webMethods OFTP Adapter

User's Guide

VERSION 7.1

MARCH 2009

This document applies to webMethods OFTP Adapter Version 7.1 and to all subsequent releases.

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

ADAPTER-OFTP-UG-71-20090301

CONTENTS

About This Guide

Welcome!

The ODETTE File Transfer Protocol (OFTP) according to the VDA recommendation 4914/2 is a popular file transfer protocol for exchanging data of any format or contents. Specifically, it is widely used in automotive industry to exchange business documents in ODETTE format. The OFTP Adapter helps webMethods' European customers to exchange business documents with trading partners already doing OFTP.

You should be familiar with configuring Trading Networks to exchange business documents before performing the procedures in this guide. For information, see the *Getting Started with Trading Networks* and *the Building Your Trading Network* manuals.

You should be familiar with the OFTP communication standard.

Typographical Conventions

This document uses the following typographical conventions:

Convention	Example
Procedures are highlighted by a blue box in the left column. Procedures are described as a series of numbered steps.	**Procedure Title** 1. Select the **File** command from the **Activity** menu.
Characters that you must type exactly are shown in a typewriter (`courier`) font.	Type: `setup` and then press ENTER.
Variable information that you must type based on your specific situation or environment is shown in italics.	Type: *`webMethods7`*`\`*`IntegrationServer`*`\setup` and then press ENTER.
Keyboard keys are shown in uppercase.	Press ENTER; then press TAB.
Keys that you must press simultaneously are joined with the "+" symbol.	Press CTRL+ALT+M.
Directory paths are shown with the "\" directory delimiter unless the subject is UNIX-specific. In these cases, the "/" is used. If you are working in a UNIX environment, substitute a "/" for the "\" shown in the procedures in this book.	*`webMethods7`*`\`*`IntegrationServer`* `\packages\Default`

Convention	Example
Information that you must read before beginning a procedure or that alerts you to negative consequences of certain actions is denoted with this special symbol.	 **Caution:** If the folder is not already open in the Document Browser, open it before you start the following procedure.
Notes that provide related, but non-critical, information are denoted with this special symbol.	 **Note:** When you start the Trading Networks Console, you will be asked to log on to a webMethods for Trading Networks Server.

Program Code Conventions

For programming code and command syntax, this document uses the following typographical conventions:

Convention	Example
Keywords and values that you must type exactly as printed are shown in typewriter (`courier`) font.	`%CoSymbol%`
Variable values or parameters that you must supply are shown in italics.	`%`*`VarName`*`%`
Keywords or values that are optional are enclosed in [ ]. Do *not* type the [ ] symbols in your own code.	`%loop` *`LoopVar`* `[null=NullValue]%`

Related Documentation

This section lists the documentation that webMethods provides with webMethods OFTP Adapter. In addition, it lists documentation provided with webMethods Trading Networks, webMethods Integration Server, webMethods Developer and rvs software that you might also find useful.

webMethods OFTP Adapter Documentation

The following table lists other manuals that webMethods provides with webMethods OFTP Adapter.

Refer to this book…	For…
webMethods OFTP Adapter: Installation Guide	Describes how to install the OFTP Adapter. This book is for Server Administrators.

T-Systems rvs Documentation

The following table lists the documentation that **T-Systems Enterprise Services GmbH** provides with the rvs software. All the guides mentioned below can be downloaded from the rvs homepage at https://servicenet.t-systems.de/tsi/en/278798/Home/Business-Integration/rvs/rvs-Home (select the link Download –> Manuals).

Refer to this book…	For…
Installation Manual for rvsXP, rvsX	Detailed information concerning rvs program environment, the required prerequisites and the installation of rvs
Operator Manual for rvsXP, rvsX, rvs400	Detailed information concerning configuration of rvs
Reference Manual for rvsXP, rvsX, rvs400	Common information on how to use rvs software
User Manual for rvsXP	Information on how to work with rvsNT software
User Manual for rvsX	Information on how to work with rvsX software
rvs Client Server V1.0, User Manual	Information on installation and configuration of rvs middleware

Additional Information

You can find additional information about webMethods products at the locations described below.

webMethods Central Documentation Directory

During product installation, you can download the webMethods product documentation to a single directory called "_documentation." This directory is located by default under the webMethods installation directory.

webMethods Advantage Web Site

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

- **Product Documentation.** You can find documentation for all webMethods products on the webMethods Bookshelf http://advantage.webmethods.com/Bookshelf.

- **Troubleshooting Information.** The webMethods Knowledge Base http://advantage.webmethods.com/knowledgebase provides troubleshooting information for many webMethods products.
- **Documentation Feedback.** To provide feedback on webMethods documentation, go to the Documentation Feedback Form on the webMethods Bookshelf.

Software AG Developer Community

Additional articles, demos, and tutorials are available on the webMethods portion of the Software AG Developer Community. The various Developer Communities feature technical information, useful resources, and online discussion forums, moderated by Software AG professionals, to help you do more with webMethods technology.

1 Product Overview

Technical Architecture Overview

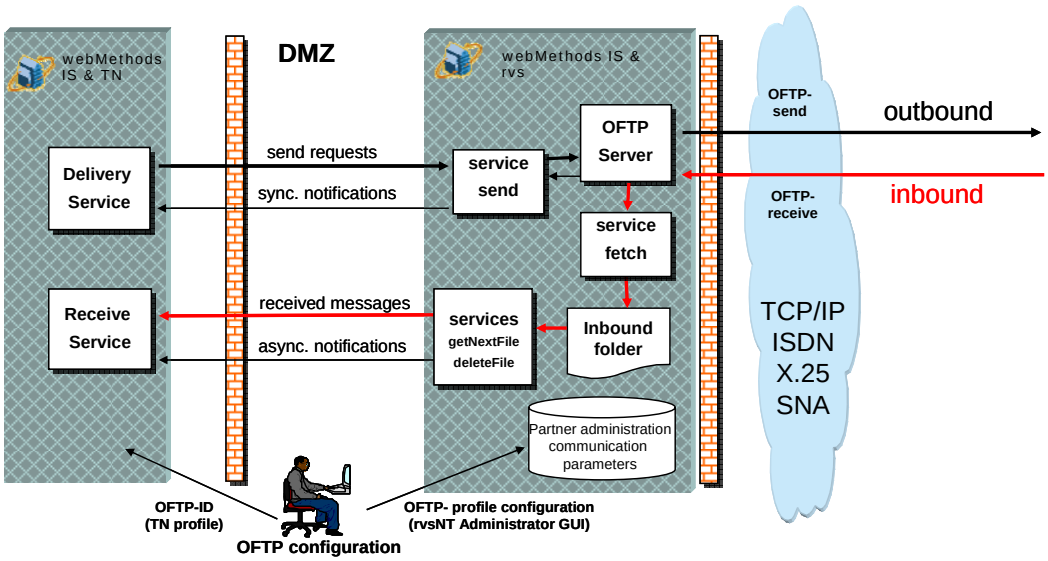
The OFTP Adapter is built on top of webMethods Trading Networks and webMethods Integration Server and is provided as two packages, the OFTP Gateway package (`OFTP`) and the OFTP TN package (`OFTPforTN`).

It uses rvsNT (or rvsX) and rvs Client Server as third party OFTP client and server to send and receive documents to and from trading partners. rvsNT (or rvsX) and rvs Client Server is a software requirement to run the OFTP Adapter (see *webMethods OFTP Adapter Installation Guide*).

The OFTP Adapter provides an outbound delivery service for sending outbound documents from Trading Networks. Inbound documents as received by the rvs server are stored in the file system of the rvs server first. A scheduled service on the OFTP Adapter side polls for these files.

The OFTP Adapter uses the rvs client Java API (rvs middleware of rvs Client Server component). The benefits are to be widely version independent of the rvs server and to support distributed rvs installations in future versions of OFTP Adapter.

Your own user account (**My Enterprise**) and the user accounts of your OFTP partners are represented by Trading Networks profiles. For each profile, you must specify an OFTP address as the **OFTP ID** external ID type. During initialization of the package a new External ID Type **OFTP ID** is automatically created.



OFTP Adapter Components Overview

To support configurations using a DMZ (demilitarized zone), webMethods OFTP Adapter consists of two parts:

- An **OFTP Gateway** service on the webMethods Integration Server installation in the DMZ. This OFTP Gateway service is Trading Networks-aware and contained in the `OFTP` package. You configure the OFTP Gateway service using the Integration Server Administrator.
- **Delivery and receive services** in the Integration Server running Trading Networks. These services are in the `OFTPforTN` package. The delivery and receive services are behind the firewall and use remote invoke to access the OFTP Gateway service in the DMZ. These services also use Trading Networks partner profiles. You configure the OFTP partner information in profiles using the Trading Networks Console.

You can install the gateway configuration together with Trading Networks on one machine if there is no need for a DMZ.

OFTP Adapter Folder Structure

The OFTP Adapter temporarily stores the inbound messages and notifications received from the OFTP partner stations into an inbound folder before they are moved to Trading Networks.

The OFTP Adapter also temporarily stores outbound messages into an outbound folder.

The OFTP Gateway contains the following folders:

Folder name	Description of contents
`inbound`	contains - inbound messages (oftpMessageIn) - notifications and error messages (oftpStatusIn) Note: *.tmp files are also stored here, but the adapter collects only *.xml messages.
`outbound`	contains outbound messages (oftpMessageOut)

OFTP Adapter Feature Overview

Features

The webMethods OFTP Adapter supports the following:

- Sending and receiving of data using the OFTP protocol
- Initiate the fetching of data from the OFTP partner station.
- Inbound processing of the following formats
  - ANSI X.12
  - EDIFACT
  - EANCOM
  - VDA
  - ODETTE
  - IDOC
  - XML
  - HTML
  - Binary format
  - Text format
- Processing of status and error messages
- Access to ISDN and TCP/IP networks

Trading Networks Document Types

- When you install the webMethods OFTP Adapter, the following Trading Networks XML document types are installed:

Document Type	Description
oftpMessageIn	Contains the inbound OFTP message (with base64 encoded payload and a message format identifier).
oftpMessageOut	Contains the outbound OFTP message (payload data and receiver parameters).
oftpStatusIn	Contains inbound notifications and error messages.
oftpPayload	Contains the extracted content out of an **oftpMessageIn** with unknown format.
oftpStatusOut	Contains information on positive or negative EERP.

Introduction to T-Systems rvs

The abbreviation rvs stands for the German word **R**echner-**V**erbund-**S**ystem. The rvs computer communication system is a well established base service for electronic data interchange, EDI.

Task of the system is, to guarantee transmission of electronic data between heterogeneous computer platforms using different network protocols.

rvsXP / **rvsX** provide an efficient and reliable transport service for both standardized EDI message types and files of any format or content. You can get only data files which have been provided explicitly by rvs. This means that rvs does not provide an unauthorized access to remote data files.

The **Client Server** solution extends rvs to become a server, with the services file transfer, file conversion, file encryption and rvs administration. These services can be used by clients distributed over the network.

For a detailed description of the features of the rvs software please refer to the “User Manual for rvsXP”, “User Manual for rvsX” and “rvs Client Server User Manual” documentation.

2 How OFTP Messages are Processed

Define Trading Networks Profiles for the OFTP Adapter

Use My webMethods to define sender and receiver profiles.

Defining the Sender Profile

Use the following procedure to define the sender (**My Enterprise)** profile.

Define the sender profile

1. In My webMethods: **Administration > Integration > B2B > Partner Profiles**
2. Click **My Enterprise**
3. Click **External IDs**
4. Click **Add ID**... and select **OFTP ID** as **ID Type**.
5. Enter the OFTP ID of your enterprise In the **ID Value**.

Caution:
The value for the **OFTP ID** must be identical to the rvs **Station-Id** configured within the rvsXP Administrator GUI (select View -> Stations menu, **Station-Id** field).

Defining the Receiver Profile

Use the following procedure to define the receiver.

Define the Receiver Profile

1. In My webMethods: **Administration > Integration > B2B > Partner Profiles**
2. Search the partner and click its **Corporation Name.**
3. Click **External IDs**
4. Click **Add ID**... and select **OFTP ID** as **ID Type**
5. Enter the OFTP ID of your OFTP partner station in the **ID Value**.
6. Click into the **Value** field in the same line and enter the value of the **OFTP ID** of your OFTP partner station.

Caution:
The value for the **OFTP ID** must be identical to the rvs **Station-Id** configured within the rvsXP Administrator GUI (select View -> Stations menu, **Station-Id** field).

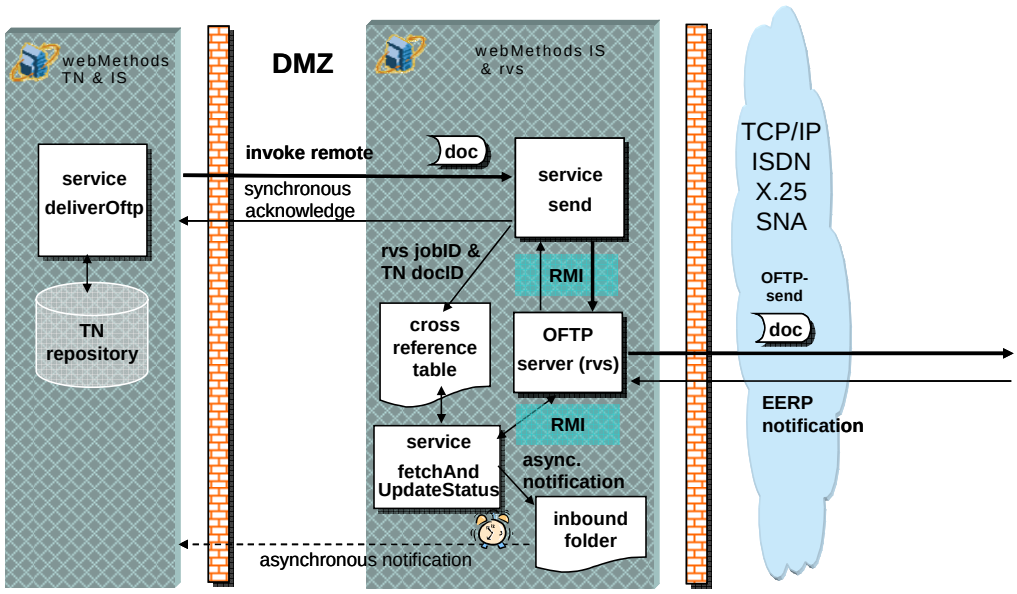
7. If the option of polling the partner station is to be used (see "Polling of Partner Stations")
    - Select the **Extended Fields** tab of your partner's profile.
    - Locate under **Field Groups** the group **OFTP**
    - Set the value of **activatePartner** to **Yes**

Sending Outbound Messages

Overview

The OFTP Adapter provides a set of services for sending of documents by OFTP and for receiving of inbound notifications from the OFTP partner station.

Overview outbound message processing



The following illustrates how the OFTP Adapter processes outbound messages.

- The delivery service **wm.oftp.tn:deliverOftp** must be called for sending OFTP messages. The service takes the document you want to send and gets the OFTP parameters from the Trading Networks partner profiles.
- The **wm.oftp.gateway.Outbound:send** service of the gateway invokes the rvs server to send the message to the OFTP partner station.
- The service **wm.oftp.gateway.Gw:fetchAndUpdateStatus** queries the rvs server for the status of sent jobs.

Sending of OFTP Messages

To send OFTP messages you have to process the following steps:

1. Configure the profile of the receiving partner in Trading Networks if not already done.

2. Call the service **wm.oftp.tn:deliverOftp** which converts the BizDoc into an OFTP message (document type **oftpMessageOut**) and sends it to your OFTP partner by using the OFTP Gateway.

   The **wm.oftp.gateway.Outbound:send** service of the gateway pushes the documents to the rvs server and invokes a send process.
   This service additionally creates an entry in a cross-reference table used to relate the rvs job-ID to the Trading Networks internal ID (doc-ID).

   The OFTP Adapter sets the Trading Networks user status to **OFTP queued**.

   If there occurs a problem when addressing the rvs middleware an OFTP message **oftpStatusIn** with statusType '**error'** is created.
   In this case the user status within Trading Networks is set to **OFTP send failed.**
   Additionally the problem is added as entry into the Server Log of the Integration Server (for further details see "Error Handling").

Caution: To be able to send OFTP messages the rvs Monitor and rvs middleware must be started.

Note: When the user status within Trading Networks is set to **IGNORED**, please check the consistency of configuration parameters.

Note: When the creation of a job entry in rvs fails an exception is created. The user is responsible for handling this exception in Trading Networks.

Receiving of Inbound Notifications

To process inbound notifications from your OFTP partner station you have to process the following steps:

1. Call the service **wm.oftp.gateway.Gw:fetchAndUpdateStatus**. This service queries the rvs database for the status of sent jobs as listed in the cross-reference table.

   For each job in rvs with status 'processing' it generates a notification message **oftpStatusIn** with statusType **transmitStatus** and saves it in the inbound folder (for the structure of **oftpStatusIn**, see "Structure of Sent Message").

   For each job in rvs with status 'deleted' it generates a notification message **oftpStatusIn** with status type **nonTransmitStatus** and saves it in the inbound folder.

   For each **completed job** (rvs job status 'finished') it
   - generates a notification message **oftpStatusIn** with statusType **positiveEERPin** and saves it in the inbound folder
   - deletes the entry in the reference table
   - deletes the entry of the rvs job-ID at the rvs server

Note: The service **wm.oftp.gateway.Gw:fetchAndUpdateStatus** has to be configured as a scheduled service within webMethods Integration Server.
The time interval detemines when the OFTP Adapter will fetch all new files available.

2. In Trading Networks you have to call the service **wm.oftp.tn:receiveOftp**. The service takes the **oftpStatusIn** documents and stores it in the Trading Networks database.

   The sender and receiver information for Trading Networks is gained out of the message header automatically.

   After receiving an inbound notification, the OFTP Adapter then sets the actual value of the user status of the related **oftpMessageIn** within Trading Networks document to
   - **OFTP transmit status** (when value of statusType is **transmitStatus**)
   - **OFTP non-transmit status** (when value of statusType is **nonTransmitStatus**)
   - **OFTP positive-EERP** (when value of statusType is **positiveEERPin**)

Note: This service has to be configured as a scheduled service within webMethods Integration Server.
The time interval determines when the OFTP Adapter will fetch all new files available.

The **Alias** name of the remote Integration Server has to be set via a webMethods Developer map step at the beginning of the service **wm.oftp.tn:receiveOftp** if the OFTP Gateway does not reside on the local machine.

3. The OFTP Adapter creates a Trading Networks relationship between the original **oftpMessageOut** message sent to the OFTP partner station and the particular **oftpStatusIn** notification messages.

Status Summary

The following table provides an overview on rvs status, value of status type and corresponding Trading Networks user status.

Document type / statusType	Trading Networks user status	rvs status	Remark
oftpMessageOut / --	OFTP queued	various	File is successfully transferred to rvs and rvs has created a send job; temporary status
oftpStatusIn / error	OFTP send failed	---	Problem when addressing rvs middleware
oftpStatusIn / transmitStatus	OFTP transmit status	processing	File transmitted, waiting for EERP temporary status
oftpStatusIn / nontransmitStatus	OFTP non-transmit status	deleted	Send job deleted (for example by rvs administrator) final status
oftpStatusIn / positiveEERPin	OFTP positive-EERP	finished	Positive EERP received; job finished final status *)

*) As the OFTP standard does not define that sending of an EERP notification is mandatory, it can not be guaranteed that all OFTP partner stations will send this message. That means the final status depends on your arrangement with your OFTP partner.

Input Parameters of the Service wm.oftp.tn:deliverOftp

The following specifies the input parameters for the **wm.oftp.tn:deliverOftp** service

1. **Bizdoc** (BizDocEnvelope) with content and address information out of Trading Networks Profiles. This is the message to be sent.
2. **Alias** name of the remote Integration Server if the OFTP Gateway does not reside on the local machine.

Input Parameters of the Service wm.oftp.tn:receiveOftp

The following specifies the input parameters for the **wm.oftp.tn:receiveOftp** service

1. **Alias** name of the remote Integration Server if the OFTP Gateway does not reside on the local machine.

Structure of Sent Message

The following describes the structure of a sent message (Trading Networks document type **oftpMessageOut**).

```
<?xml version="1.0" encoding="UTF-8"?>
<oftpMessageOut>
<sender>
  <stationID>LOC</stationID>
  <stationIDalias>   </stationIDalias>
</sender>
<receiver>
  <stationID>OBELIX</stationID>
</receiver>
  <documentID>c0a80102f70d37f900000064</documentID>
  <VDSN>PRICAT.EDI</VDSN>
  <inputFormat>   </inputFormat>
  <maxRecSize>       </maxRecSize>
  <inputCode>    </inputCode>
  <outputCode>    </outputCode>
  <data>      </data>
</oftpMessageOut>
```

Structure of the Inbound Notifications

The following describes the structure of inbound notifications (Trading Networks document type **oftpStatusIn,** value of statusType is **positiveEERPin**, **transmitStatus** or **nonTransmitStatus**).

The **oftpStatusIn** notification contains the references for the rvs job-ID and the Trading Networks internal ID (doc-ID).

Structure of inbound notification oftpStatusIn, status type positiveEERPin

```
<?xml version="1.0" encoding="UTF-8"?>
<oftpStatusIn>
<sender>
  <stationID>OBELIX</stationID>
</sender>
<receiver>
  <stationID>LOC</stationID>
  <stationIDalias>   </stationIDalias>
</receiver>
<statusType>positiveEERPin</statusType>
<positiveEERPin>
<!--inbound positive End to End Response-->
  <documentID>c0a80102f70d37f900000064</documentID>
  <rvsJobID>631</rvsJobID>
  <rvsJobStatus>finished (3)</rvsJobStatus>
  <timeOfLastStateChange>2003/03/0711:39:29</timeOfLastStateChange>
</positiveEERPin>
</oftpStatusIn>
```

Structure of inbound notification oftpStatusIn, status type transmitStatus

```
<?xml version="1.0" encoding="UTF-8"?>
<oftpStatusIn>
<sender>
  <stationID>OBELIX</stationID>
</sender>
<receiver>
  <stationID>LOC</stationID>
  <stationIDalias>   </stationIDalias>
</receiver>
<statusType>transmitStatus</statusType>
<transmitStatus>
  <documentID>c0a80102f70d37f900000064</documentID>
  <rvsJobID>631</rvsJobID>
  <rvsJobStatus>forwarded</rvsJobStatus>
```

```
  <timeOfLastStateChange>2003/03/0711:39:29</timeOfLastStateChange>
</transmitStatus>
</oftpStatusIn>
```

Structure of inbound notification oftpStatusIn, status type nonTransmitStatus

```
<?xml version="1.0" encoding="UTF-8"?>
<oftpStatusIn>
<sender>
  <stationID>OBELIX</stationID>
</sender>
<receiver>
  <stationID>LOC</stationID>
  <stationIDalias>   </stationIDalias>
</receiver>
<statusType>nonTransmitStatus</statusType>
<nonTransmitStatus>
  <documentID>c0a80102f70d37f900000064</documentID>
  <rvsJobID>631</rvsJobID>
  <rvsJobStatus>deleted</rvsJobStatus>
  <timeOfLastStateChange>2003/03/0711:39:29</timeOfLastStateChange>
  <reason>    </reason>
</nonTransmitStatus>
</oftpStatusIn>
```

Structure of an Error Notification

The following describes the structure of inbound error notification (Trading Networks document type **oftpStatusIn,** value of statusType is **error**).

Structure of error notification oftpStatusIn

```
<?xml version="1.0" encoding="UTF-8"?>
<oftpStatusIn>
<sender/>
<receiver/>
<statusType>error</statusType>
<error>
<!--OFTP error status-->
  <documentID>c0a80102f70d37f900000064</documentID>
  <functionName>Transfer.processPositiveEERPout()</functionName>
  <class>OFTP Gateway</class>
  <code>-1</code>
  <text>              </text>
  <timestamp>2003/03/0711:39:29</timestamp>
```

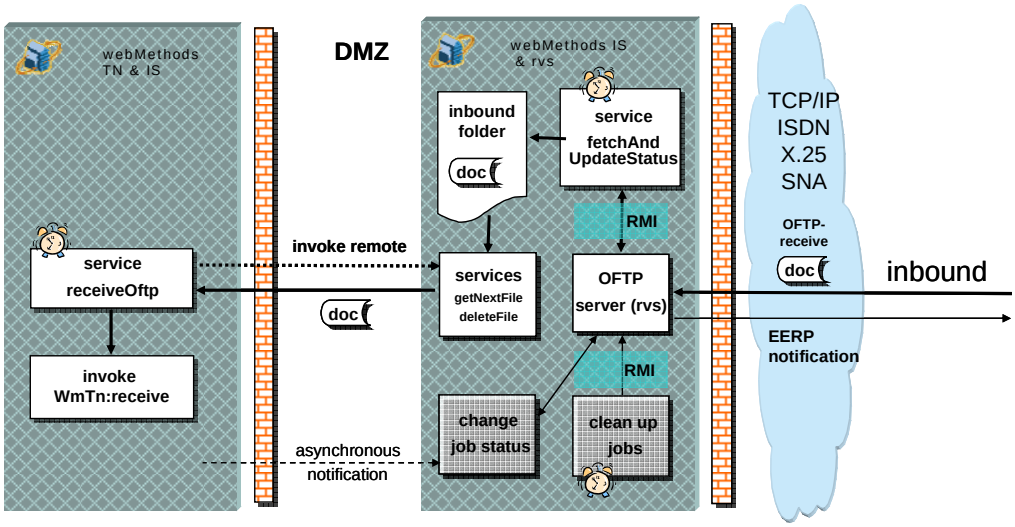
```
</error>
</oftpStatusIn>
```

Receiving Inbound Messages

Overview

The OFTP Adapter provides a set of services for receiving of documents by OFTP.

Overview inbound message processing



The following illustrates how the OFTP Adapter processes inbound messages.

- The scheduled service **wm.oftp.tn.Gw:fetchAndUpdateStatus** fetches the files received from the OFTP server and checks the format of the received payload data.

  For each file the service creates an **oftpMessageIn** document and saves the document as XML file in the inbound folder.

- The service **wm.oftp.tn:receiveOftp** moves the data from the OFTP Gateway to Trading Networks.

Receiving of OFTP Messages

To receive messages by OFTP you have to process the following steps:

1. Configure the profile of the receiving partner in Trading Networks if not already done.

2. Call the service **wm.oftp.gateway.Gw:fetchAndUpdateStatus** which fetches the files sent by the partner OFTP station to your OFTP server.

   For each file the **wm.oftp.gateway.Gw:fetchAndUpdateStatus** service parses the payload data and tries to recognize its format.

   The following formats are identified:
   - ANSI X.12
   - EDIFACT
   - EANCOM
   - VDA
   - ODETTE
   - IDOC
   - XML
   - HTML

   All other files are classified as format UNKNOWN.

   For each file the service creates an **oftpMessageIn** message type, adds a <format> identifier and saves the document as XML file in the inbound folder for further processing.

Note: This service has to be configured as a scheduled service within webMethods Integration Server.
The time interval determines when the OFTP Adapter will fetch all new files available.

Caution To be able to receive OFTP messages the rvs Monitor and rvs middleware must be started

3. For receiving messages in Trading Networks, you should create a client that invokes the service **wm.oftp.tn:receiveOftp** to retrieve the messages from the inbound folder.
   The **wm.oftp.tn:receiveOftp** service stores the message in the Trading Networks database.

   The **wm.oftp.tn:receiveOftp** service automatically pulls the sender and receiver information for Trading Networks out of the message header.

   Next the service **wm.oftp.tn:receiveOftp** parses the message for the content:

   Known format –
   The OFTP Adapter routes contents of a known format (e.g. ANSI X.12, EDIFACT, EANCOM, IDOC) as 'edidata' to the service **wm.tn:receive**.

If the format is XML or HTML the service **routeXml** is used to send the base64 encoded content to Trading Networks.

Unknown format –

The OFTP Adapter stores contents of an unknown format (e.g. VDA, ODETTE, UNKNOWN) as the TN document type **oftpPayload**. The **oftpPayload** document contains the OFTP IDs of sender and receiver, the format identifier UNKNOWN, VDA or ODETTE and the base64-encoded content.

Note: This service has to be configured as a scheduled service within webMethods Integration Server.
The time interval determines when the OFTP Adapter will fetch all new files available.

The **Alias** name of the remote Integration Server has to be set via a webMethods Developer map step at the beginning of the service **wm.oftp.tn:receiveOftp** if the OFTP Gateway does not reside on the local machine.

Note: A set of Trading Networks processing rules has to be defined to process the resulting documents.

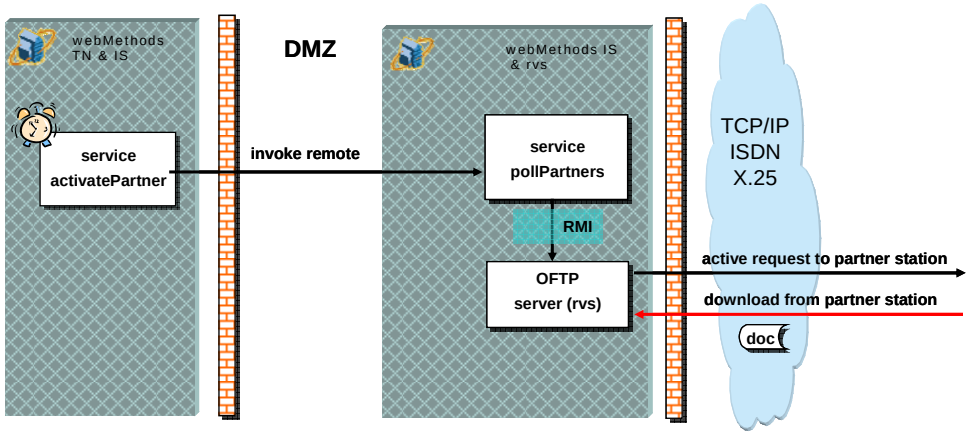
4. The OFTP Adapter creates a Transactions relationship between the original message retrieved from the `Inbound` folder and the extracted, whether the content part is of known or unknown format.
   To see relationships go to **My webMethods > Monitoring > Integration > B2B > Transactions**, search for a transaction and click on the icon in the column **Related Documents**

3. The OFTP Adapter sets the Trading Networks user status of the original message retrieved from the `Inbound` folder to **OFTP received.**

Polling of Partner Stations

Instead of listening for inbound documents, the rvs server can also poll OFTP partner stations for new document. Here the rvs server itself activates a connection to the partner station and downloads available documents.

Poll partner station



To download new documents from the OFTP partner station you have to process the following steps:

1. Call the service **wm.oftp.tn:pollPartners** available at Trading Networks to trigger the rvs server for activating a connection to those partner stations that are configured to be polled and downloads the available documents to the inbound directory of the rvs server.

Note: This service has to be configured as a scheduled service within webMethods Integration Server.

Caution: When this configuration for polling the rvs partner stations is to be used, your trading partner has to specify the behavior for sending messages as follows:
At the rvsXP Monitor for his remote station (i.e. for your system) he has to deselect the field “Active Connection Setup” (via **View** – **Stations** – **Edit** Station, click on **Advanced** for the used Connection)

2. The transport is done by using the services **wm.oftp.gateway.Gw:fetchAndUpdateStatus** and **wm.oftp.tn:receiveOftp** as described in "Receiving of OFTP Messages".

Using the Enhanced EERP (End-to-End-Response) Functionality

The OFTP protocol uses the EERP functionality to inform the sender of a file that the file has arrived at the receiving OFTP partner station. By default, the rvs partner station immediately generates this EERP acknowledgement. Optionally for enhanced EERP functionality, the acknowledgment is not sent until release is done within a release command of the receiving application.

To use the extended EERP functionality you have to process the following steps:

1. Call the services **wm.oftp.gateway.Gw:fetchAndUpdateStatus** to fetch the files sent by the partner OFTP station to your OFTP server and the service **wm.oftp.tn:receiveOftp** to retrieve the messages from the Inbound folder to Trading Networks.
   For a detailed description of these services, see "Receiving of OFTP Messages".

Note:
To use the enhanced EERP functionality at the rvsXP Administrator GUI (View -> Stations menu , Edit – Odette Parameters (Advanced) tab) for the **EERP OUT** parameter the value `HOLD/IMMEDIATE` or `HOLD/NORMAL` has to be selected

2. Define a processing rule within Trading Networks to submit the information on received inbound messages to the backend system. The backend application then has to return a status to the Trading Networks side process logic.

3. Create the Trading Networks document **oftpStatusOut** with identifier **positiveEERPout** if the status is positive or **negativeEERPout** if the status is negative (see "Structure of Messages").

   The rvsJobID must have the same value as available within the original inbound message. Additionally the documentID must correspond to the Trading Networks internal ID (doc-ID) of the inbound message.

4. If the notification status is positive, you have to call the service **wm.oftp.tn.deliverPositiveEERPout** to release the hold EERP notification for this job.
   In case of a negative notification, you have to call the service **wm.oftp.tn.deliverNegativeEERPout** to delete the related EERP hold job.

   The service **wm.oftp.gateway.outbound:send** is invoked for sending the notification to the OFTP partner station.

Input Parameters of the Services

Input parameters of service wm.oftp.tn:receiveOftp

The following specifies the input parameters for the **wm.oftp.tn:receiveOftp** service

1. **Alias** name of the remote Integration Server if the OFTP Gateway does not reside on the local machine.

Input parameters of service wm.oftp.tn:pollPartners

The following specifies the input parameters for the service **wm.oftp.tn:pollPartners:**

1. If the OFTP Gateway resides on a remote Integration Server, the **Alias** name and the **OFTP ID** have to be specified.

Input parameters of services deliverPositiveEERPout and deliverNegativeEERPout

1. **Bizdoc** (TN document **oftpStatusOut**)
2. **Alias** name of the remote Integration Server if the OFTP Gateway does not reside on the local machine.

Structure of Messages

The following describes the structure of an inbound message (Trading Networks document type **oftpMessageIn**) and the messages used for enhanced EERP functionality (Trading Networks document type **oftpStatusOut** with identifier **positiveEERPout** or **negativeEERPout**).

Structure of oftpMessageIn

```
<?xml version="1.0" encoding="UTF-8" ?>
<oftpMessageIn>
<!-- inbound OFTP message -->
  <sender>
    <stationed>OBELIX</stationID>
  </sender>
```

```
  <receiver>
    <stationID>LOC</stationID>
    <stationIDalias>   </stationIDalias>
  </receiver>
  <rvsJobID>597</rvsJobID>
  <VDSN>PRICAT.EDI</VDSN>
  <sfidDate>2003/01/31 10:12:48</sfidDate>
  <compression>false</compression>
  <encryption>false</encryption>
  <filedate>2003/01/31 10:12:48</filedate>
  <filesize>103329</filesize>
  <filename>PRICAT.EDI</filename>
  <format>EDIFACT</format>
  <maxRecSize>       </maxRecSize>
  <receiveAttemps>      </receiveAttempts>
  <data>      </data>
</oftpMessageIn>
```

Structure of oftpStatusOut

```
<?xml version="1.0"?>
<oftpStatusOut>
  <positiveEERPout> (respectively <negativeEERPout>)
    <documentID>c0a80102f70dfecf0000128a</documentID>
    <rvsJobID>105</rvsJobID>
  </positiveEERPout>
</oftpStatusOut>
```

Error Handling

The webMethods OFTP Adapter uses the common principles of Integration Server and Trading Networks error handling. It uses the standard routines for error logging.

Every OFTP Adapter error is written into the Integration Server “Server” Log.

Since the OFTP Adapter could reside on a remote Integration Server, the OFTP Gateway errors result in an error message (document type **oftpStatusIn** with statusType ‘error’), which is stored in the inbound folder.

With the next receiving schedule, the error message is transferred to Trading Networks together with messages and notifications.

3 Restrictions

Clustering

The webMethods OFTP Adapter is not usable in a clustered webMethods environment. It is not possible to run the adapter parallel on two or more Integration Servers in a load-balanced scenario.
The OFTP Adapter only supports fail-over configurations.

In these configurations the adapter is installed on two integration servers (production and fail over system), but there is only one running instance on the active Integration Server.

Large Document Handling

If you send outbound documents via OFTP Adapter there is full support for large document handling of the contents within Trading Networks. By the guaranteed delivery mechanism, the documents are submitted to the OFTP Gateway service.

If documents are received by the OFTP Gateway, they are held in memory until they are stored in the Trading Networks database (document type **oftpMessageIn**).
During extracting the content they reside in memory until they are submitted to **tn:receive** service (known format) or stored as document type oftpPayload in the Trading Networks database (unknown formats). Therefore sufficient memory has to be provided by the hardware.

Common Restrictions

No transaction protection (monitoring) along the inbound/outbound process within OFTP Gateway services and between Trading Networks and T-Systems OFTP Server.