate how you want the service to report logical errors:

Value of *logicalErrorStatus*	Meaning
`Rejected`	The logical error status is reported as "Rejected" if logical errors are encountered. Specify `Rejected` if you want to reject elements that have logical errors.
`Accepted, But Errors Were Noted`	The logical error status is reported as "Accepted, But Errors Were Noted" if logical errors are encountered. Specify `Accepted, But Errors Were Noted` if you want to know whether there are logical errors, but do not want to reject an element because of them.
`Accepted`	The logical error status is *always* reported as "Accepted" regardless of any logical errors that might be encountered. Specify `Accepted` if you do not want to check for logical errors.

childTransactionRejectedStatus

String How you want the service to report the child transaction rejected status for a group or UN/EDIFACT interchange. The child transaction rejected status indicates whether child elements of a group or UN/EDIFACT interchange have an FA status of "Rejected". Specify one of the following:

Value of *childTransactionRejectedStatus*	Meaning
`Rejected`	Reports the child transaction rejected status as: ■ "Rejected" if the FA status of any of the child transactions is "Rejected". ■ "Accepted, But Errors Were Noted" if the FA statuses of all child transactions are "Accepted" and "Accepted, But Errors Were Noted". ■ "Accepted" if the FA statuses of all the child transactions are "Accepted".
`Partially Accepted`	Reports the child transaction rejected status as: ■ "Rejected" if the FA statuses of all of the child transactions are "Rejected". ■ "Partially Accepted" if the FA status of at least one child transaction is "Rejected", but the FA status of other child transactions are "Accepted" or "Accepted, But Errors Were Noted". ■ "Accepted" if the FA statuses of all the child transactions are "Accepted".
`Accepted, But Errors Were Noted`	Reports the child transaction rejected status as: ■ "Accepted, But Errors Were Noted" if the FA status any child transaction is "Rejected" or "Accepted, But Errors Were Noted". ■ "Accepted" if the FA statuses of all the child transactions are "Accepted".

standardVC

When you use the *ICValues* input variable, use this to specify the substandard of the EDI standard. You can specify one of the following for *standardVC*: EANCOM, UCS, UNEDIFACT, VICS, X12, and ODETTE.

encoding

String (optional) The encoding of the data passed in to *edidata.*

userFFSchema

String (optional) A flat file schema that overrides the predefined flat file schema that wm.b2b.edi:convertToValues uses.

ctlNumberWleadingZero String (optional for ANSI X12) Whether to add leading zeros to the interchange control numbers to make them nine-digit numbers.

Note: This parameter only works when the value of the *generateControlNumber* parameter is `FromControlNumberTable`.

Value of *ctlNumberWleadingZero*	Meaning
false	Do *not* add leading zeros to the control number. This is the default.
true	Add leading zeros to the control number to make it a nine digit number, e.g., `12` becomes `000000012`.

additional_Info Document (optional; for use with ODETTE only) Additional information about the input message, to be used in the outbound FA.

Variable in *additional_Info*	Description
`Code`	String A qualification and identification of the purpose and function of a text segment. Maximum length: 3 characters.
`Text`	String Text. Maximum length: 70 characters.

Output Parameters

outDocument String List The outbound FA.

Note: UN/EDIFACT and ODETTE CONTRLs both use the version 4 UN/EDIFACT CONTRL error codes.

Envelope Document List Summary of information about the interchanges, groups, and transactions from the input EDI document.

Important! This output variable is for EDI Module, internal use only. Do not code services that rely on the content and format of *Envelope* because it can change between releases of the EDI Module.

Usage Notes

- The service does not specify what to do with the acknowledgement that it has created.
- This service can acknowledge all levels of an EDI document.
- You can use the *EDIResolveDuplicates* property to control how the EDI Module assigns FA status when you send (or receive) a document multiple times before the receiver returns an FA. For details, see Chapter 21, "Reconciling Functional Acknowledgements", in the *webMethods EDI Module User's Guide*.

wm.b2b.edi.util.FA:lite997

Creates a functional acknowledgement (997) that acknowledges EDI documents based on input values.

This service takes *AK9* and *ICValues* IS document (IData object) as input, which is parsed and validated, and creates a 997 as output on the functional group level.

Input Parameters

ICValues — Document Parsed EDI values of envelope, including error results. This IData object contains only one envelope and one functional group.

EDI_delimiters — Document (optional) Delimiters used in the outbound FA.

Variable in *delimiters*	Description
record	String The segment terminator for the FA, e.g., +. The default is the ' character.
field	String The field separator for each EDI segment e.g., !. The default is the + character.
subfield	String The separator for composite elements, e.g., ^. The default is the : character.
release	String Character used to ignore a record, field, or subfield delimiter in a field. If a release character occurs in a field or subfield before the delimiter, it will be prefixed with this character in the output.

AK9 — Document The values of the AK9 segment. These values acknowledge the functional group listed from ICValues.

senderIDQualifier — String (optional) The EDI ID qualifier of functional group level sender.

receiverIDQualifier — String (optional) The EDI ID qualifier of functional group level receiver.

grpCtlNumber — String (optional) The group control number of the output 997 document.

ctlFromTable String (optional) Whether you want the service to obtain the control number from the EDITPA (EDI Trading Partner Agreement). Specify `true` or `false`.

Value of *ctlFromTable*	Meaning
`true`	Obtain the control number from the EDITPA.
`false`	Do not obtain the control number from the EDITPA. Use the value specified in the grpCtlNumber variable.

addGroups String Whether to add a functional group to the FA.

Value of addGroups	Meaning
`true`	Add a functional group to the FA.
`false`	Do not add a functional group to the FA.

addICEnvelopes String Whether to add an interchange envelope to the FA. Specify `true` or `false`.

Value of addICEnvelopes	Meaning
`true`	Add an interchange to the FA.
`false`	Do not add an interchange to the FA.

Output Parameters

outDocument String The outbound FA string.

Usage Notes

The service does not specify what to do with the acknowledgement that it has created. This service can be used only for the ANSI X12 standards.

wm.b2b.edi.util:getEDIDictionaryName

Given an EDI standard and version, returns the name of the flat file dictionary that should be used to store the definitions for that standard and version.

Input Parameters

standard String EDI standard name, e.g., `X12` or `UNEDIFACT`.

version String EDI version, e.g., `4010`, `96A`, etc.

Output Parameters

ffDictionaryName String Standard name of the flat file dictionary that should be used to store entries for this standard/version combination.

Usage Notes

The *standard* and *version* follow this naming convention:

`EDIFFSchema.`*`standard`*`.V`*`version`*`:T`*`transaction`*, where:

- *`standard`* represents the EDI standard (e.g., `X12`).
- *`version`* represents the EDI standard version (e.g., `4010`).
- *`transaction`* represents the EDI transaction (e.g., `850`).

For example: `EDIFFSchema.X12.V4010:T850`

wm.b2b.edi.util:getEDIFFSchemaName

Given an EDI standard, version, and transaction set name (or TRADACOMS file name), returns the name of the flat file schema that should be used.

Input Parameters

standard String EDI standard name, e.g., `X12` or `UNEDIFACT` or `TRADACOMS`.

version String EDI version (e.g., `4010`, `96A`, etc.) or TRADACOMS file version (e.g., `v2`).

name String Name of the EDI transaction set (e.g., `810`, `ORDRS`, etc.) or TRADACOMS file (`ACKMNT`).

message String (required for use with TRADACOMS files) The header name of the selected TN document type File. For example, if you selected the TN document type `ACKMNT`, you would provide the value `ACKHDR`.

Output Parameters

ffSchemaName String Standard name of the flat file schema that should be used for this EDI transaction set or TRADACOMS file.

Usage Notes

For ANSI X12 and UN/EDIFACT, the *standard*, *version*, and *name* follow this naming convention:

`EDIFFSchema.`*`standard`*`.V`*`version`*`:T`*`name`*, where:

- *`standard`* represents the EDI standard (e.g., `X12`).
- *`version`* represents the EDI standard version (e.g., `4010`).

- *name* represents the EDI transaction (e.g., 850).

For example: EDIFFSchema.X12.V4010:T850

For TRADACOMS, the *version*, *name*, and *message* follow this naming convention:

EDIFFSchema.Tradacoms.V*version*.T*name*:M*message*, where:

- *version* represents the TRADACOMS version (for example, v2)
- *name* represents the name of the TRADACOMS File document type (for example, TLPRHDR)
- *message* is derived from the MHD0201.

For example: EDIFFSchema.Tradacoms.v2.TLPRHDR:MLPRHDR

wm.b2b.edi.util:getEDIstring

Creates a complete EDI transaction set (either String or InputStream type) from an EDI transaction set header segment.

When processing traditional inbound EDI documents, this service should be used after the wm.b2b.edi:envelopeProcess service and before the wm.b2b.edi:convertToValues service.

Input Parameters

Values	Document The single EDI transaction set within a single interchange and a single functional group. (This input object is part of the output of the wm.b2b.edi:envelopeProcess service.)

Output Parameters

EDIstring	Object A single EDI transaction set in String or InputStream type.

Usage Notes

The getEDIstring service should not be invoked within a flow service until you have LOOPED over the interchange and group segments and arrived at the document-level segments.

If you are working with non-EDI flat files, you do not use this service for processing. Instead, see the *Flat File Schema Developer's Guide*.

wm.b2b.edi.util:invoke

Executes the service identified by the input variables.

The service you want to invoke must be defined in the file *webMethods6**IntegrationServer*\packages\WmEDI\config\services.cnf.

Input Parameters

interface	String The name of the folder and subfolders in which the service you want to execute is located. For example, the `wm.b2b.edi.util`. You can specify the full path for a service in the format of `folder.subfolder:service`, for example, `wm.b2b.edi.util:ctlNumber`. If you use this format, do not specify *service*.
service	String (optional) The name of the service that you want to execute, for example, `ctlNumber`. If you specified the full path for the service in *interface*, do not specify anything for *service*.
input	Document Information you want passed to the service. This data will be the data in the pipeline when the service is invoked.

Output Parameters

output	Document Information that the invoked service returns.

wm.b2b.edi.util:makeArray

Creates an array out of the input object.

For example, if the input object is an IData object, the output will be a one dimensional IData object array. This is useful for a service that expects array input. If the input object is an array, the output object also will be an array of the same type as the input object.

Input Parameters

input	Object The input object.

Output Parameters

type	String Type of the output array object, e.g., `String` or `Data`.
array	Object List Output array object of the same type as the input object.

wm.b2b.edi.util:nullBlankCheck

Determines whether a specified String is null or contains only blank spaces.

Input Parameters

input **String** String data that you want to check for null or blanks.

Output Parameters

isNullorBlank **String** Whether the input String is null or contains only blank spaces.

Value of isNullorBlank	Meaning
`true`	The input is null or contains only blank spaces.
`false`	The input is *not* null or contains non-blank characters.

wm.b2b.edi.util:nullifyIfBlank

Returns null if the input is null or contains only blank spaces.

Input Parameters

input **String** String data.

Output Parameters

output **String** Same data as *input* or null.

wm.b2b.edi.util:pad

Adds blank spaces to a field, left- or right-justifying the field as specified.

Also truncates the data field if the data is longer than the length specified. Leading and trailing spaces are deleted prior to padding spaces.

Input Parameters

input **String** String data.

length **String** The total length that you want the resulting field to be.

from String (optional) Whether you want the String left or right justified.

Value of from	Meaning
`left`	Align the field's value to the left, adding blank spaces to the right of the field value. If the length specified is shorter than the actual input data, the input data string is truncated from the right.
`right`	Align the field's value to the right, adding blank spaces to the left of the field value. If the length specified is shorter than the actual input data, the input data string is truncated from the left.

Output Parameters

result String The resulting String data.

wm.b2b.edi.util:standardCheck

Checks the input String or InputStream starting with an array of string and outputs the result based on the array of string.

Input Parameters

edidata String or InputStream Input data.

lookFor Document (optional) A list of name/value pairs. The name will be compared with the starting character string of *edidata*. If a match is found, the value related to the name is returned as output.

By default, it contains:

Name	Value
ISA	X12
UNA	UNEDIFACT
UNB	UNEDIFACT
EDI_DC	IDOC

Output Parameters

edidata String or InputStream Reflects the input data.

standard String Returns the value portion of the name/value pair if the name matched the input String. Returns `null` if no match was found.

subStandard	String Sub-standard under the standard above. For example, UCS and VICS are sub-standards under X12, and ODETTE and EANCOM are sub-standards under UN/EDIFACT.
version	String Return the version of the EDI document. No value for non-EDI documents.

wm.b2b.edi.util.formatServices

Services in this folder ensure that the fields of an EDI document conform to the ANSI X12 or UN/EDIFACT standard formats for various data types.

- The wm.b2b.edi:convertToValues service invokes format services when parsing a String to an IData object.
- The wm.b2b.edi:convertToString service invokes format services when creating a String from an IData object.

The format services transform field values from an "internal" format to an "external" format, or vice versa. The external format is the format that will appear in the actual EDI document. The internal format is the format that the field meets in the parsed form (after calls to the convertToValues service or before calls to the convertToString service). The internal and external formats are stored in an external configuration file. For more information about this file, see Chapter 1, "Before You Can Process EDI Documents" in the *webMethods EDI Module User's Guide*. The external formats provided match the EDI ANSI standard formats and should not be changed. The internal formats should be modified to conform to your internal application formats.

Important! If a particular field does not have a value (that is, a value is not in the input String or IData object, the format service assigned to that field will not be executed.

Inputs and Outputs for all Format Services

All of the EDI format services have the same following inputs and outputs variables.

Input Parameters

value String The field value to format.

direction String Indicates the type of formatting to apply to the field. Specify one of the following for *direction*:

Value of *direction*	Meaning
`convertToString`	Apply external formatting.
`convertToValues`	Apply internal formatting.

validate

String The value of the input variable, *validate*, from the wm.b2b.edi:convertToValues service. This indicates whether the service should update the value to be validated for this field. The value is `true` or `false`.

Value of *validate*	Meaning
`true`	Set the *valueToValidate* output variable with the formatted value, so the formatted value is validated.
`false`	Value will not be validated. Note that *validate* is always `false` when *direction* is `convertToString`.

minLength

String The minimum length of the field. The following describes how this field is used.

- If the field is extracted via the Fixed Position Extractor, *minLength* is used to determine the minimum number of characters to extract.
- If the field is *not* extracted via the Fixed Position Extractor and is associated with a Length Validator, *minLength* is used to determine the minimum length to consider valid.
- Otherwise, *minLength* is *not* used and will *not* be present in the pipeline.

maxLength

String The maximum length of the field. The following describes how this field is used.

- If the field is extracted via the Fixed Position Extractor, *maxLength* is used to determine the maximum number of characters to extract.
- If the field is *not* extracted via the Fixed Position Extractor and is associated with a Length Validator, *maxLength* is used to determine the maximum length to consider valid.
- If the maximum length is unlimited (-1) or there is no Length Validate, *maxLength* is *not* used and will *not* be present in the pipeline.

FormatInfo Document Information that can be used by individual formatting services. This information is obtained from one of three locations:

- If the wm.b2b.edi:convertToString service is invoking the format service, this is the value of the input variable, *FormatInfo*, of the convertToString service.
- If the wm.b2b.edi:convertToValues service is invoking the format service, this is the value of the input variable, *delimiters/FormatInfo*, of the convertToValues service
- If converting fields for a UN/EDIFACT document, the EDI document type automatically extracts the decimal separator from the UNA segment.

The only format services that use this feature are the decimal formatting services (for implied decimal and decimal formats). The *FormatInfo* variable should contain a String name/value pair named *DecimalCharacter*. If *DecimalCharacter* is ',' the number would be formatted as 100,10 (European format) instead of 100.10, as is common in the US.

Note: Changes to the data in this object will be reflected in all other format services that are invoked during execution of the convertToString and convertToValues services.

Output Parameters

formattedValue String The field value with appropriate formatting applied.

meetsFormat String Whether the value could be formatted properly. It will be `true` or `false`.

Value of *meetsFormat*	Meaning
`true`	The value could be properly formatted.
`false`	The value could not be properly formatted.

errorMessage String If *meetsFormat* is `false`, this parameter provides a text message describing the formatting error.

valueToValidate String The value that will be used by the validator for this field. If this value is not present, the value passed in the input variable *value* will be validated. This field is used only when the input variable *validate* is set to `true`.

Important! All service descriptions assume that the configuration file has not been modified from its original settings.

wm.b2b.edi.util.formatServices:formatDate6

This service is used to format fields that have an EDI data type of "DT," with a minimum length of 6 and a maximum length of 6.

The format string for internal and external date type formats follow the conventions described in the java class java.text.SimpleDateFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/SimpleDateFormat.html).

wm.b2b.edi.util.formatServices:formatDate8

This service is used to format fields that have an EDI data type of "DT," with a minimum length of 8 and a maximum length of 8.

The format string for internal and external date type formats follow the conventions described in the java class java.text.SimpleDateFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/SimpleDateFormat.html).

wm.b2b.edi.util.formatServices:formatDecimal

This service is used to format fields that have an EDI data type of "D."

The format string for internal and external date type formats follow the conventions described in the java class java.text.DecimalFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html).

If the *FormatInfo* variable contains a String called *DecimalCharacter*, that character will be used as the decimal separator character in the formatted number. Examples of decimal separator character are '.' used in the US and ',' used in some European countries.

The value for *DecimalCharacter* is obtained in one of three ways:

- As specified in the *FormatInfo* variable of the wm.b2b.edi:convertToString, wm.b2b.edi:convertToValues, pub.flatFile:convertToString, or pub.flatFile:convertToValues service. This method overrides all others.
- Through the UNA segment in an UN/EDIFACT document. This will override the system default decimal separator character.
- The default decimal separator for the locale of the JVM in which the Integration Server is running.

wm.b2b.edi.util.formatServices:formatN0

This service is used to format fields that have an EDI data type of "N0."

The external format of implied decimal fields cannot be modified. The internal format string for implied decimal fields follows the conventions described in the java class java.text.DecimalFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html).

If the *FormatInfo* variable contains a string called *DecimalCharacter*, that character will be used as the decimal separator character in the formatted number. Examples of decimal separator character are '.' used in the US and ',' used in some European countries.

The value for *DecimalCharacter* is obtained in one of three ways:

- As specified in the *FormatInfo* variable of the wm.b2b.edi:convertToString, wm.b2b.edi:convertToValues, pub.flatFile:convertToString, or pub.flatFile:convertToValues service. This method overrides all others.
- Through the UNA segment in an UN/EDIFACT document. This will override the system default decimal separator character.
- The default decimal separator for the locale of the JVM in which the Integration Server is running.

wm.b2b.edi.util.formatServices:formatN1

This service is used to format fields that have an EDI data type of "N1."

The external format of implied decimal fields cannot be modified. The internal format string for implied decimal fields follows the conventions described in the java class java.text.DecimalFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html).

If the *FormatInfo* variable contains a string called *DecimalCharacter*, that character will be used as the decimal separator character in the formatted number. Examples of decimal separator character are '.' used in the US and ',' used in some European countries.

The value for *DecimalCharacter* is obtained in one of three ways:

- As specified in the *FormatInfo* variable of the wm.b2b.edi:convertToString, wm.b2b.edi:convertToValues, pub.flatFile:convertToString, or pub.flatFile:convertToValues service. This method overrides all others.
- Through the UNA segment in an UN/EDIFACT document. This will override the system default decimal separator character.
- The default decimal separator for the locale of the JVM in which the Integration Server is running.

wm.b2b.edi.util.formatServices:formatN2

This service is used to format fields that have an EDI data type of "N2."

The external format of implied decimal fields cannot be modified. The internal format string for implied decimal fields follows the conventions described in the java class java.text.DecimalFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html).

If the *FormatInfo* variable contains a string called *DecimalCharacter*, that character will be used as the decimal separator character in the formatted number. Examples of decimal separator character are '.' used in the US and ',' used in some European countries.

The value for *DecimalCharacter* is obtained in one of three ways:

- As specified in the *FormatInfo* variable of the wm.b2b.edi:convertToString, wm.b2b.edi:convertToValues, pub.flatFile:convertToString, or pub.flatFile:convertToValues service. This method overrides all others.
- Through the UNA segment in an UN/EDIFACT document. This will override the system default decimal separator character.
- The default decimal separator for the locale of the JVM in which the Integration Server is running.

wm.b2b.edi.util.formatServices:formatN3

This service is used to format fields that have an EDI data type of "N3."

The external format of implied decimal fields cannot be modified. The internal format string for implied decimal fields follows the conventions described in the java class java.text.DecimalFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html).

If the *FormatInfo* variable contains a string called *DecimalCharacter*, that character will be used as the decimal separator character in the formatted number. Examples of decimal separator character are '.' used in the US and ',' used in some European countries.

The value for *DecimalCharacter* is obtained in one of three ways:

- As specified in the *FormatInfo* variable of the wm.b2b.edi:convertToString, wm.b2b.edi:convertToValues, pub.flatFile:convertToString, or pub.flatFile:convertToValues service. This method overrides all others.
- Through the UNA segment in an UN/EDIFACT document. This will override the system default decimal separator character.
- The default decimal separator for the locale of the JVM in which the Integration Server is running.

wm.b2b.edi.util.formatServices:formatN4

This service is used to format fields that have an EDI data type of "N4."

The external format of implied decimal fields cannot be modified. The internal format string for implied decimal fields follows the conventions described in the java class java.text.DecimalFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html).

If the *FormatInfo* variable contains a string called *DecimalCharacter,* that character will be used as the decimal separator character in the formatted number. Examples of decimal separator character are '.' used in the US and ',' used in some European countries.

The value for *DecimalCharacter* is obtained in one of three ways:

- As specified in the *FormatInfo* variable of the wm.b2b.edi:convertToString, wm.b2b.edi:convertToValues, pub.flatFile:convertToString, or pub.flatFile:convertToValues service. This method overrides all others.
- Through the UNA segment in an UN/EDIFACT document. This will override the system default decimal separator character.
- The default decimal separator for the locale of the JVM in which the Integration Server is running.

wm.b2b.edi.util.formatServices:formatN5

This service is used to format fields that have an EDI data type of "N5."

The external format of implied decimal fields cannot be modified. The internal format string for implied decimal fields follows the conventions described in the java class java.text.DecimalFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html).

If the FormatInfo parameter contains a string called *DecimalCharacter* that character will be used as the decimal separator character in the formatted number. Examples of decimal separator character are '.' used in the US and ',' used in some European countries.

The value for *DecimalCharacter* is obtained in one of three ways:

- As specified in the *FormatInfo* variable of the wm.b2b.edi:convertToString, wm.b2b.edi:convertToValues, pub.flatFile:convertToString, or pub.flatFile:convertToValues service. This method overrides all others.
- Through the UNA segment in an UN/EDIFACT document. This will override the system default decimal separator character.
- The default decimal separator for the locale of the JVM in which the Integration Server is running.

wm.b2b.edi.util.formatServices:formatN6

This service is used to format fields that have an EDI data type of "N6."

The external format of implied decimal fields cannot be modified. The internal format string for implied decimal fields follows the conventions described in the java class java.text.DecimalFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html).

If the *FormatInfo* variable contains a string called *DecimalCharacter*, that character will be used as the decimal separator character in the formatted number. Examples of decimal separator character are '.' used in the US and ',' used in some European countries.

The value for *DecimalCharacter* is obtained in one of three ways:

- As specified in the *FormatInfo* variable of the wm.b2b.edi:convertToString, wm.b2b.edi:convertToValues, pub.flatFile:convertToString, or pub.flatFile:convertToValues service. This method overrides all others.
- Through the UNA segment in an UN/EDIFACT document. This will override the system default decimal separator character.
- The default decimal separator for the locale of the JVM in which the Integration Server is running.

wm.b2b.edi.util.formatServices:formatN7

This service is used to format fields that have an EDI data type of "N7."

The external format of implied decimal fields cannot be modified. The internal format string for implied decimal fields follows the conventions described in the java class java.text.DecimalFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html).

If the *FormatInfo* variable contains a string called *DecimalCharacter*, that character will be used as the decimal separator character in the formatted number. Examples of decimal separator character are '.' used in the US and ',' used in some European countries.

The value for *DecimalCharacter* is obtained in one of three ways:

- As specified in the *FormatInfo* variable of the wm.b2b.edi:convertToString, wm.b2b.edi:convertToValues, pub.flatFile:convertToString, or pub.flatFile:convertToValues service. This method overrides all others.
- Through the UNA segment in an UN/EDIFACT document. This will override the system default decimal separator character.
- The default decimal separator for the locale of the JVM in which the Integration Server is running.

wm.b2b.edi.util.formatServices:formatN8

This service is used to format fields that have an EDI data type of "N8."

The external format of implied decimal fields cannot be modified. The internal format string for implied decimal fields follows the conventions described in the java class java.text.DecimalFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html).

If the *FormatInfo* variable contains a string called *DecimalCharacter*, that character will be used as the decimal separator character in the formatted number. Examples of decimal separator character are '.' used in the US and ',' used in some European countries.

The value for *DecimalCharacter* is obtained in one of three ways:

- As specified in the *FormatInfo* variable of the wm.b2b.edi:convertToString, wm.b2b.edi:convertToValues, pub.flatFile:convertToString, or pub.flatFile:convertToValues service. This method overrides all others.
- Through the UNA segment in an UN/EDIFACT document. This will override the system default decimal separator character.
- The default decimal separator for the locale of the JVM in which the Integration Server is running.

wm.b2b.edi.util.formatServices:formatN9

This service is used to format fields that have an EDI data type of "N9."

The external format of implied decimal fields cannot be modified. The internal format string for implied decimal fields follows the conventions described in the java class java.text.DecimalFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html).

If the *FormatInfo* variable contains a string called *DecimalCharacter*, that character will be used as the decimal separator character in the formatted number. Examples of decimal separator character are '.' used in the US and ',' used in some European countries.

The value for *DecimalCharacter* is obtained in one of three ways:

- As specified in the *FormatInfo* variable of the wm.b2b.edi:convertToString, wm.b2b.edi:convertToValues, pub.flatFile:convertToString, or pub.flatFile:convertToValues service. This method overrides all others.
- Through the UNA segment in an UN/EDIFACT document. This will override the system default decimal separator character.
- The default decimal separator for the locale of the JVM in which the Integration Server is running.

wm.b2b.edi.util.formatServices:formatTime4_4

This service is used to format fields that have an EDI data type of "TM," with a minimum length of 4 and a maximum length of 4.

The format string for internal and external date type formats follow the conventions described in the java class java.text.SimpleDateFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/SimpleDateFormat.html).The only symbols that are supported are 'H', 'm', 's', and 'S'. Any other constructs from the SimpleDateFormat class ('G, 'y', 'M', 'E', 'D', 'F', 'w', 'W', 'a', 'K', 'k', 'z' and ''') are not supported.

wm.b2b.edi.util.formatServices:formatTime4_6

This service is used to format fields that have an EDI data type of "TM," with a minimum length of 4 and a maximum length of 6.

The format string for internal and external date type formats follow the conventions described in the java class java.text.SimpleDateFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/SimpleDateFormat.html). The only symbols that are supported are 'H', 'm', 's', and 'S'. Any other constructs from the SimpleDateFormat class ('G, 'y', 'M', 'E', 'D', 'F', 'w', 'W', 'a', 'K', 'k', 'z' and ''') are not supported.

wm.b2b.edi.util.formatServices:formatTime4_8

This service is used to format fields that have an EDI data type of "TM," with a minimum length of 4 and a maximum length of 8.

The format string for internal and external date type formats follow the conventions described in the java class java.text.SimpleDateFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/SimpleDateFormat.html). The only symbols that are supported are 'H', 'm', 's', and 'S'. Any other constructs from the SimpleDateFormat class ('G, 'y', 'M', 'E', 'D', 'F', 'w', 'W', 'a', 'K', 'k', 'z' and ''') are not supported.

wm.b2b.edi.util.formatServices:formatTime6_6

This service is used to format fields that have an EDI data type of "TM," with a minimum length of 6 and a maximum length of 6.

The format string for internal and external date type formats follow the conventions described in the java class java.text.SimpleDateFormat (http://java.sun.com/products/jdk/1.2/docs/api/java/text/SimpleDateFormat.html). The only symbols that are supported are 'H', 'm', 's', and 'S'. Any other constructs from the SimpleDateFormat class ('G, 'y', 'M', 'E', 'D', 'F', 'w', 'W', 'a', 'K', 'k', 'z' and ''') are not supported.

webMethods

CHAPTER 2

WmEDIforTN Package

wm.b2b.editn

Use the services in this folder when processing EDI documents through webMethods Trading Networks (Trading Networks).

wm.b2b.editn:addAttributeTypeToBizDoc

Associates the custom document attribute you have created with a TN document type for an EDI document that you have installed.

Input Parameters

BizDocName String Name of the TN document type with which you want to associate the custom document attribute.

BizDocAttributeName String Name of the custom document attribute you want to associate with the TN document type.

Required String Whether the document attribute is required.

Value of *Required*	Meaning
true	The document attribute is required.
false	The document attribute is *not* required.

Output Parameters

None

Usage Notes

- For more information about using this service, see Chapter 13, "Processing Inbound ANSI X12 and UN/EDIFACT Documents Using Trading Networks" in the *webMethods EDI Module User's Guide*.
- For more information about TN document types and document attributes, including the difference between required and not required document attributes, see the *webMethods Trading Networks User's Guide*.

wm.b2b.editn:bizdocToRecord

Returns an IData object that represents the EDI document based on the input BizDocEnvelope.

Input Parameters

bizDoc	Document The BizDocEnvelope for the EDI document.
encoding	String (optional) Used to convert bytes to a String. If it is not specified, the service uses the EDIencoding property in the *webMethods6**IntegrationServer*\packages\WmEDI\config\properties.cnf file. If the encoding property EDIencoding is not specified, encoding UTF-8 is used.

Output Parameters

boundNode	Document The IData object that represents the EDI document.
standard	String The name of the EDI standard of the EDI document, e.g., X12, UNEDIFACT.

Note: The service uses the values for *standard*, *version*, and *transactionName* to locate the flat file schema to use. This service uses the following flat file schema naming convention.

EDIFFSchema.*standard*.V*version*:T*transaction*, where

- *standard* represents the EDI standard (e.g., X12).
- *version* represents the EDI standard version (e.g., 4010).
- *transaction* represents the EDI transaction (e.g., 850).

For example: EDIFFSchema.X12.V4010:T850

version	String The version of the transaction set's standard that the EDI document uses, e.g., 4010 for ANSI X12, or 98A for UN/EDIFACT.
transactionName	String The name of the transaction associated with the flat file schema to be copied, e.g., 850.
errors	Document List EDI document validation errors that is the output from wm.b2b.edi:convertToValues.

wm.b2b.editn:getTspace

Retrieves the content of an EDI document from a BizDocEnvelope.

Input Parameters

bizdoc	Object The BizDocEnvelope from which to retrieve the content for an EDI document.

Output Parameters

edidata	Object The content of the EDI document.

Usage Notes

For more information about using this service, see Chapter 13, "Processing Inbound ANSI X12 and UN/EDIFACT Documents Using Trading Networks" in the *webMethods EDI Module User's Guide*.

wm.b2b.editn:trackEDIdocs

Scans the input EDI data to add the interchange/group to the EDITRACKING table, which is an EDI Module-specific table in the Trading Networks database.

This service is used for FA Reconciliation reporting.

Input Parameters

edidata	String Input EDI document.

Output Parameters

None

Usage Notes

- For more information about using the trackEDIdocs service, see Chapter 16, "Forming EDI Documents to Send Outbound When Using Trading Networks" in the *webMethods EDI Module User's Guide*.
- For more information about FA reconciliation, see Chapter 19, "Reconciling Functional Acknowledgements" in the *webMethods EDI Module User's Guide*.

wm.b2b.editn:validateEnvelope

Validates the EDI envelope and adds entries to the Trading Networks activity log for the validation errors encountered in the EDI envelope.

Input Parameters

bizdoc	Document The BizDocEnvelope that represents the EDI document.

Output Parameters

errors	Document List Errors from the EDI envelope validation and compliance check, which are inserted into the Trading Networks activity log.
errorCount	String Number of error entries.

wm.b2b.editn:validateTransaction

Validates the EDI document and adds entries to the Trading Networks activity log for the validation errors encountered in the EDI document.

Input Parameters

bizdoc	Document The BizDocEnvelope that represents the EDI document.

Output Parameters

errors	Document List Errors from the EDI document validation, which are inserted into the Trading Networks activity log.
errorCount	String Number of error entries.

wm.b2b.editn:wrapData

Inserts a character string at a fixed length position to form fixed length data. This service is used to create wrap data to send to a mainframe system.

Input Parameters

input String Input source data.

wrapSize String Fixed-length position at which to insert the character string.

wrapValue String Character string to insert.

fillCharacter String Character used to fill empty positions in data that does not match the desired fixed length.

style String How the character string is inserted. Specify one of the following:

Value of *style*	Meaning
`fully wrap`	Count to the fixed-length position, and then insert the character string.
`semi wrap`	Each carriage return/line feed will reset the length. The *wrapValue* character string will be inserted only when the record is longer than the *wrapSize*.

Output Parameters

output String The output data with the wrap value inserted.

wm.b2b.editn.batch

The services in this folder are used with the batching feature of the EDI Module.

wm.b2b.editn.batch:batchProcess

The EDI Module provides this service to batch EDI documents. You do *not* invoke this service from one of your own services. Rather, the EDI Module registers this service in Trading Networks as a scheduled delivery service and assigns it the name EDI Batch.

You can then define a scheduled delivery queue in Trading Networks and associate the EDI Batch service with the queue. When you define the scheduled delivery queue, you specify the values to use for input when the service is invoked. For more information about batching, see Chapter 17, "Batching EDI Documents", in the *webMethods EDI Module User's Guide*.

Input Parameters

queue
String Name of the queue from which to get the EDI documents to batch. Trading Networks provides the name of the queue when it invokes this service.

senderIDQualifier
String For ANSI X12 and UN/EDIFACT, the EDI ID qualifier for the sender, e.g., 01 if the sender is represented as a D-U-N-S number.

For TRADACOMS, this specifies the sender code to be used in the transmission.

How the batchProcess service uses *senderIDQualifer* depends on the value of the *oneBatchQueue* variable. See *senderID* below for more information.

senderID
String For ANSI X12 and UN/EDIFACT, the sender. For example, if you specified 01 for *senderIDQualifier*, specify the sender's D-U-N-S number for *senderID*.

For TRADACOMS, this specifies the sender name to be used in the transmission.

How the batchProcess service uses *senderID* and *senderIDQualifier* depends on the value of the *oneBatchQueue* variable.

If *oneBatchQueue* is:	Description of how the batchProcess service uses *sender*, *senderIDQualifier*, *receiver*, and *receiverIDQualifier*
`SINGLEOUTPUT`	When sorting EDI documents in the queue, the batchProcess service uses the *senderID*, *senderIDQualifier*, *receiverID*, and *receiverIDQualifier* variables only when an EDI document in the queue does not have headers. In this case, the sender and receiver values are used to determine into which collection and subcollection areas the batchProcess service is to sort the transactions in the EDI document.

When creating the BizDocEnvelope for the final EDI batch document, the batchProcess service uses the sender and receiver input variables as the sender and receiver identified in the BizDocEnvelope. As a result, these are the sender and receiver you can use for criteria in a processing rule.

For more information about batching when *oneBatchQueue* is `SINGLEOUTPUT`, see Chapter 17, "Batching EDI Documents", in the *webMethods EDI Module User's Guide*.

`MULTIPLEOUTPUTS` When sorting EDI documents in the queue, the batchProcess service uses the *senderID*, *senderIDQualifier*, *receiverID*, and *receiverIDQualifier* variables only when an EDI document in the queue does not have headers. In this case, the sender and receiver values are used to determine into which collection and subcollection areas the batchProcess service is to sort the transactions in the EDI document.

For more information about batching when *oneBatchQueue* is `MULTIPLEOUTPUTS`, see Chapter 17, "Batching EDI Documents", in the *webMethods EDI Module User's Guide*.

`NONE` The batchProcess service uses *senderID*, *senderIDQualifier*, *receiverID*, and *receiverIDQualifier* input variables to locate the partner-specific EDITPA. The `batchProcess` service uses the variables that you define in the partner-specific and default EDITPAs to control how it combines the EDI documents in a queue into the final EDI batch document. For more information about batching when *oneBatchQueue* is `NONE`, see Appendix B, "Using the 6.0.1 Version of the Batching Feature", in the *webMethods EDI Module User's Guide*.

If the EDITPA *envelopeIdentifier* variables are null in the default and partner-specific EDITPAs, the batchProcess service uses the value you specify for corresponding batchProcess input variable.

ReceiverIDQualifier String For ANSI X12 and UN/EDIFACT, the EDI ID qualifier for the receiver, e.g., 01 if the receiver is represented as a D-U-N-S number.

For TRADACOMS, this specifies the receiver code to be used in the transmission.

How the batchProcess service uses *receiverIDQualifer* depends on the value of the *oneBatchQueue* variable. See *senderID* above for more information.

receiverID

String The receiver. For example, if you specified 01 for *ReceiverIDQualifier*, specify the receiver's D-U-N-S number for *receiverID*.

For TRADACOMS, this specifies the receiver name to be used in the transmission.

How the batchProcess service uses *receiverID* and *ReceiverIDQualifier* depends on the value of the *oneBatchQueue* variable. See *senderID* above for more information.

oneBatchQueue

String (optional) How you want the service to batch the document. Specify one of the following.

Value of *oneBatchQueue*	Meaning
`SINGLEOUTPUT`	Batch the EDI documents in the queue into a single batch EDI document that contains multiple interchanges or transmissions.
`MULTIPLEOUTPUTS`	Batch the EDI documents in the queue into multiple batch EDI documents, each with a single interchange or transmission.
`NONE`	Batch the documents in the queue using the method provided in version 6.0.1 of the EDI Module. For more information, see Appendix B, "Using the 6.0.1 Version of the Batching Feature", in the *webMethods EDI Module User's Guide*. This is the default.

mode

String (optional) The types of envelopes you want the batch EDI document to have. This variable is only used when *oneBatchQueue* is `NONE`. Specify one of the following:

Value of *mode*	Meaning
`IC`	Interchange or TRADACOMS transmission envelope only
`GP`	Group or TRADACOMS batch envelope only
`IC&GP`	Both interchange and group (or TRADACOMS transmission and batch) envelopes. This is the default.

standard

String The EDI standard to use.

- If *oneBatchQueue* is `SINGLEOUTPUT` or `MULTIPLEOUTPUTS`, the batchProcess service only uses *standard* for an interchange in the batch EDI document when a document in the queue does not have an interchange header. When documents have interchange headers, the batchProcess service uses the standard from the header.
- If *oneBatchQueue* is `NONE`, the batchProcess service uses *standard* for the batch EDI document.

Specify one of the following.

Value of *standard*	Meaning

`X12`
`VICS`
`UCS`
	Use ANSI X12 envelopes (ISA/IEA and GS/GE).

`UNEDIFACT` `EANCOM`	Use UN/EDIFACT envelopes (UNA, UNB/UNZ and UNG/UNT).
`TRADACOMS`	Use TRADACOMS envelopes (STX/END, BAT/EOB, and MHD/MTR)

version

String Version of the EDI standard for which to build the envelope, e.g., `4010`.

Note: For TRADACOMS, the only valid version is `1`.

- If *oneBatchQueue* is `SINGLEOUTPUT` or `MULTIPLEOUTPUTS`, the batchProcess service only uses *version* for a batch EDI document when a document in the queue does not have headers. When documents have headers, the batchProcess service uses the version from the header.
- If *oneBatchQueue* is `NONE`, the batchProcess service uses *version* for the batch EDI document.

environment String The environment to indicate in the envelope headers. Specify either `Test` or `Production`.

Note: For TRADACOMS, the only valid value is `Production`.

- If *oneBatchQueue* is `SINGLEOUTPUT` or `MULTIPLEOUTPUTS`, the batchProcess service only uses *environment* for an interchange in the batch EDI document when a document in the queue does not have an interchange header. When documents have interchange headers, the batchProcess service uses the production mode from the header.
- If *oneBatchQueue* is `NONE`, the batchProcess service uses *environment* for the batch EDI document.

controlNumber String How you want the service to create the control number for the envelope. Specify one of the following.

Value of *controlNumber*	Meaning
`fromTable`	The number from the EDIControlNumber table plus the configured control number increment. The EDIControlNumber table is an EDI Module-specific table in the Trading Networks database.
`Sequentialize`	Sequentialize the control number starting from 1.

contentControlNumber String How you want the transaction control numbers to be created.

Note: Not applicable to TRADACOMS documents because transmission references always start from 1.

Specify one of the following:

Value of *contentControlNumber*	Meaning
`sequentialize`	Sequentialize the control number starting from 1.
`none`	Use the control number from the group or transaction header (or TRADACOMS batch or file header).

groupControlNumber String How you want the group control numbers (or TRADACOMS batch transmission references) to be created. Specify one of the following:

Value of *groupControlNumber*	Meaning
`sequentialize`	Sequentialize the group or TRADACOMS batch control number according to the value specified for the *controlNumber* parameter.

`none`	The batchProcess service will not overwrite the group or TRADACOMS batch control number.

Note: If *groupControlNumber* is left blank, the value of *contentControlNumber* is used.

acknowledgement String Whether you want to request FAs for your interchange header segment (UNB/ISA); not applicable to TRADACOMS documents.

Value of *acknowledgement*	Meaning
`true`	Set the envelope to request an FA.
`false`	Set the envelope to *not* request an FA.

delimiters Document (optional) Delimiters to use when forming the batch EDI document. If you do not specify *delimiters*, the batchProcess service attempts to obtain the delimiters from the EDITPA. For more information, see Chapter 17, "Batching EDI Documents", in the *webMethods EDI Module User's Guide*.

Note: The *delimiters* parameter is not applicable to TRADACOMS documents. The EDI Module provides built-in support for the following TRADACOMS delimiters: segment terminator, data element separator, sub-element separator, and segment code separator. You cannot modify this list of delimiters.

Variables in *delimiters*	Description
record	String The segment terminator to use for the batch EDI document, e.g., +.
field	String The field separator for each EDI segment in the batch EDI document, e.g., `!`.
subfield	String The separator for composite elements in the batch EDI document, e.g., *.
release	String The release character to use for the batch EDI document, e.g., ?.

createGroup String (optional) Add a group or a TRADACOMS batch.

Value of *createGroup*	Meaning
`true`	Add a group or a TRADACOMS batch
`false`	Do not add a group or a TRADACOMS batch. This is the default.

priorityCode String (optional) Add a TRADACOMS priority code and a priority for the batch.

Value of *priorityCode*	Meaning
`urgent`	The batch is urgent priority.
`normal`	The batch is normal priority.
`low`	The batch is low priority.

Output Parameters

None

Usage Notes

- For more information about EDITPAs and the variables contained in the, see Chapter 9, "Defining Partner Information (ANSI X12 and UN/EDIFACT", in the *webMethods EDI Module User's Guide*.
- For more information about how the EDI Module batches EDI documents, including how this service is used, see Chapter 17, "Batching EDI Documents", in the *webMethods EDI Module User's Guide*.

wm.b2b.editn.batch:getControlNumber

Gets the current control number from the EDIControlNumber table, which is an EDI Module-specific table in the Trading Networks database that the EDI Module uses to manage control numbers.

Input Parameters

TNsenderID	String For ANSI X12 and UN/EDIFACT, the sender associated with the control number that you want to obtain, e.g., the sender's D-U-N-S number. For TRADACOMS, the sender name specified in the transmission.
senderQualifier	String (optional) For ANSI X12 and UN/EDIFACT, the EDI ID qualifier for the sender, e.g., 01 for a D-U-N-S number. For TRADACOMS, the sender code specified in the transmission.
TNreceiverID	String For ANSI X12 and UN/EDIFACT, the receiver associated with the control number that you want to obtain, e.g., the receiver's D-U-N-S number. For TRADACOMS, the receiver name specified in the transmission.
receiverQualifier	String (optional) For ANSI X12 and UN/EDIFACT, the EDI ID qualifier for the receiver, e.g., 01 for a D-U-N-S number. For TRADACOMS, the receiver code specified in the transmission.
standard	String The EDI standard associated with the control number that you want to obtain. Specify one of the following: `X12`, `UNEDIFACT`, `VICS`, `UCS`, `EANCOM`, `ODETTE`, or `TRADACOMS`.
type	String (optional) The control number type. ■ For an interchange control number, specify `ENVELOPE`. ■ For group control numbers, specify the group type. For example, for ANSI X12 4010 850, specify `PO`. ■ For a TRADACOMS file control number, the type will be the File type, such as INVFIL or ORDHDR ■ For a TRADACOMS batch control number, the type will be the Batch type.

update String (optional) Whether you want the service to update the control number in the database to the next control number to use.

Value of *update*	Meaning
`true`	Update the control number.
`false`	The control number in the database table will not be updated. This is the default

prodMode String (optional) The production mode associated with the control number. Specify one of the following: `Production` (default), `Test`, or `Custom`.

Note: For TRADACOMS, the only valid value is `Production`.

version String (optional) Version of the EDI standard for the control number, e.g., `4010`.

For a TRADACOMS envelope, the version is always `1`. For a TRADACOMS batch, the version is the version number of the file type. For example, for the file type `ORDHDR:5`, the version is `5`.

Output Parameters

controlNumber String (optional) The control number.

error String (optional) If an error was encountered obtaining the control number, this is the description of the error.

Usage Notes

For more information about control numbers, see Chapter 11, "Defining Control Number Information for Partners" in the *webMethods EDI Module User's Guide*.

wm.b2b.editn.crossRef

Use the services in this folder to manage interchange sender/receiver pair information you have defined.

wm.b2b.editn.crossRef:deleteEnvInfo

Deletes interchange sender/receiver pair information from the EDIEnvelope table, which is an EDI Module-specific table in the Trading Networks database.

You define interchange sender/receiver pair information if you want to process EDI documents at the group level.

Input Parameters

senderID	String The sender id of the interchange sender/receiver pair information you want to delete.
senderQual	String The sender EDI ID qualifier of the interchange sender/receiver pair information you want to delete.
receiverID	String The receiver id of the interchange sender/receiver pair information you want to delete.
receiverQual	String The receiver EDI ID qualifier of the interchange sender/receiver pair information you want to delete.
productionMode	String The production mode associated with the interchange sender/receiver pair information you want to delete. Specify one of the following: `Production`, `Test`, -or- `Custom`.

Output Parameters

message	String Status of delete operation.

Usage Notes

- You can define and view interchange sender/receiver pair information from the WmEDIforTN home page by clicking on the Interchange Info link.
- For more information about processing levels and interchange sender/receiver pair information, see Chapter 9, "Defining Partner Information (ANSI X12 and UN/EDIFACT" in the *webMethods EDI Module User's Guide*.
- If you delete interchange information of a specified sender/receiver pair (identified by *senderID*, *senderQual*, *receiverID*, and *receiverQual)* for all production modes, any group pair associated with the envelope information is also deleted. For more information about group pairs that you associate with interchange envelope information, see Chapter 9, "Defining Partner Information (ANSI X12 and UN/EDIFACT" in the *webMethods EDI Module User's Guide*.

wm.b2b.editn.crossRef:getEnvInfo

Obtains interchange sender/receiver pair information from the EDIEnvelope table, which is an EDI Module-specific table in the Trading Networks database.

You define interchange sender/receiver pair information if you want to process EDI documents at the group level.

Input Parameters

senderID	String The sender id of the interchange sender/receiver pair information you want to retrieve.
senderQual	String (optional) The sender EDI ID qualifier of the interchange sender/receiver pair information you want to retrieve.
receiverID	String The receiver id of the interchange sender/receiver pair information you want to retrieve.
receiverQual	String (optional) The receiver EDI ID qualifier of the interchange sender/receiver pair information you want to retrieve.
productionMode	String (optional) The production mode associated with the interchange sender/receiver pair information you want to retrieve. Specify one of the following: `Production` (default), `Test`, -or- `Custom`.

Output Parameters

envInfo Document Variables defined for the interchange sender/receiver pair.

Variable in *envInfo*	Meaning
inboundInfo	Document Variables for processing inbound documents.

Variable in *inboundInfo*	Description
verifyCtrl	String Whether to verify the inbound control number. The value will be either `yes` or `no`.
createDoc	String Whether the Interchange document should be saved. The value will be either `yes` or `no`.
GSSenderQual	String The EDI ID qualifier that corresponds to the sender value on the group header. If the value is *, the EDI ID sender qualifier of the interchange header is used.

	GSReceiverQual	String The EDI ID qualifier that corresponds to the receiver value on the group header. If the value is *, the EDI ID receiver qualifier of the interchange header is used.
	outboundInfo	Document Variables used for outbound processing. That is, delimiters and some field values in the Interchange header segment. For the exact contents, use the webMethods Developer to view the wm.b2b.e3ditn.crossRef:envInfo IS document type, which is the IS document type that the *envinfo* variable references.

Usage Notes

- Delimiters are inside of the *outboundInfo* variable. The output (*outboundInfo*) of this service can be used as input to the wm.b2b.edi.util:addICEnvelope, wm.b2b.edi.util:addGroupEnvelope, and wm.b2b.edi.util:addICEnvelopeEDIFACT services.The ID and EDI ID qualifier pair can be either the interchange or group level.
- You can define and view interchange sender/receiver pair information from the WmEDIforTN home page by clicking on the Interchange Info link.
- For more information about processing levels and interchange sender/receiver pair information, see Chapter 9, "Defining Partner Information (ANSI X12 and UN/EDIFACT" in the *webMethods EDI Module User's Guide*.

wm.b2b.editn.db

Use the services in this folder to delete unwanted information for the EDI Module-specific tables of the Trading Networks database.

wm.b2b.editn.db:deleteControlNumber

Deletes a control number from the EDIControlNumber table, which is an EDI Module-specific table in the Trading Networks database that the EDI Module uses to manage control numbers.

Input Parameters

senderID	String The sender ID associated with the control number table entry that you want to delete.
senderQual	String The sender EDI ID qualifier associated with the control number table entry that you want to delete.
receiverID	String The receiver ID associated with the control number table entry that you want to delete.
receiverQual	String The receiver EDI ID qualifier associated with the control number table entry that you want to delete.
standard	String The EDI standard associated with the control number that you want to delete. Specify one of the following: `X12`, `UNEDIFACT`, `VICS`, `UCS`, `EANCOM`, or `TRADACOMS`.
productionMode	String The production mode associated with the control number. Specify one of the following: `Production` (default), `Test`, or `Custom`.
version	String Version of the EDI standard for the control number, e.g., `4010`.
isEnvelope	String Whether the control number to be deleted is for an envelope or group document. If it is an envelope control number, the value should be “true”. Otherwise the value should be “false”
groupType	String The group type associated with the control number. ■ For an interchange control number, specify `ENVELOPE`. ■ For group control numbers, specify the group type. For example, for ANSI X12 4010 850, specify `PO`.

Output Parameters

None

Usage Notes

For more information about control numbers, see Chapter 11, "Defining Control Number Information for Partners", in the *webMethods EDI Module User's Guide*.

wm.b2b.editn.db:deleteFAInfo

Deletes entries from the EDITRACKING table, which is an EDI Module-specific table in the Trading Networks database that the EDI Module uses to track functional acknowledgements. By deleting entries, you reduce the size of the database.

This service determines the entries to remove by subtracting the value of the input *deleteAfterDays* variable from the current date and deletes entries that are older than the resulting date.

Input Parameters

deleteAfterDays	String The maximum number of days that a functional acknowledgement entry remain in the functional acknowledgement tracking table. If you specify 0, all records are deleted.

Output Parameters

deleteCount	String Number of records deleted.

Usage Notes

For more information about FA reconciliation and how the EDI Module uses the EDITRACKING table, see Chapter 19, "Reconciling Functional Acknowledgements" in the *webMethods EDI Module User's Guide*.

wm.b2b.editn.doc

The EDI Module uses the services in this folder when you work with instance ID queries on the WmEDIforTN package home page.

wm.b2b.editn.doc:listTransactionTypes

Returns a list of all TN document types for EDI transaction sets that exist on the server as well as the instance ID query associated with that TN document type, if one exists.

Input Parameters

None

Output Parameters

DocTypes Document List (optional) The names and instance ID query for each TN document type for an EDI transaction set. The format is:

Variable in *DocTypes*	Description
Name	String Name of the TN document type for the EDI transaction set, e.g., `X12 V4010 T850`.
Query	String (optional) The instance ID query associated with the TN document type.

Usage Notes

This service is generally used by the WmEDIforTN user interface, not by end users.

wm.b2b.editn.doc:saveQuery

Saves the instance ID query for an input TN document type that is for an EDI transaction set.

Input Parameters

DocumentName	String Name of the TN document type for the EDI transaction set, e.g., `X12 V4010 T850`.
Query	String (optional) The instance ID query to associate with the TN document type. If you do not specify *Query*, the instance ID query that is currently associated with the TN document type is cleared and at run time, Trading Networks will not extract a conversation ID when processing transactions set that use this TN document type.

Output Parameters

None

Usage Notes

This service is generally used by the WmEDIforTN user interface, not by end users. The query is saved successfully unless this service throws an exception.

wm.b2b.editn.FAReport

Use services the service in this folder to generate FA reconciliation reports.

wm.b2b.editn.FAReport:generateFAReport

Based on the inputs you specify, this service retrieves the matching rows from the EDITRACKING table and generates a report.

The EDITRACKING table is an EDI Module-specific table in the Trading Networks database that contains information about FA reconciliations.

Input Parameters

reportFileName String (optional) The name you want to assign the FA report file. If you do *not* specify *reportFileName*, this service assigns the file name will be *timestamp reportFileName*.template.

For example, `2003_07_01testReport.html`

template String Whether you want the report in text or HTML format. Specify one of the following:

Value of *template*	Meaning
`TXT`	The report will be in text format.
`HTML`	The report will be in HTML format.

senderID String (optional) The corporate name (as specified in the Trading Networks profile) of the sender of the EDI document you want included in the report. If you do *not* specify *senderID*, the report will contain information for documents from all senders.

receiverID String (optional) The corporate name (as specified in the Trading Networks profile) of the receiver of the EDI document you want included in the report. If you do *not* specify *receiverID*, the report will contain information for documents from all senders.

status String (optional) The number that represents the FA status of the documents that you want included in the report. If you do *not* specify *status*, the report will contain information for documents with any FA status. The following lists the numbers to specify for each FA Status:

Value of *status*	Meaning
100	FA_STATUS_NONE
110	FA_STATUS_DISABLE

120	FA_STATUS_DUPLICATE
130	FA_STATUS_ERROR
140	FA_STATUS_DUP_FA
150	FA_STATUS_ACCEPT
160	FA_STATUS_ACCEPT_ERROR
170	FA_STATUS_ACCEPT_PARTIAL
180	FA_STATUS_REJECT
190	FA_STATUS_FA_ERROR
200	FA_STATUS_INTERCHANGE_RECEIVED

For more information about FA status, see Chapter 19, "Reconciling Functional Acknowledgements", in *webMethods EDI Module User's Guide*.

docBeforeYear String (optional) Indicates that you want the report to include information for original EDI documents sent before the year you specify. For example, if you specify `2002`, the service would include entries in which the original EDI documents were sent before 2002.

docBeforeMonth String (optional) Indicates that you want the report to include information for original EDI documents sent before the month you specify. For example, if you specify `3`, the service would include entries in which the original EDI documents were sent before March.

docBeforeDay String (optional) Indicates that you want the report to include information for original EDI documents sent before the day you specify. For example, if you specify `12`, the service would include entries in which the original EDI documents were sent before the 12th.

docBeforeHour String (optional) Indicates that you want the report to include information for original EDI documents sent before the hour you specify (in 24-hour format). For example, if you specify `13`, the service would include entries in which the original EDI documents were sent before 1 P.M.

docBeforeMinute String (optional) Indicates that you want the report to include information for original EDI documents sent before the minute you specify. For example, if you specify `30`, the service would include entries in which the original EDI documents were sent before the 30th minute.

docAfterYear String (optional) Indicates that you want the report to include information for original EDI documents sent after the year you specify. For example, if you specify `2002`, the service would include entries in which the original EDI documents were sent after 2002.

docAfterMonth String (optional) Indicates that you want the report to include information for original EDI documents sent after the month you specify. For example, if you specify `3`, the service would include entries in which the original EDI documents were sent after March.

docAfterDay String (optional) Indicates that you want the report to include information for original EDI documents sent after the day you specify. For example, if you specify `12`, the service would include entries in which the original EDI documents were sent after the 12th.

docAfterHour String (optional) Indicates that you want the report to include information for original EDI documents sent after the hour you specify (in 24-hour format). For example, if you specify `13`, the service would include entries in which the original EDI documents were sent after 1 P.M.

docAfterMinute String (optional) Indicates that you want the report to include information for original EDI documents sent after the minute you specify. For example, if you specify `30`, the service would include entries in which the original EDI documents were sent after the 30th minute.

FABeforeYear String (optional) Indicates that you want the report to include information for FAs sent before the year you specify. For example, if you specify `2002`, the service would include entries in which the FAs were sent before 2002.

FABeforeMonth String (optional) Indicates that you want the report to include information for FAs sent before the month you specify. For example, if you specify `March`, the service would include entries in which the FAs were sent before March.

FABeforeDay String (optional) Indicates that you want the report to include information for FAs sent before the day you specify. For example, if you specify `12`, the service would include entries in which the FAs were sent before the 12th.

FABeforeHour String (optional) Indicates that you want the report to include information for FAs sent before the hour you specify (in 24-hour format). For example, if you specify `13`, the service would include entries in which the original EDI documents were sent before 1 P.M.

FABeforeMinute String (optional) Indicates that you want the report to include information for FAs sent before the minute you specify. For example, if you specify `30`, the service would include entries in which the original EDI documents were sent before the 30th minute.

FAAfterYear String (optional) Indicates that you want the report to include information for FAs sent after the year you specify. For example, if you specify `2002`, the service would include entries in which the FAs were sent after 2002.

FAAfterMonth String (optional) Indicates that you want the report to include information for FAs sent after the month you specify. For example, if you specify `March`, the service would include entries in which the FAs were sent after March.

FAAfterDay	String (optional) Indicates that you want the report to include information for FAs sent after the day you specify. For example, if you specify `12`, the service would include entries in which the FAs were sent after the 12th.
FAAfterHour	String (optional) Indicates that you want the report to include information for FAs sent after the hour you specify (in 24-hour format). For example, if you specify `13`, the service would include entries in which the original EDI documents were sent after 1 P.M.
FAAfterMinute	String (optional) Indicates that you want the report to include information for FAs sent after the minute you specify. For example, if you specify `30`, the service would include entries in which the original EDI documents were sent after the 30th minute.
groupType	String (optional) The group type (e.g., `PO`, `IN`) of the of the documents that you want included in the report.
version	String (optional) The version (e.g., `4010`) of the of the documents that you want included in the report.
orderBy	String The number that represents the value by which you want to order the entries included in the report.

Value of *orderBy*	Meaning
1	ORDER_BY_DOCID
2	ORDER_BY_DOCTYPEID
3	ORDER_BY_SENDERID
4	ORDER_BY_RECEIVERID
5	ORDER_BY_ENVELOPEID
6	ORDER_BY_GROUPID
7	ORDER_BY_TRANSACTIONSETID
8	ORDER_BY_GROUPTYPE
9	ORDER_BY_GROUPVERSION
10	ORDER_BY_DOCTIMESTAMP
11	ORDER_BY_FATIMESTAMP
12	ORDER_BY_FASTATUS
13	ORDER_BY_RELATEDDOCID

Output Parameters

None

Usage Notes

For more information about FA reconciliation, see Chapter 19, "Reconciling Functional Acknowledgements" in the *webMethods EDI Module User's Guide*.

wm.b2b.editn.migration

Use the service in this folder when migrating data from EDI Module 4.x to EDI Module 6.*x*.

wm.b2b.editn.migration:updateBizDocs

Creates flat file schemas for all TN document types that are for EDI transaction sets installed on this machine.

Input Parameters

None

Output Parameters

None

Usage Notes

Use this service when upgrading from EDI Module version 4.*x* to version 6.*x*. This service creates a flat file schema in the WmEDIforTN package in the location specified by the wm.b2b.edi.util:getEDIFFSchemaName service.

wm.b2b.editn.rec

This folder contains IS document types that the EDI Module provides.

wm.b2b.editn.rec:batchFailDocument

Defines the format of the IS document that the EDI Module can publish when it is unable to include an EDI document that is queued for batching into the final batch EDI document.

To handle this failure, you can use the webMethods Developer to create an Integration Server trigger that subscribes to the published document. For information on creating an Integration Server trigger, see the *Publish-Subscribe Developer's Guide*.

The EDI Module publishes the IS document if the *publishBatchFailEvent* EDITPA variable is set to `true`. For more information about this EDITPA variable, see Chapter 9, "Defining Partner Information (ANSI X12 and UN/EDIFACT" in the *webMethods EDI Module User's Guide*. For more information about batching EDI documents, see Chapter 17, "Batching EDI Documents" in the *webMethods EDI Module User's Guide*.

Variables in the *batchFailDocument* IS document type

taskID	String The task ID that Trading Networks generated for the delivery task that is associated with the document that the EDI Module could not include into the final batch EDI document (i.e., the failed document).
TNinternalID	String The internal ID that Trading Networks generated for the failed document.
senderID	String The internal ID that Trading Networks generated for the partner that is the sender of the failed document.
receiverID	String The internal ID that Trading Networks generated for the partner that is the receiver of the failed document.
doctypeName	String The name of the TN document type for the failed document.
mode	String The production mode of the failed document, i.e., `Production` or `Testing`.
groupType	String The EDI group type that is associated with the failed document. ■ For an Interchange document, *groupType* is `Envelope`. ■ For a Group document, *groupType* is the type associated with the group, e.g., `PO`. ■ For a Transaction document, *groupType* is the type associated with the group with which the transaction is associated, e.g., `PO` for an ANSI X12 850 transaction.
standard	String The EDI standard that is associated with the failed document, e.g., `X12` or `UNEDIFACT`.
version	String The version of the EDI standard associated with the failed document, e.g., `4010` for ANSI X12 or `98A` for UN/EDIFACT.

wm.b2b.editn.TPA

Use the services in this folder to work with EDITPAs.

wm.b2b.editn.TPA:getEDITPAData

Retrieves the EDITPA data for a specified sender and receiver.

The service retrieves the default EDITPA and partner-specific EDITPA if one is available. The service returns a single set of EDITPA values. The returned EDITTPA values use the partner-specific EDITPA values unless the value is not specified in the partner-specific EDITPA. If a partner-specific EDITPA value is not specific, the returned EDITPA value contains the value from the default EDITPA.

Input Parameters

sender	String The sender ID for the sender of the sender/receiver pair for which you want to retrieve EDITPA values, for example, a D-U-N-S number.
senderQualifier	String The EDI ID qualifier for the sender, for example, `01` if you specify a D-U-N-S number for *sender*.
receiver	String The receiver ID for the receiver of the sender/receiver pair for which you want to retrieve EDITPA values, for example, a D-U-N-S number.
receiverQualifier	String The EDI ID qualifier for the receiver, for example, `01` if you specify a D-U-N-S number for *receiver*.

Output Parameters

EDITPAData	Document A single set of EDITPA values that contains values from the partner-specific EDITPA if they are available and values from the default EDITPA if the partner-specific value is null or not specified.

wm.b2b.editn.util

Use the service in this folder to return the content of the Trading Networks BizDocContentPart as an InputStream object.

wm.b2b.editn.util:getContentPartDataAsInputStream

Returns the content of the Trading Networks BizDocContentPart as an InputStream object.

This service correctly handles content parts that are stored as large data or small data.

Input Parameters

contentPart	Object An IData object that represents a Trading Networks *bizDocEnvelope* content part. This is the output returned by wm.b2b.edi.tradacoms.doc:getContentPart.

Output Parameters

partInputStream	Object An InputStream object that represents the input *contentPart*.
encoding	String The encoding that can be used to convert the input stream to a string.

wm.b2b.editn.util.reprocess

This folder contains utility services that you can use to work with documents that you might want to reprocess.

wm.b2b.editn.util.reprocess:listInSequence

Retrieves a list of EDI documents that contained out-of-sequence control numbers when they were originally received, but now the control numbers are in sequence due to other EDI documents arriving.

Input Parameters

None

Output Parameters

docList String List A list that contains the Trading Networks internal IDs for the retrieved EDI documents that are now in sequence.

wm.b2b.editn.util.reprocess:listUnprocessedDocuments

Retrieves a list of EDI documents that were not processed due to a validation error.

Input Parameters

type String Specifies the types of EDI documents that you want to retrieve. Specify one of the following:

Value of *type*	Meaning
`Duplicate`	List EDI documents that were not processed because they contained duplicate control numbers.
`OutOfSequence`	List EDI documents that were not processed because they contained control numbers that were out of sequence.
`Rejected`	List EDI documents that were not processed because they have FA statuses that are defined as unacceptable. You define what the acceptable and unacceptable FA statuses are using the *FAGeneration/processDocument* EDITPA variable. For more information, see the section about automatic FA generation in Chapter 13, "Optional Inbound Processing When using Trading Networks" of the *webMethods EDI Module User's Guide*.

Output Parameters

docList	String List A list that contains the Trading Networks internal IDs for the retrieved EDI documents that match the *type* you specified.

Usage Notes

To obtain the content for an EDI document, invoke the wm.tn.doc:view service, supplying the Trading Networks internal IDs. For more information about the wm.tn.doc:view service, see the *webMethods Trading Networks Built-In Services Reference*.

wm.b2b.editn.util.reprocess:nextInSequenceDoc

Determines whether there is another EDI document that had an out-of-sequence control number that can now be processed because the specified document has been processed.

Input Parameters

bizdoc	Document A BizDocEnvelope that represents an EDI document that had an out-of-sequence control number that became in sequence after the EDI Module processed other documents. You have previously invoked the wm.b2b.editn.util.reprocess:reprocessDocument service against this BizDocEnvelope to process it. Now you are invoking this service to determine whether there is another EDI document that is now in sequence because the EDI document represented by the specified BizDocEnvelope was processed.

Output Parameters

nextDoc	String The Trading Networks internal ID of an EDI document that is now in sequence. If there is no next document, *nextDoc* is null.

wm.b2b.editn.util.reprocess:reprocessDocument

Force the reprocessing of an EDI document, which was not processed due to a validation error. To reprocess the EDI document, it is sent to Trading Networks processing rules, bypassing document recognition.

This service splits the EDI document according to the EDITPA *splitOption* variable and sends the resulting documents that it split out to Trading Networks processing rules separately. For example, if you specify an Interchange document and the *splitOption* variable is set to Transaction, this service splits the EDI document into an Interchange document, Group documents, and Transaction documents; then sends the Interchange, Group, and Transaction documents to Trading Networks processing rules for processing.

Input Parameters

internalID String The Trading Networks internal ID associated with the EDI document that you want to reprocess.

generateFA String Whether you want this service to automatically generate functional acknowledgements (FA) when reprocessing the document. Specify `true` or `false`.

You should *only* specify `true` when you are reprocessing a document that contains an invalid interchange control number. For documents that have a REJECTED status or were rejected due to invalid group control number, the FAs were already generated when the document was originally processed.

Value of *generateFA*	Meaning
`true`	Automatically generate FAs.
`false`	Do *not* generate FAs.

updateControlNumber String Whether you want this service update the next expected control number in the EDIControlNumber table.

This service calculates the next expected control number by adding the control number from the reprocessed document to the configured control number increment. If the result exceeds the configured control number cap, this service sets the next expected control number to the configured control number minimum. For more information, see the chapter about control numbers in the *webMethods EDI Module User's Guide*. Specify `true` or `false`.

Value of *updateControlNumber*	Meaning
`true`	Update the next expected control number in the EDIControlNumber table.
`false`	Do *not* update the next expected control number in the EDIControlNumber table.

Output Parameters

bizdoc Document The BizDocEnvelope for the document that was reprocessed.

Usage Notes

To obtain a list of EDI documents that were not processed due to validation errors and obtain their Trading Networks internal IDs, you can use the wm.b2b.editn.util.reprocess:listUnprocessedDocuments service.

wm.b2b.editn.util.reprocess:validateControlNumber

Determines whether the specified control number is the one expected for the specified sender/receiver pair. If the specified control number (in the *numberToValidate* input variable) matches the next expected control number for the sender/receiver pair, this service updates the next expected control number.

This service calculates the new expected control number by adding the specified control number (in the *numberToValidate* input variable) to the configured control number increment. If the result exceeds the configured control number cap, this service sets the next expected control number to the configured control number minimum. For more information, see the chapter about control numbers in the *webMethods EDI Module User's Guide*.

Input Parameters

senderID String The sender ID associated with the control number.

senderQualifier String The sender EDI ID qualifier associated with the control number.

receiverID String The receiver ID associated with the control number.

receiverQualifier String The receiver EDI ID qualifier associated with the control number.

standard String EDI standard associated with the control number. Specify one of the following: `X12`, `UNEDIFACT`, `VICS`, `UCS`, -or- `EANCOM`.

productionMode String Production mode associated with the control number. Specify one of the following: `Production` (default), `Test`, or `Custom`.

version String Version of the EDI standard for the control number, e.g., `4010`.

isEnvelope String Whether the control number is for an envelope or a group. Specify `true` or `false`.

Value of *isEnvelope*	Meaning
`true`	The control number is associated with an envelope.
`false`	The control number is associated with a group.

groupType String The group type associated with the control number.

- For an interchange control number, specify `ENVELOPE`.
- For group control numbers, specify `the group type`. For example, for ANSI X12 4010 850, specify `PO`.

createIfNotFound String Whether you want to create an entry for this control number this sender/receiver pair if there is currently no entry. Specify `true` or `false`.

Value of *createIfNotFound*	Meaning
`true`	Creates a control number entry for the specified sender/receiver, production mode, standard and version combination and sets the next expected control number. The next expected control number is calculated by adding the specified control number (in the *numberToValidate* input variable) to the configured control number increment. If the result exceeds the configured control number cap, the next expected control number is set to the configured control number minimum
`false`	Does not create a control number entry for the specified sender/receiver, production mode, standard and version combination.

numberToValidate String The control number that you want to validate.

Output Parameters

result String Whether the control number you specified in the *numberToValidate* input variable is valid. The value of *result* will be one of the following:

Value of *result*	Meaning
`Valid`	The control number that you specified is valid and the next expected control number has been incremented.
`OutOfSequence`	The control number that you specified is *not* valid. The service has determined that the control number is out-of-sequence.
`Duplicate`	The control number that you specified is *not* valid. The service has determined that the control number is a duplicate.

wm.b2b.editn.util.VersionSupport

This folder contains a utility service that you can use to enable the EDI Module to support a new version of an EDI standard.

wm.b2b.editn.util.VersionSupport:addNewEDIVersion

Enables the EDI Module to support a new version of an EDI standard that is already supported by the EDI Module.

Input Parameters

SEFfileName String The fully qualified path of the SEF file.

replace Specifies whether to overwrite an existing SEF file of the same name and its associated TN document type file. Specify `yes` or `no`.

Value of *replace*	Meaning
`yes`	If a SEF file of the same name already exists in the pub\SEFS\standard of the WmEDI package (and its associated TN document type file already exists in the config directory of the WmEDIforTN package), the service overwrites the existing SEF file and TN document type with the ones specified in this service.
`no`	Does not overwrite the specified SEF file and its associated TN document type file; the service throws an exception. This is the default.

Output Parameters

None

VAN.VANConnectivity

The EDI Module uses the services in this folder to enable VAN connectivity as described in Chapter 18, "Retrieving and Delivering EDI Documents from and to VANs", of the *webMethods EDI Module User's Guide*. However, you can use these services for testing purposes if needed.

VAN.VANConnectivity:FTPConnection

Opens the initial FTP connection.

Input Parameters

serverName String The name or IP address of the VAN, e.g., `ftp.icctrade.com`.

userName String Valid user on the remote VAN server, e.g., `anonymous`.

passWord String A valid password for the VAN user specified in *username*, e.g., `someone@somewhere`.

portNum String (optional) The port number on which the FTP server listens for requests, e.g., `4566`. The default is `21`.

account String (optional) A valid account for the VAN user specified in *username*, e.g., `someone`.

retryLimits String (optional) The number of times to attempt to reconnect to the VAN in the event the initial FTP connection fails. The default is zero (0).

dataPort String (optional) The listener port number of the data transfer channel, e.g., `3345`. If you do not specify this information, the Integration Server chooses the listener port number.

waitTime String (optional) The number of seconds to wait between attempts to build the connection to the VAN. If you do not specify the value, the default is zero (0).

encoding String (optional) Character set in which the document is encoded, e.g., `ISO-8859-1`. This variable converts the String object to bytes correctly. Specify an IANA-registered character set. If this variable is null, the default JVM encoding is used.

timeout String (optional) The number of seconds to wait for a response from the FTP server before timing out and aborting the request. The default is zero (0), which signifies to wait indefinitely.

secureFTP String Indicates whether the remote FTP server is a secure server.

Value of *secureFTP*	Meaning
`true`	The FTP server is a secure server.
`false`	The FTP server is not a secure server.

secureFTPOption Document Includes the options *securedata* and *auth*.

Variable in *secureFTPOption*	Meaning
securedata	Specify `true` to protect the FTP data channel, or `false`.
auth	Authentication/security mechanism. Specify `SSL`, `TLS`, or `TLS-P`.

Output Parameters

sessionkey String Unique key that identifies current session information.

VAN.VANConnectivity:getFromVAN

Gets documents from the specified VAN.

Input Parameters

connectionInfo Document The information needed to connect to the VAN.

Variable in *connectionInfo*	Description
serverName	String The name or IP address of the VAN, e.g., `ftp.icctrade.com`.
userName	String Valid user on the remote VAN server, e.g., `anonymous`.
passWord	String A valid password for the VAN user specified in *username*, e.g., `someone@somewhere`.
portNum	String (optional) The port number on which the FTP server listens for requests, e.g., `4566`. The default is `21`.
account	String (optional) A valid account for the VAN user specified in *username*, e.g., `someone` .
dataPort	String (optional) The listener port number of the data transfer channel, e.g., `3345`. If you do not specify this information, the Integration Server chooses the listener port number.
encoding	String (optional) Character set in which the document is encoded, e.g., `ISO-8859-1`. This variable converts the String object to bytes correctly. Specify an IANA-registered character set. If this variable is null, the default JVM encoding is used.

	waitTime	String (optional) The number of seconds to wait between attempts to build the connection to the VAN. If you do not specify the value, the default is zero (0).
	timeout	String (optional) The number of seconds to wait for a response from the FTP server before timing out and aborting the request. The default is zero (0), which signifies to wait indefinitely.
	retryLimits	String (optional) The number of times to attempt to reconnect to the VAN in the event the initial FTP connection fails. The default is zero (0).
	secureFTP	String Indicates whether the remote FTP server is a secure server. Specify `true` or `false`.
	secureFTPOption	Document Includes the options *securedata* and *auth*. *securedata* — Specifies whether to protect the FTP data channel. Specify `true` or `false`. *auth* — Authentication/security mechanism. Specify `SSL`, `TLS`, or `TLS-P`.
command	String The command to use to get inbound documents from the VAN. Specify one of the following:	
	Value of *command*	Meaning
	`GET`	Get only the files that you specify from the VAN.
	`MGET`	Get *all* files from the VAN.
remotefile	String List Name(s) of the specific inbound document(s) you would like to get. You must specify *remotefile* if you set *command* to `GET`.	
filenamepattern	String (optional) A pattern that specifies to get files with a specific file pattern. For example, if you want to get all files ending in a .dat extension, specify `*.dat`. You can use *filenamepattern* when you set *command* to `MGET` (but is not mandatory). This input variable (*filenamepattern*) is ignored if you set *command* to `GET`.	
inboundDirpath	String (optional) Local path to the directory from which you pick up documents from the VAN.	
transfermode	String (optional) The FTP file transfer mode. Specify either `ascii` or `binary`.	

saveInboundtoTN

String Whether you want to submit inbound documents to Trading Networks (wm.tn:receive). Specify either `yes` or `no`.

Value of *saveInboundtoTN*	Meaning
`yes`	Submit the inbound documents to Trading Networks.
`no`	Do *not* submit the inbound documents to Trading Networks.

getReport

String Whether you want to get VAN-generated reports after receiving inbound documents. This occurs within the same session. Specify either `yes` or `no`.

Value of *getReport*	Meaning
`yes`	Get VAN-generated reports.
`no`	Do *not* get VAN-generated reports.

encoding

String (optional) Character set in which the document is encoded, e.g., `ISO-8859-1`. This variable converts the String object to bytes correctly. Specify an IANA-registered character set. If this variable is null, the default JVM encoding is used.

PGPEnable

String Whether you want to verify and PGP decrypt documents. Specify `true` or `false`.

Note: PGP-encryption support is deprecated; it will not be available in a future release of the EDI Module.

PGP-encryption is supported *only* as part of the VAN connectivity to ICC.net. PGP-encryption is *not* generically supported across webMethods components.

Value of *PGPEnable*	Meaning
`true`	Verify and decrypt the documents.
`false`	Do *not* verify and decrypt the documents.

PGPInfo

Document Information needed for PGP encryption/decryption.

Variable in *PGPInfo*	Description
PGPprivateKeyfile	String PGP private key file name, e.g., `c:\PGP\private.asc`. If *PGPEnable* is set to `true`, it is required.
PGPpublicKeyfile	String PGP public key file name, e.g., `c:\PGP\public.asc`. If *PGPEnable* is set to `true`, it is required.
passphrase	String PGP password or passphrase. If *PGPEnable* is set to `true`, it is required.

reportInfo Document Information about the reports you want retrieved and where to store them on your system.

Variable in *reportInfo*	Description
reports	String List Name(s) of the reports you want to get, e.g., `statfile`. If *getReport* is enabled, it is required.
repDir	String (optional) Local path to the directory from which you pick up VAN-generated reports. e.g., `root\records`.
saveToDir	String (optional) Local file path directory in which you want to save reports. This is in addition to saving them in the Trading Networks activity log. The activity log truncates messages that are larger than 1KB, so webMethods recommends that you provide a local file path in which to save reports that exceed 1KB. The default directory is *`webMethods6`*`\`*`IntegrationServer`*`\packages\WmEDIforTN\pub\VANReports`. To save reports to a directory other than the default, the directory must be listed in the *`webMethods6`*`\`*`IntegrationServer`*`\packages\WmEDIforTN\config\VANReportsDirectory.cnf` file. Open the file and add any additional directories in which you want to allow reports to be saved.

logout String Whether you would like to terminate the FTP session after transactions have been completed. Specify either `yes` or `no`.

Value of *logout*	Meaning
`yes`	Terminate the FTP session.
`no`	Do *not* terminate the FTP session.

Output Parameters

EDIdata Document The data content that was retrieved from the VAN.

Variable in *EDIdata*	Meaning
string	String The data from the VAN.
stream	Object Data from the VAN in a java.io.InputStream object.

Usage Notes

For more information about using this service to retrieve EDI documents to a VAN, see Chapter 18, "Retrieving and Delivering EDI Documents from and to VANs" in the *webMethods EDI Module User's Guide*.

VAN.VANConnectivity:getReportFromVAN

Gets VAN-generated reports from the VAN.

Input Parameters

reports	String List Name(s) of the reports that you want to retrieve, e.g., `statfile`.
repDir	String (optional) Local path to the directory from which you pick up VAN-generated reports. e.g., `root\records`.
saveToDir	String (optional) Local file path directory in which you want to save reports. This is in addition to saving them in the Trading Networks activity log. The activity log truncates messages that are larger than 1KB, so webMethods recommends that you provide a local file path in which to save reports that exceed 1KB. The default directory is *`webMethods6`*`\`*`IntegrationServer`*`\packages\WmEDIforTN\pub\VANReports`. To save reports to a directory other than the default, the directory must be listed in the *`webMethods6`*`\`*`IntegrationServer`*`\packages\WmEDIforTN\config\VANReports\Directory.cnf` file. Open the file and add any additional directories in which you want to allow reports to be saved.
sessionkey	String Unique key that identifies current session information.

Output Parameters

None

Usage Notes

The services VAN.VANConnectivity:getFromVAN and VAN.VANConnectivity:putToVAN invoke this service.

VAN.VANConnectivity:putToVAN

The EDI Module provides this service to send EDI documents to a VAN.

You do *not* invoke this service from one of your own services. Rather, the EDI Module registers this service in Trading Networks as a scheduled delivery service and assigns it the name VANFTP. You can then define a scheduled delivery queue in Trading Networks and associate the VANFTP service with the queue. When you define the scheduled delivery queue, you specify the values to use for input when the service is invoked.

Trading Networks invokes the service to act on the documents in the scheduled delivery queue. When invoked, the VANFTP service extracts all the documents in the queue to send them to the VAN.

Input Parameters

queue String Name of the queue from which to get the EDI documents that you want to send to the VAN. Trading Networks provides the name of the queue when it invokes this service.

connectionInfo Document The information needed to connect to the VAN.

Variable in *connectionInfo*	Description
serverName	String The name or IP address of the VAN, e.g., `ftp.icctrade.com`.
userName	String Valid user on the remote VAN server, e.g., `anonymous`.
passWord	String A valid password for the VAN user specified in *username*, e.g., `someone@somewhere`.
portNum	String (optional) The port number on which the FTP server listens for requests, e.g., `4566`. The default is `21`.
account	String (optional) A valid account for the VAN user specified in *username*, e.g., `someone` .
dataPort	String (optional) The listener port number of the data transfer channel, e.g., `3345`. If you do not specify this information, the Integration Server chooses the listener port number.
encoding	String (optional) Character set in which the document is encoded, e.g., `ISO-8859-1`. This variable converts the String object to bytes correctly. Specify an IANA-registered character set. If this variable is null, the default JVM encoding is used.
waitTime	String (optional) The number of seconds to wait between attempts to build the connection to the VAN. If you do not specify the value, the default is zero (0).
timeout	String (optional) The number of seconds to wait for a response from the FTP server before timing out and aborting the request. The default is zero (0), which signifies to wait indefinitely.
retryLimits	String (optional) The number of times to attempt to reconnect to the VAN in the event the initial FTP connection fails. The default is zero (0).
secureFTP	String Indicates whether the remote FTP server is a secure server. Specify `true` or `false`.
secureFTPOption	Document Includes the options *securedata* and *auth*.

securedata — Specifies whether to protect the FTP data channel. Specify `true` or `false`.

auth — Authentication/security mechanism. Specify `SSL`, `TLS`, or `TLS-P`.

outboundDirpath String (optional) Local path to the directory in which you drop off documents to the VAN.

PGPEnable String Whether you want to sign and PGP encrypt documents. Specify `true` or `false`.

Note: PGP-encryption is supported *only* as part of the VAN connectivity to ICC.net. PGP-encryption is *not* generically supported across webMethods components.

Value of *PGPEnable*	Meaning
`true`	Sign and PGP encrypt the documents.
`false`	Do *not* sign and PGP encrypt the documents.

PGPInfo Document Information needed for PGP encryption/decryption.

Variable in *PGPInfo*	Description
PGPprivate Keyfile	String PGP private key file name, e.g., `c:\PGP\private.asc`. If *PGPEnable* is set to `true`, it is required.
PGPpublic Keyfile	String PGP public key file name, e.g., `c:\PGP\public.asc`. If *PGPEnable* is set to `true`, it is required.
passphrase	String PGP password or passphrase. If *PGPEnable* is set to `true`, it is required.

getInbound String Whether you would like to get inbound documents from the VAN after sending outbound documents to the VAN. This occurs within the same session. All documents retrieved are submitted to Trading Networks for processing.

Value of *getInbound*	Meaning
`yes`	Retrieve EDI documents during the same session. You must specify *InboundInfo*.
`no`	Do *not* retrieve EDI documents during the same session.

InboundInfo

Document Information describing the files to retrieve from the VAN. This variable is required when *getInbound* is set to `yes`.

Variable in *InboundInfo*	Description
command	String The command to use to get inbound documents from the VAN. Specify one of the following:

Value of *Command*	Meaning
`GET`	Get only the files that you specify from the VAN.
`MGET`	Get *all* files from the VAN.

remotefile	String List Name(s) of the specific inbound document(s) you would like to get. You must specify *remotefile* if you set *command* to `GET`.
filenamepattern	String (optional) A pattern that specifies to get files with a specific file pattern. For example, if you want to get all files ending in a .dat extension, specify `*.dat`. You can use *filenamepattern* when you set *command* to `MGET` (but is not mandatory). This input variable (*filenamepattern*) is ignored if you set *command* to `GET`.
inboundDirpath	String (optional) Local path to the directory from which you pick up documents from the VAN.

getReport

String Whether you want to get VAN-generated reports. This occurs within the same session. Specify either `yes` or `no`.

Value of *getReport*	Meaning
`yes`	Get VAN-generated reports.
`no`	Do *not* get VAN-generated reports.

reportInfo

Document Information about the reports you want retrieved and where to store them on your system.

Variable in *reportInfo*	Description
reports	String List Name(s) of the reports you want to get, e.g., `statfile`. If *getReport* is enabled, it is required.
repDir	String (optional) Local path to the directory from which you pick up VAN-generated reports. e.g., `root\records`.

	saveToDir	String (optional) Local file path directory in which you want to save reports. This is in addition to saving them in the Trading Networks activity log. The activity log truncates messages that are larger than 1KB, so webMethods recommends that you provide a local file path in which to save reports that exceed 1KB. The default directory is *webMethods6**IntegrationServer*\packages\WmEDIforTN\pub\VANReports. To save reports to a directory other than the default, the directory must be listed in the *webMethods6**IntegrationServer*\packages\WmEDIforTN\config\VANReports\Directory.cnf file. Open the file and add any additional directories in which you want to allow reports to be saved.
logout	String Whether you would like to terminate the FTP session after transactions have been completed. Specify either `yes` or `no`.	

Value of *logout*	Meaning
`yes`	Terminate the FTP session.
`no`	Do *not* terminate the FTP session.

Output Parameters

None

Usage Notes

- For more information about using this service to send EDI documents to a VAN and how to set up the scheduled delivery queue, see Chapter 18, "Retrieving and Delivering EDI Documents from and to VANs" in the *webMethods EDI Module User's Guide*.
- This service will not publish a Notification Failure document if it is unable to connect to the VAN.

Index

D

E

F

G

I

J

L

M

N

O

P

R

S

T

U

V

W

X