

MailBox X.400

MessageGate for OpenMS V5.7

User's Guide

V2.6



All products or services mentioned in this document are identified by the trademarks or registered trademarks of their respective companies or organizations. Telekom Deutschland GmbH claims no responsibility for specifying which trademarks are owned by which companies or organizations.

The information in this guide is subject to change without notice. The information does not represent a commitment on the part of Telekom Deutschland GmbH. Telekom Deutschland GmbH is not responsible for any errors that may appear in this manual. It is illegal to copy this document except with specific permission in a license or non-disclosure agreement. The manual or parts of the manual may not be reproduced or transmitted in any form or by any means, electronically or mechanically, including photocopying and recording, for any purpose, without the express written permission of Telekom Deutschland GmbH. No parts of this publication may be transcribed, stored in a retrieval system, or translated into any language without the prior written consent of Telekom Deutschland GmbH.

Copyright © 2023

**Telekom Deutschland GmbH**  
**BusinessMail X.400**  
**Postfach 107300**  
**68165 Mannheim**

# Index

<b>0</b>	<b>PREFACE</b>	<b>1</b>
0.1	Please send any comments to the following E-mail address	1
0.2	The following conventions are used in this guide	1
0.3	The following abbreviations are used in this guide	2
0.4	New in V2.6	3
<b>1</b>	<b>INTRODUCTION TO MESSAGEGATE FILE INTERFACE</b>	<b>4</b>
1.1	With MessageGate File Interface you can:	4
1.2	X.400 – an International Standard	4
1.3	The motives in implementing MessageGate	4
<b>2</b>	<b>INTERFACE DESCRIPTION</b>	<b>6</b>
2.1	Overview	6
2.2	MessageGate User's directory	11
2.3	The Message File	13
2.3.1	The Message Header	13
2.3.2	Address format	16
2.3.3	Message Content	18
2.3.4	S/MIME secured content	21
2.4	The Forwarding of a message is not supported	24
2.5	Transmission Set File Format	25
2.5.1	The central EDI functionality	25
2.5.2	The Content of Transmission Set Files	27
2.6	Closed User Group (CUG)	27
2.7	The Status Report	28
2.7.1	General	28
2.7.2	Request a Status Report	28
2.7.3	Status Report Syntax (readable structure)	30
2.7.4	Syntax of Status reports (CSV structure)	35
2.8	Send Receipt Notification	37
2.9	Communication and Trading Relation Profile	38
2.9.1	General	38
2.9.2	X.400 Reports	40
2.9.3	X.400 Header Information	41
2.9.4	X.400 Body parts	41
2.9.5	Encoding of binary data	42
2.9.6	File format (Format of output)	42
2.9.7	Request Status Report in WebConfig	42
2.9.8	Configure automatically generated Status Report	42
2.9.9	EDI Relation	43
2.9.10	SMTP Filter	44
2.9.11	Web Service of WebConfig	45
<b>3</b>	<b>ACCESS VIA SFTP (SSH)</b>	<b>46</b>
3.1	General Information	46

3.2	Features to note	46
3.3	Recommended SFTP Communication modules	47
3.3.1	Using Microsoft® Windows 32 Bit Operating systems	47
3.3.2	Using Microsoft® Windows 64 Bit Operating systems	53
3.3.3	Using Linux and UNIX Operating systems	54
3.3.4	Using Apple Mac OS X	54
3.3.5	Using other operating systems	55
4	ACCESS VIA HTTPS/WEBDAV	56
4.1	General Information	56
4.2	Features to note	57
4.3	Recommended WebDAV Clients	57
4.3.1	Using Microsoft® Windows 32 Bit Operating systems	57
4.3.2	Using Microsoft® Windows 64 Bit Operating systems	65
4.3.3	Using Linux and UNIX Operating systems	65
4.3.4	Apple iOS	66
4.3.5	Using other operating systems	68
5	ACCESS VIA HTTPS/WEB SERVICES	70
5.1	General Information	70
5.2	Features to note	71
5.3	Web Service API	71
5.3.1	The Web Service profile v1	72
5.3.2	The Web Service profile v2 (2a)	74
5.3.3	The Web Service profile v3 (3a)	83
5.4	REST based modules used for test purposes	85
6	AS2 AND MESSAGEGATE	86
6.1	General Information	86
6.2	Differences between File Interface and AS2 users	88
7	SMTP MTA AND MESSAGEGATE	96
7.1	General Information	96
7.2	Difference between File Interface and SMTP MTA users	96
8	IMPLEMENTATION OF MESSAGEGATE SOLUTION	104
8.1	Using Standard E-Mail Clients	104
8.1.1	Test with Outlook Express for older Windows OS	104
8.1.2	Test with Mozilla Thunderbird	104
8.1.3	Test with Microsoft Live Mail for newer Windows OS	105
8.2	Designing a MessageGate solution	106
	APPENDIX A X.400 ADDRESS ELEMENTS	108
	APPENDIX B: ERROR CODES	110
B1.	Error codes of MessageGate Poller process:	110
B2.	MessageGate Error codes	111
B3.	MTA Error codes (Non Delivery Notification)	118
B4.	X.400 User Agent Error codes (Non Receipt Notification)	126

B5. Mapping rules NDN to DSN	127
<b>APPENDIX C: EXAMPLES FOR MESSAGES AND REPORTS</b>	<b>130</b>
C1. Delivered Message with text attachment	130
C2. Delivered Message with binary attachment	130
C3. Delivered Message with Multiple attachments	131
C4. Delivered Message with Multi-Recipients	131
C5. Submitted Message and no Report Request	132
C6. Submitted Message with Report Request	133
C7. Submitted Message with Multi-Recipients	134
C8. Delivered Signed Message	135
C9. Delivered Encrypted Message	136
C10. Transmission Set with two Interchanges	137
C11. Status Report without History	138
C12. Status Report with History	140
C13. Status Report for a selected Order-ID	141
C14. Status Report for a selected Message-ID	142
C15. Status Report for denied Messages	142
C16. Report for submitted message (Multi-Recipients)	143
<b>APPENDIX D: CHARACTER SETS</b>	<b>146</b>



# 0 Preface

## The Product

This guide contains information about MessageGate, a file interface to the *BusinessMail X.400* MailBox service, to send and receive messages based on the ITU/ISO X.400 standard.

## The Customer

This new interface will help those customers, who do not have the possibility of using a standard X.400 Mail client (a so-called P7 Remote User Agent) for communicating with X.400 partners. MessageGate is available in addition to the existing Batch User Agent (BUA) file interface, which it will replace in the long term.

## Additional Documents

- See also Batch User Agent Reference Guide
- RFC 2822, RFC 1521

## Registered trademarks

Microsoft® and Windows TM are trademarks of the Microsoft Corporation. Other product and company names that are mentioned in this manual are included to identify the products. They may be trademarks or registered trademarks for the companies they belong to.

## 0.1 Please send any comments to the following E-mail address

[helpdesk.businessmailx400@telekom.de](mailto:helpdesk.businessmailx400@telekom.de), Subject: MessageGate User Guide

## 0.2 The following conventions are used in this guide

This typeface	Indicates contents of messages/headers submitted over the MessageGate file interface
This typeface	Indicates MessageGate interface parameters, elements or values

## 0.3 The following abbreviations are used in this guide

API	Application Programming Interface
AS2	Applicability Statement 2 - Standard (EDIINT) for B2B communication via Internet (RFC 4130)
BUA	Batch User Agent – File Interface of <i>BusinessMail X.400</i> MailBox service accessing a customer's mailbox
BP14	Body part 14 – Bilaterally defined (binary) ITU X.400 Standards (1984)
CA	Certificate Authority, an entity that issues digital certificates.
CR	Carriage Return
CSV	comma-separated values - a simple text format for a database table where a record in the table is one line of text file
CUG	Closed User Group, data transfer only valid between members of this group
DDA	Domain Defined Attributes (X.400 address elements)
DN / NDN	X.400 Delivery Notification (report) or Non-Delivery Notification (report) generated by X.400 MTA
EDI	Electronic Data Interchange, a particular set of standards for computer-to-computer exchange of information
EDIFACT	Standard for Electronic Data Interchange – ISO 9735
FTBP	FTAM (File Transfer, Access, and Management) body part, used to transfer binary content including file information (e.g., file name, type of content etc.).
FTP	File Transfer Protocol (RFC 959)
GDI	Global Domain Identifier (X.400 Terminology)
GLN	Global Location Number (registered EDI address)
HTTP(S)	Hypertext Transfer Protocol. HTTP is the protocol to transfer data over the web.
ILN	International Location Number (registered EDI address)
ITU-T	International Telecommunication Union - Telecommunication
LF	Line Feed
MB	Megabyte – 1 000 000 bytes – definition recommended by the International System of Units (SI)
MDN	Message Disposition Notification – Report type used for SMTP (RFC 3798) and AS2 (RFC 4130)
MIME	Multipurpose Internet Mail Extensions (RFC 2045, RFC 2046, RFC 2047, RFC 4288, RFC 4289 and RFC 2049)
MPLS	Multiprotocol Label Switching – IP Backbone using a labeling mechanism to implement a VPN
MS	(X.400) Message Store
MTA	(X.400) Message Transfer Agent



P2/P22/P35	X.400 message types
P7	X.400 protocol standard describing how an X.400 Client communicates with a MS (Message Store)
PEDI	Special X.400 EDI message type defined in X.435 (equivalent to P35)
RFC	Request for Comment – Internet Standards Track
RN / NRN	X.400 Receipt Notification (report) or Non-Receipt Notification (report) generated by X.400 user agent (client)
SFTP	Secure File Transfer Protocol (Part of the Secure Shell Protocol suite, RFC 4259 ff)
S/MIME	Secure/ Multipurpose Internet Mail Extensions (V3.2, RFC 5751), MIME Security Extension (Signature/Encryption)
SMTP	Simple Mail Transfer Protocol – Internet Mail Protocol Standard (RFC 2822 ff)
TS	Transmission Set (EDIFACT) – a file containing a sequence of Interchanges
UNB	EDIFACT Interchange Header
UNZ	EDIFACT Interchange Trailer
VPN	Virtual Private Network
WebDAV	HTTP Extensions for Distributed Authoring (RFC 2518) – increases the command set of http (copy, move)
<i>WebConfig</i>	<i>BusinessMail X.400</i> management configuration platform, with which a customer can configure his environment and his partnerships (aka. relations)
Web Service	A collection of open protocols and standards used for exchanging data between applications or systems. It does so over HTTP using technologies such as XML, SOAP, WSDL and UDDI.
X.435	See PEDI

## 0.4 New in V2.6

- Additional information in chapter 3 regarding new SFTP server and for the SFTP access when using WebDrive Next Generation in chapter 3.3.2
- Additional information for the configuration of AS2 message submission in chapter 6.2.

# 1 Introduction to MessageGate File Interface

## 1.1 With MessageGate File Interface you can:

- Send messages to other *MailBox X.400* users, to partners on other X.400 systems and to Internet mail users. MessageGate supports both the 1984 and 1988/92 X.400 Standards.
- Send messages to Fax recipients.
- Check the status of your messages and import the results into your application.
- Create a Receipt Report for a message that you have received in your application.
- Use standard mechanism of the TCP/IP protocol suite (SFTP, HTTPS with Web-DAV extension or Web Service) to upload messages/data to the MailBox service or to download received messages/data.

## 1.2 X.400 – an International Standard

X.400 is the name of an international standard (ITU, ISO) for the exchange of electronic messages. It specifies the requirements and recommendations of programs used for electronic mail. It includes specifications how a message is to be addressed, which characters or data types are allowed and how communication is to take place.

The X.400 standard defines Delivery Reports/ Notifications and Receipt Reports/ Notifications, to track the status of a submitted message.

Around the world, there are many telecommunication networks, which provide services enabling messages to be exchanged in accordance with the X.400 standard. One of X.400 greatest advantages are that it enables the exchange of messages with users working with different types of computers using secure networks.

## 1.3 The motives in implementing MessageGate

*BusinessMail X.400* has over several years provided a file interface called Batch User Agent (BUA) for host server communication. With the BUA, a customer specific directory is provided on the MailBox service server environment where a customer can use active FTP to upload/download data and command/result files.

Using the BUA, a customer defines in his command file the message data and the recipients to which it should be sent and whether message data should be fetched from his message store (mailbox polling). The user data of the downloaded messages is stored in his customer specific subdirectory and the header information is provided in a result file. The customer can download his user data and the result file via FTP and process these in his application. This interface has stood the test of time, but the syntax used with this interface is not trivial, so the development of customer applications can be expensive.

During the development of an AS2 Gateway for *BusinessMail X.400*, a new interface was designed for customers who used this new communication access method but still needed to communicate with their existing X.400 partners. The new file interface, now called MessageGate File Interface (in the document often referred to as MessageGate), is available to customers that use HTTPS (with WebDAV extension or Web Service) or SFTP to upload and download messages and data. The new MessageGate File Interface has the following advantages:

- SMTP/MIME compatible syntax
- Messages are delivered directly to the customer's specific directory so the polling of the customer's Message Store, as is done with the BUA, is not necessary

There are a lot of tools and libraries available in the public domain supporting the SMTP/MIME syntax. Hence, the cost and complexity of developing applications that use the new MessageGate interface to send and fetch X.400 messages should be a lot less in comparison to the existing BUA interface. MessageGate conforms to RFC 2822 and RFC 1521 and all other MIME relevant RFC.

## 2 Interface Description

### 2.1 Overview

This Chapter gives a short description of all MessageGate File Interface functions. More details of each of the MessageGate functions can be found in subsequent Chapters in this manual. Examples of messages and reports and a list of all error codes can be found in the Appendix C: Examples for Messages and Reports.

A working directory is configured on the *BusinessMail X.400* application servers for each MessageGate user. A MessageGate user must decide whether to use SFTP or HTTPS (with WebDAV extension or the Web Service) to upload/download messages. The filename extension defines how the data will be processed (".IN" → defines data that is to be sent, ".OUT" → defines data that has been delivered by MessageGate).

As the MessageGate process cannot verify in all cases when an uploaded file has been transferred completely using SFTP or HTTPS with WebDAV extensions, a file that is being uploaded should initially use the file extension ".TMP". When the file upload is finished, the file extension should then be renamed by the sending application to "\*.IN". The MessageGate process uses a similar mechanism when delivering messages to the customer specific directory, whereby these temporary files are kept invisible to the MessageGate user.

Please note that the term Order-ID used in the following sections is not to be confused with any EDI terminology, but just implies a processing related reference number.

#### Submit a message

Step 1: Client → Upload the file "M\_Order-id.tmp" using HTTPS/WebDAV or SFTP to MessageGate directory

Step 2: Client → Rename the file "M\_Order-id.tmp" using HTTPS/WebDAV or SFTP to "M\_Order-id.IN". When using HTTPS/Web Services the step 1 is not necessary and the transmission of a file with the extension ".in" is possible because the Web Service will make sure that only a completely uploaded file will be processed.

Step 3: MessageGate processes the file "M\_Order-id.IN", submits an X.400 message and deletes this file

#### Fetch a delivered message

Step 1: MessageGate fetches a message out of the X.400 MTA queue and stores this message in file "M\_Order-id.OUT" in the MessageGate user's directory. When the Closed User Group feature is enabled, the User-ID of the sender of the message will be added to file name.

Step 2: Client → Download the file "M\_Order-id.OUT" (or M\_Order-id\_User-id.OUT in case of enabled CUG) using HTTPS (WebDAV or Web Service) or SFTP and deletes this file in the MessageGate user's directory.

MessageGate will process all files starting with "M\_" as messages. The Order-ID should not be longer than 26 characters. Only integer numbers, the letters A-Z or a-z and a few special characters ("\_", "-", "\$") are valid for the field Order-ID. **Please refer to the information regarding file version numbers in the next Chapter in this manual.**

MessageGate processes a message file that includes header information (sender, recipient, subject, Message-ID ...) and content (one or multiple attachments and S/MIME) in SMTP/MIME (Version 1.0) format. MessageGate will not accept messages having no content. You may add a list of up to 50 recipients (tested) for each message. If the content of this message is signed and/or encrypted, then there is a restriction that the message can only have a single recipient.

**Example of a message that illustrates the syntax of a delivered (e.g., to a MessageGate user) message that has multiple attachments**

**Name: M\_5K00AG0HBDM0F2F8.OUT**

To: "G=MG1;S=MGATE;CN=MG1 MGATE;O=TESTAG;P=MGATE;A=VIAT;C=DE" <49603@viat.de>

From: "G=ipm;S=tester;O=testag;A=viat;C=de" <49637@viat.de>

Message-ID: 614 07/11/13

X-MPDUID: 8B0663A011DCEC4417009682

Date: 13 Nov 2010 13:10:22 +0100

Subject: Test with 3 Body parts

Disposition-Notification-To: "G=ipm;S=tester;O=testag;A=viat;C=de"

MIME-Version: 1.0

Content-Type: multipart/mixed; boundary="MG=\_CA610D0211DC91E900007CAD=\_MG"

--MG=\_CA610D0211DC91E900007CAD=\_MG

Content-Type: text/plain

Content-Transfer-Encoding: 8bit

Test äöüÄÖÜß

--MG=\_CA610D0211DC91E900007CAD=\_MG

Content-Type: application/octet-stream

Content-Disposition: attachment; filename="4d654d1d.zip"

Content-Transfer-Encoding: binary

PK tYr2ÄQa6+ < 4d654d1d.0µ"ls£: ...÷Tñ"ë¥Ói3xJU/\$!DØb2xgd °1Ø

.

• 4d654d1d.0PK 8

--MG=\_CA610D0211DC91E900007CAD=\_MG

Content-Type: application/octet-stream

Content-Transfer-Encoding: binary

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<!-- saved from url=(0062)https://securep7.viat-test.de/~0000049603/result/fetch\_all.RES -->

<HTML><HEAD>

<META http-equiv=Content-Type content="text/html; charset=windows-1252">

<META content="MSHTML 5.50.4930.1700" name=GENERATOR></HEAD>

<BODY><XMP>LOGIN::

.

</XMP></BODY></HTML>

--MG=\_CA610D0211DC91E900007CAD=\_MG—

### Rules when sending messages

Mandatory message elements are: "To:" and Content-Type/Encoding. All other message elements are optional. If the Message-ID is not defined MessageGate will use the Order ID instead to map this to the X.400 Message-ID.

The addresses used for "To:", "Cc:", "Bcc:" and "From:" (Mixture of small and capital letters possible) have an Alias part, where the X.400 address (elements separated by semicolon ";") might be defined, and a RFC2822 address part (<x@viat.de> or <User-id@viat.de>):

#### Possible address forms:

1. "c=de;a=viat;P=prmd;o=org;s=Surname;g=Given name" <x@viat.de>

External address without User-ID

2. "" <73237@viat.de>

Local or external address with User-ID

3. "c=de;a=viat;o=org;s=Surname;g=Given name" <71111@viat.de>

Address, MessageGate will use when delivering a message and the User-ID of the sender is available

When sending a message using the third address form for "To:", "Cc:" or "Bcc:" MessageGate will use the X.400 address elements and will not verify the defined User-ID.

If the central EDI function is activated for a MessageGate user, it is also possible to upload one or several EDIFACT interchanges while using a Transmission Set file to send this data to partners or to download a file that includes such an EDIFACT interchange without any other header information.

**Examples of Transmission Set file**

**Name:** T\_5K00AG0HBDM0F2F8.OUT

UNA:+.? '

UNB+UNOA:2+MGATE1:65+TESTER:65+020508:1413+0709210008'

UNH+EVA0000001+INVOIC:D:95A:UN:ETEIB++0'

.

.

UNT+37+EVA0000001'

UNZ+1+0709210008'

It is no longer necessary to enable the central EDI function (→ only specific EDI/EDIFACT sender will be able to deliver messages) to get a Closed User Group functionality. The CUG might be enabled for any kind of message /message content (only configured MessageGate Partner will be able to deliver messages to directory).

To minimize the number of required elements in the header of the SMTP/MIME messages the mapping rules to create an X.400 message (message type, mapping of MIME body parts into X.400 body parts, request for report etc.) can be defined in so called Host based profiles (user profile, partner profile). It is not necessary to set up a host profile for each partner. MessageGate will use the parameters defined in the user profile if it does not find a partner profile. Defining a partner profile is only necessary if the required communication parameters differ from the default parameters (for example different message type or different report type).

**Parameters in user profile or partner profile**

- Mapping request for report in SMTP header (Disposition-Notification-To:) in X.400 request for Non-Delivery Notification (NDN), Delivery Notification (DN) or Receipt Notification (RN) when sending messages (Default is DN)
- Mapping of Receipt Notification into "Disposition-Notification-To:" for delivered messages (Default is to map)
- Expiry time of X.400 message in minutes (Default is 1440 → 24 hours)
- Type of X.400 message (Default is IPM88) while sending the X.400 message
- Mapping of MIME body into X.400 body part (Default is Variable → Mapping into correspondent body part) while sending (similar rules apply for the opposite direction) the X.400 message (similar rules apply for the opposite direction)
- Deliver binary MIME body in content encoding "BASE64" or "Binary" (Default is Binary)
- Deliver EDIFACT Documents in Transmission Set file or in SMTP syntax file (only available when central EDI Function is activated und only available in user profile → Default is Transmission Set file)

MessageGate will not deliver X.400 reports into the user directory. Information about the status of submitted messages may be requested via status reports. A status report can be requested manually via the interface or delivered automatically (parameter setting in User profile).

**Example for request of status report**

**Name:** S\_040308001.IN

Format: History

Direction: both

**Parameters valid for request of status report (logical “and” notation)**

Disposition: All, Modified (Default is All, all entries, not only those that have changed since the last request)  
 Direction: Sent, Received, Both (Default is Sent)  
 Format: History, CSV-C, CSV-S, Actual (Actual -> Status of messages and not all status changes)  
 Message-ID: Message number or only a part of it  
 Order-ID: Order number or part of it  
 Since: dd-mmm-yyyy hh:mm:ss, since date: day, month, year, hours, minutes, seconds

MessageGate will store the requested status report in a structured file. Hence, it is possible to import the information about the submitted and received messages and the corresponding reports into the customer's application. When sending a message to more than one recipient the status report will contain an entry for each recipient. In this case the Message-ID and Order ID will only be unambiguous in conjunction with a specific recipient address. When receiving messages there is still one entry displaying the originator's address. The complete recipient list is available in the RFC2822 header of the message file.

**Example of a status report entry with parameter Format set to “Actual” (S\_Order-ID.OUT)**

Status Report for UserID 49603; generated 13-NOV-2010 14:56:23  
 Filters: Disposition=All, Direction=Both, Format=Actual, Since=13-Nov-2010

From: "G=ipm;S=tester;O=testag;A=viaT;C=de" 49637@viaT.de  
 Order-ID: 5K00AG0HBDM0F2F8  
 Message-ID: 614 10/11/13  
 MTS-ID: CA610D0211DC91E900007CAD  
 Status: Read  
 Date: 13-Nov-2010 14:01:18 +0100

**Example of a status report entry with parameter Format set to “History” (S\_Order-ID.OUT)**

Status Report for UserID 49603; generated 13-NOV-2010 14:56:22  
 Filters: Disposition=All, Direction=Both, Format=History

To: "" 49637@viaT.de  
 Order-ID: Test\_3\_Body011  
 Message-ID: 260002 12/11/10 MGATE Test  
 MTS-ID: 098FC66111DC91F80000A6BD  
 Sent: 13-Nov-2010 14:52:21 +0100  
 Delivered: 13-Nov-2010 14:52:27 +0100  
 Read: 13-Nov-2010 14:54:00 +0100

Status report entries, where the parameter Format is set to “Actual” will always have a length of 6 lines. If the parameter Format is set to “History”, the numbers of lines is variable (5-7) depending on the message status.

As an alternative to the readable reports output format a report with the format “CSV-S” (CSV format separated by semicolon) or “CSV-C” (CSV format separated by colon), that is optimized for automatic processing, can be selected.

In Chapter 2.9 Communication and Trading Relation Profile, detailed information is available of how to configure the automatic delivery of status reports to the file interface.

Based on a bilateral agreement with your X.400 partner and if the partner has requested this in his message a receipt notification for a delivered message can be generated:

**Example of sending a Receipt Notification**

**Name: R\_5K00AG0HBDM0F2F8.IN**

Possible Values are:

Processed → Receipt Report/Notification (RN)  
 Failed → Non-Receipt Report/Notification (NRN)

The MessageGate process will delete all the processed files (\*.IN) in the user's directory. A "Purger" process will delete all other files (delivered messages, status reports or any other kind of files uploaded to this directory) which are stored in the user's directory, after a predefined lifetime. The *BusinessMail X.400* support team will configure this "lifetime" for each customer individually. It is our recommendation that the customer's application proactively deletes all downloaded files to reduce the number of files stored in the user's directory.

A "Purger" process also exists to process the database relation (Trace\_Tab) where all messages/transactions status information is stored. The *BusinessMail X.400* support team will configure the "lifetime" of these entries (Default is 240 hours) for each customer individually.

The "MessageGate" process will handle two classes of processing errors. It will check the syntax and the completeness of message files in the user's directory and will stop processing the content if there is an error. The message file will remain in the user's directory and the file name will be extended by an error code (see also Appendix B). MessageGate will also add an entry in the database so that the problem is reflected in the status report. Similarly, when the "MessageGate" process experiences an error whilst sending a message via the X.400 MTA, it will add this error code to the respective entry in the database/status report.

It is also possible to subscribe to a MessageGate account that has reduced functionality. In such a case, the user can only receive messages. The transmission of messages or receipt notifications is not possible, nor the manual request of status reports. The automatic delivery of status reports, configured via *WebConfig*, is still possible.

Using the central EDI function in combination with the reduced MessageGate mailbox functionality described above, it would be feasible to use a Web browser to download the data (EDIFACT documents) and feed this into an application.



## 2.2 MessageGate User's directory

The MessageGate process does not require any subdirectories in the user's directory. Only the file extension defines whether a message should be sent ("\*.IN") or a message has been delivered ("\*.OUT") to the user's directory.

The MessageGate process delivers all messages to the user's directory immediately after having processed these from the X.400 MTA queue.

The "Poller" process also frequently checks if there are files in the user's directory to process ("\*.IN" → send messages or Transmission Sets, send Status report and send Receipt Notifications). The scheduling of this check is defined on a user group basis. At the present time only one user group has been defined (schedule interval: 1 minute).

When the "Poller" process finds files with the file extension "\*.IN", it tries to process these files immediately. However, the "Poller" process is not always able to verify that a file has been uploaded completely.

**Therefore, a file should be always uploaded via SFTP or HTTPS/WebDAV using the ".TMP" file extension. The file should then only be renamed to "\*.IN" once the upload has completed.**

Please consider that the actual configuration of the OpenVMS Operating System on which Message Gate's application server is running on, will only accept one dot (".") in the filename. The part of the filename following the dot is known as the file extension.

The first character of the filename defines the type of file (message, status report/report request, receipt report request). The next character, an underscore "\_", separates the Order ID, which can have a maximum length of 26 characters.

Only integer numbers, the letters A-Z or a-z and the following special characters are valid for this Order ID:

Hyphen "-"

Underscore "\_"

Dollar sign "\$"

**The Order-ID as well as the Message-ID must be unique otherwise MessageGate cannot update the transaction log data (Trace\_Tab) in the database when receiving reports. Please consider that the OpenVMS Operating System on which Message Gate's application server is running on stores files using version numbers. That means, that the files with the same file name will not be replaced but will be stored with different version numbers. When accessing the directory via HTTPS/WebDAV or Web Service this behavior is not visible. And, the new SFTP server (OpenSSH) will not show this version number. When accessing a file (open, delete etc.), the file with the highest version number will always be used. In the case where multiple versions with the same file name are present, if a file is deleted, then the next oldest file (ie. with the preceding version number) will become visible in the directory.**

If the Closed User Group of MessageGate is enabled, the file name of a delivered message will be extended and an underscore “\_” separates the User-ID of the sender. The length of the User-ID is 6 digits (including trailing zero “0” if necessary).

MessageGate uses the following file naming convention:

- a) Message file with header for sending:

**M\_<Order-ID>.IN**

- b) Transmission Set file (one or several Interchanges) without header for sending:

**T\_<Order-ID>.IN**

- c) Receipt report file for sending:

**R\_<Order-ID>.IN**

- d) Status report file request:

**S\_<Order-ID>.IN**

- e) Delivered message file with header:

**M\_<Order-ID>.OUT (or M\_<Order-ID>\_User-ID.OUT, if CUG is enabled)**

- f) Delivered Transmission Set file (one Interchange) without header:

**T\_<Order-ID>.OUT**

- g) Delivered Status report file:

**S\_<Order-ID>.OUT**

Capital and small letters for the file name and extension can be used, but please note that the filter mechanism (parameter Order-ID) in status requests is case sensitive.

Please refer to later Chapters in this document to find more details regarding the format of these files.

If the MessageGate “Poller” process discovers a problem in message files (Syntax error, file is incomplete etc.) the file will remain in the directory, but the name of the file will be extended with the respective error code.

Example: M\_12345.in\_ERR0005 (the file was not uploaded completely).

## 2.3 The Message File

The message file includes header and content (MIME or S/MIME content). The syntax of the content conforms to RFC1521 and subsequent RFC or to RFC 5751 when using S/MIME. The header of a message file is based on RFC2822 and includes information that MessageGate as well as the recipient requires to handle/process the content. MessageGate uses only those header elements, which are necessary to build the X.400 message. All other header elements will be ignored.

To minimize the number of header elements the required X.400 message parameters are stored in host profiles (user and partnership profiles)

The content part of the message file may contain a single or several document/ body (multipart) secured or not secured using signature and/or encryption. The MIME content type must be defined in the RFC2822 header information. See the examples in Appendix C. MessageGate does not accept messages having no content.

The maximum message size must not exceed 100 MByte. When sending messages up to 50 recipients (tested) can be added, but there must be at least one "To:" recipient in the message. When using S/MIME content only a single recipient is supported. In delivered message, all recipients are present in the RFC2822 header.

### 2.3.1 The Message Header

MessageGate uses the following header elements to create X.400 messages or to deliver X.400 messages to the MessageGate file interface. MessageGate differentiates between mandatory and optional header elements during the sending of messages (\*.IN). MessageGate provides all header elements when delivering message files (\*.OUT).

**From:**

Message originator: optional (for format rules: see next Chapter), if used, the address must be valid!

**To:**

Message recipient: mandatory (for format rules see next Chapter)

At least one "To:" recipient must be added to the RFC2822 header.

**Cc:**

Message copy recipient: optional (for format rules see next Chapter)

**Bcc:**

Message blind copy recipient: optional (for format rules see next Chapter)

**Message-ID:**

Message-ID that is also used for the X.400 Message-ID when sending the X.400 message or mapped from the X.400 Message-ID in delivered messages, optional

Maximum length is 64 characters from the printable string character set. Please ensure that this Message-ID is unique so that the message recipient has no problems when processing the message and the MessageGate can update the transaction log in the database.

If the MessageGate process does not find a Message-ID in the header of the message, it will map the Order ID to the X.400 Message ID.

**Please be aware that the *BusinessMail X.400 Fax Gateway* only supports a Message-ID with a maximum length of 16 characters. When a longer Message-ID is used, the Message-ID visible on the fax document will be truncated.**

### Subject:

The subject element will be mapped when sending an X.400 message and will be mapped from an X.400 message subject field when delivering a message to the file interface, optional.

The maximum length of the subject element is 128 characters (Teletex character set T.61).

**Please be aware that the “MessageGate” process will not convert German characters (ä,ö,ü,Ä,Ö,Ü,ß) in the X.400 message subject field when delivering a message (e.g., against the conventions of RFC2822). In addition to accepting this format for messages destined for transmission, MessageGate will also accept messages with the standard encoding (ISO-8859-1 character set → “=?iso-8859-1?x?...txt...?” where x=Q → quoted-printable or x=B → Base64).**

### Date:

Date of message (Date and Time Specification of RFC2822): optional  
MessageGate will not use this date when sending an X.400 message.

Examples of the format:

2 Nov 2010 09:31:44 +0100

(Format of the send date of the delivered message, English abbreviation of month)

Tue, 2 Nov 2010 09:31:44

### Disposition-Notification-To:

Request a report: optional.

This request will be mapped to an X.400 report request based on the Host profile parameter “When sending X.400 Messages map a requested Notification into”:

0 → Requests Non-Delivery Notification (NDN)

1 → Requests Delivery Notification (DN)

2 → Requests Receipt and Delivery Notification (RN and DN)

For more information, see Chapter 2.9.2 X.400 Reports.

MessageGate will map this RFC2822 header element to a delivered message if a receipt notification was requested by the sender of the X.400 message and if this RN is not suppressed in the profile. The customer’s application may send a receipt notification (see Chapter 2.8

Send Receipt Notification).

In a delivered message MessageGate the X.400 address of the originator within the quotation marks will be the value of the element "Disposition-Notification-To:". When sending a message, it is not necessary to add a value within the quotation marks because MessageGate will ignore this information.

Examples:

Disposition-Notification-To: "" (when sending messages)

Disposition-Notification-To: "c=de, a=viat; s=tester; O=testag" (delivered message)

### **X-MPDUID:**

Message number allocated by the X.400 MTA when a message has been submitted (only used in delivered messages), maximum 32 characters (the MPDU ID of a submitted message will be displayed in the status report parameter MTS-ID)

### **MIME-Version:**

Optional: Default: 1.0 (only this version will be supported).

### **Content-Type:**

Type of MIME data (RFC 2045 "Multipurpose Internet Mail Extensions" and subsequent RFC or RFC 5751 "S/MIME V3.2 Standard"), **mandatory**.

This parameter defines the content type of a message. This must be present after the header elements and only in combination with the elements "Content-Transfer-Encoding:" and if necessary "Content-Disposition:". The actual content follows a blank line.

MessageGate supports the following MIME Content Types:

Single Text body:

text/plain; charset=ISO-8859-1

Single Binary body:

application/octet-stream

Multipart:

multipart/mixed; boundary="====\_NextPart\_Nr."

Signed content:

multipart/signed; protocol="application/pkcs7-signature"; micalg=shaxxx;  
boundary="====\_NextPart\_Nr."

(supported values for xxx are 1, 256, 384, 512)

Encrypted (and optional signed) content:

application/pkcs7-mime; smime-type=enveloped-data; name="smime.p7m"  
(supported Cipher Suites 3DES, AES128, AES192, AES256)

For more details see Chapter 2.3.3 Message Content und 2.3.4 S/MIME secured content.

### **Content-Transfer-Encoding:**

Encoding of MIME content. **mandatory**

7bit

8bit

quoted-printable  
base64  
binary

### Content-Disposition:

File name of MIME content, optional.

## 2.3.2 Address format

This Chapter describes the address format of "To:", "Cc:", "Bcc:" and "From:" header elements. The address has two parts, an Alias and a RFC2822 address part.

The Alias part is used to define the X.400 address. The RFC2822 part is divided into domain "@viat.de" and a surname that may have the value "x" or "User-ID" (*BusinessMail X.400* internal identifier) of a partner.

Possible address forms are:

- a) Addressing partner using X.400 address in Alias:

"X.400 address elements, separated by semicolon" <x@viat.de>

This address form should be used if there is no partner profile and no User-ID entry configured for this partner or if a transfer to a Delivery Unit (e.g., FAX Gateway) is required. When sending an X.400 message, the MessageGate process will use the parameters configured in the user profile to determine the syntax of the message. When delivering messages MessageGate will use this address if it does not find a User-ID entry in the database for the originator of the message.

Example: "c=de;a=viat;o=testteam;s=tester;g=first" <x@viat.de>

This address form may also be used to send messages to partners that do not have direct X.400 access but reachable via *BusinessMail X.400* gateways. A good example would be a message that is to be sent via the Fax gateway. In this case, certain parameters that are necessary to define the Fax message must be written into DDA address elements (number, page length ...). The value of these elements can be changed for every Fax transaction/ destination.

"DDA:Service=Fax;DDA:Format=A4;X121=06212946911;A=viaT;C=DE" <x@viat.de>

**When using the character “;” within an address field it must be duplicated otherwise MessageGate will interpret it as a separator between address elements.**

- b) Addressing the partner using User-ID in the RFC2822 address (Alias is empty)

"" <User-ID@viat.de>

This address form should be used if the trading partner is a Customer of *BusinessMail X.400* MailBox service or if a User-ID is already configured for an external partner (MessageGate or EDI partnership → see also Chapter 2.5 *Transmission Set File Format*).

Example: "" <69365@viat.de>

- c) Addressing Partner using X.400 address in Alias and User-ID in the RFC2822 address

"X.400 address elements, separated by semicolon" <User-ID@viat.de>

This address form is used by the MessageGate process when delivering messages whose originator User-ID is in the database:

Example: "c=de;a=viat;o=testteam;s=tester;g=first" <99999@viat.de>

**If you use this address form when sending a message, MessageGate will only use the X.400 address in the alias and will not verify the User-ID.**

The maximum length of these addresses is 256 characters (Alias + RFC2822 address part). If this is insufficient to enter a unique X.400 address, we recommend that you configure a partner profile to address this partner only using the User-ID.

The following X.400 address elements (capital and small letters may be used for the element type and for the value, but the value is not case sensitive) are valid for Alias:

C=xx;	Country code (2 characters Printable String, e.g., de)
A=xxxxx;	Name of ADMD (16 characters Printable String, e.g., viaT)
P=xxxxx;	Name of PRMD (16 characters Printable String, e.g., MGI)
O=xxxxx;	Organization (64 characters Printable or Teletex String, e.g., Telekom)
OU1=xxxx;	Organization unit 1 (32 characters Printable or Teletex String)
OU2=xxxx;	Organization unit 2 (32 characters Printable or Teletex String)
OU3=xxxx;	Organization unit 3 (32 characters Printable or Teletex String)
OU4=xxxx;	Organization unit 4 (32 characters Printable or Teletex String)
DDA:xxx=xxxx;	Domain Defined Attributes (Type, 8 characters = Value, 128 characters, both Printable and Teletex String, e.g., service=fax)
S=xxxxx;	Surname (40 characters Printable or Teletex String)
G=xxxxx;	Given name (16 characters Printable or Teletex String)
CN=xxxxx;	Common name (64 characters Printable or Teletex String)
N-ID=xxxxx;	Box Identifier (Unique Agent ID, 32 characters Numerical)
X121=xxxxx;	Network Identifier (15 characters Numerical)
T-ID=xxxx;	Terminal Identifier (24 characters Printable String)
I=xx;	Initials (5 characters Printable String)
Q=xxx;	Generation (qualifier) (3 characters Printable String)

Some of these elements will be not used when sending X.400 messages depending on the parameter "X.400 Content-Type" in the host profile. For example, the Common name will not be used for "To:", "Cc:", "Bcc:" and "From:" if the value of the parameter "X.400 Content-Type" is "IPM84".

*See also information about X.400 addresses in Chapter 2.9 Communication and Trading Relation Profile and in Appendix A X.400 Address elements.*

### 2.3.3 Message Content

The message content must be defined in the header of a message in the element “Content-Type:”. A message can be sent with one body part, or several (multipart) body parts/documents secured (signed and/or encrypted) or unsecured.

In the X.400 message, the MessageGate process will send a S/MIME secured content (signed and/or encrypted) within a single FTAM body part where the signed content will be named “*smime.p7s*” and will have the OID “id-signedData” {1 2 840 113549 1 7 2}. If the content is encrypted or encrypted and signed it will be named “*smime.p7m*” and will have the OID “id-signedAndEnvelopedData” {1 2 840 113549 1 7 4} (see also Chapter 2.3.4 S/MIME secured content).

When sending (ie. upload to file interface) a message with only one unsecured binary body the value of the element “Content-Transfer-Encoding:” must be set to either *Base64* (7bit encoding) or *Binary* (8bit encoding). The MessageGate process will decode the Base64 encoded data and will send the message with 8bit encoded data.

A parameter in the host profile defines the Content-Transfer-Encoding the MessageGate process determines to use Base64 or Binary when storing a delivered message into the user’s directory.

When sending several documents in one message the value of the first element “Content-Type” must be “multipart/mixed” with a definition of a boundary character string. This boundary character string then separates the subsequent body parts with the “Content-Type” definition for the individual body part.

When mapping the content of a message into an X.400 message the MessageGate process will use two host profile parameters “X.400 Content-Type” and “Bodypart” to create this X.400 message.

The host profile parameter “X.400 Content-Type” will determine the format of the X.400 message. The default for this parameter is “IPM88” and should be not changed within the user profile. Only in exceptional circumstances (e.g., problems when using the Common name in the address) should the value “IPM84” be used in the partnership profiles.

The following rules will apply in mapping the content of an uploaded message into the corresponding X.400 body parts when the value of the communication profile parameter “Bodypart” is set to “variable”.

#### Text body:

Content-Type: text/plain; charset=ISO-8859-1

- Content-Transfer-Encoding:
  - 7bit
  - 8bit
  - quoted-printable
- Content-Disposition:
  - Attachment; Filename = < file name >

will be mapped to

- General Text body part, ISO-Latin-1 (ISO-8859-1) character set, or IA5 IRV Repertoire if a message is sent to an external mail service that only supports the 1984 X.400 Standard. When downgrading to the 1984 Standard the characters above 0x'7F' will be not converted.



- A file name defined in the “Content-Disposition” element will force the MessageGate process to create a BP15 FTBP (FTAM body part) and provide this file information to the recipient. If the partner does not support FTBP, do not add a file name to the text content or create a relation for this partner using Bodypart Mapping “IA5-text” or “ISO-Latin-1”.

### Binary body:

Content-Type: application/octet-stream

- Content-Transfer-Encoding:
  - base64
  - binary

will be mapped to

- Body part 14

application/octet-stream

- Content-Transfer-Encoding:
  - base64
  - binary
- Content-Disposition:
  - attachment; filename=<file name>

will be mapped to

- Body part 15 (FTAM body part, FTBP), if value in profile is IPM88,
- Body part 14 (Bilaterally defined body part), if value in profile is IPM84, the file name will be not mapped.

### Multipart body:

multipart/mixed; boundary="---=\_NextPart\_xxx..."

The individual body parts must be defined using the rules described above.

**Change the value in the parameter “Bodypart” in the partner profile if your partner can only process the same type of X.400 body part and your application cannot adapt to this. See more details about this parameter in Chapter 2.9 Communication and Trading Relation Profile.**

See *Appendix C: Examples for Messages and Reports* to find several examples of messages with different content types.

The MessageGate process will use the following rules when delivering messages to the file interface:

### Single Text body:

- General Text body part or IA5Text
- ISO-Latin-1 character set or IA5 Repertoire

will be mapped to

- Content-Type:
  - text/plain
- Content-Transfer-Encoding:
  - 8bit

**Single Binary body:**

- Bodypart 14 (BP14)

will be mapped to

- Content-Type:
  - application/octet-stream
- Content-Transfer-Encoding:
  - binary/base64 (depending on the parameter value in the host profile)

- Body part 15 (FTAM body part, FTBP)

will be mapped to

- Content-Type:
  - application/octet-stream
- Content-Transfer-Encoding:
  - binary/base64 (depending on the parameter value in the host profile)
- Content-Disposition:
  - attachment; filename=<file name>

**Multibodypart:**

will be mapped to

multipart/mixed; boundary="---=\_NextPart\_xxx...."

The individual body parts must be defined using the rules described above.

Please take note of the following potential issues when implementing applications:

- Older X.400 E-Mail clients may not support BP15/FTBP and will attach the body part/document file information in a separate IA5 text body (so called CDIF information) before attaching the data in a binary body part (BP14). MessageGate will use this file information when delivering messages to the file interface and will directly add the file name in the Content Disposition element preceding this binary body. Therefore, the number of body parts mapped to the delivered message will decrease.
- The X.400 standard only defines a small number of body part types whereas in the MIME standard many applications have defined their own specific content types. MessageGate attempts to analyze the start of the body part based on preconfigured pattern recognition to set a correspondent MIME content type. At present, only the patterns for EDIFACT and PDF are used in this so-called *DOCMAGIC* function.

### 2.3.4 S/MIME secured content

When sending a message with a signed and/or encrypted content to an X.400 recipient it is necessary to use a S/MIME syntax within the message file. When receiving such a message the MessageGate process will map the SMIME content unchanged (the only exception is the parameter based converting of encrypted content → Content-Transfer-Encoding Binary to Base64 and vice versa) into the message file.

In the X.400 message the S/MIME content will be transferred within a single BP15/FTAM body part to be compatible to older X.400 clients and to have the option to downgrade it to BP14 without changing the S/MIME content. Before using S/MIME content please verify that your partner is able to process this type of message.

Starting with V5.2 of FileWork and UA-FI the handling of such a S/MIME content is integrated and hence can provide the user data in the unsecured format for further processing. When using other or older X.400/P7 clients appropriate external tools, for example OpenSSL, have to be used to extract the user data out of the S/MIME content.

When sending X.400 messages to an external partner in a domain connected via a RFC2822 compliant MTA (see also Chapter 7 SMTP MTA and MessageGate) you should use an appropriate Content-Transfer-Encoding within the S/MIME content (for text documents use quoted printable and for binary Base64). Neither the File Interface nor the RFC2822 compliant MTA will be able to change this encoding without invalidating the signature. The modification of the encrypted content by the intervening processes is not possible due to the absence of the private key.

The following rules describe the details of sending and receiving secured messages.

1. Rule to send signed content (one or several documents with signature including certificate of signer in a separate MIME body, supported value for micalg=SHA1, SHA256, SHA384, SHA512):

```
Content-Type: multipart/signed; protocol="application/pkcs7-signature"; micalg=shaxxx; boundary=---...
```

```
Content-Transfer-Encoding: binary >>> (only for signed MIME Content where the encoding "Binary" will be used, otherwise no entry)
```

```
<Space>
```

```
This is a S/MIME signed message
```

```
<Space>
```

```
boundary=---...
```

```
<MIME Content>
```

```
Boundary=---...
```

```
Content-Type: application/pkcs7-signature; name="smime.p7s"
```

```
Content-Transfer-Encoding: binary/Base64 (use binary or Base64)
```

```
Content-Disposition: attachment; filename="smime.p7s"
```

```
Boundary=---...--
```

>>> will be mapped to

```
Body Part 15 FTBP (FTAM Body Part)
```

```
Name= smime.p7s
```

OID = "id-signedData" {1 2 840 113549 1 7 2}

>>> The whole S/MIME content will be mapped unchanged (including the Content-Transfer-Encoding of the signature).

2. Rule to send encrypted content (one or several documents, supported Cipher Suites 3DES, AES128, AES192, AES256):

Content-Type: application/pkcs7-mime; smime-type=enveloped-data;  
name="smime.p7m"

Content-Transfer-Encoding: binary/Base64... >>> (use binary or Base64)

Content-Disposition: attachment; filename="smime.p7m"

<Space>

<Encrypted MIME Content>

>>> will be mapped to:

Body Part 15 FTBP (FTAM Body Part)

Name= smime.p7m

OID = "id-signedAndEnvelopedData" {1 2 840 113549 1 7 4}>>> Content-Transfer-Encoding Base64 will be converted into Binary

3. Rule to send signed and encrypted content (one or several documents with signature including certificate of signer in a separate MIME body, supported value for micalg=SHA1, SHA256, SHA384, SHA512):

Content-Type: multipart/signed; protocol="application/pkcs7-signature";  
micalg=shaxxx; boundary=---....

Content-Transfer-Encoding: binary >>> (only for signed MIME Content where the encoding "Binary" will be used, otherwise no entry)

<Space>

This is a S/MIME signed message

<Space>

boundary=---...

<MIME Content>

Boundary=---...

Content-Type: application/pkcs7-signature; name="smime.p7s"

Content-Transfer-Encoding: binary/Base64 >>> (use binary or Base64)

Content-Disposition: attachment; filename="smime.p7s"

Boundary=---...--

>>> will be wrapped into

application/pkcs7-mime; smime-type=enveloped-data; name="smime.p7m"

Content-Transfer-Encoding: binary/Base64 >>> (use binary or Base64, supported Cipher Suites 3DES, AES128, AES192, AES256)

Content-Disposition: attachment; filename="smime.p7m"

<Space>

<Encrypted signed MIME Content>

>>> and mapped to:

Body Part 15 FTBP (FTAM Body Part)

Name= smime.p7m

OID = "id-signedAndEnvelopedData" {1 2 840 113549 1 7 4}

>>> Content-Transfer-Encoding Base64 will be converted into Binary

#### 4. Rule when receiving messages with signed content:

Body Part 15 FTBP (FTAM Body Part)

Name= smime.p7s

OID = "id-signedData" {1 2 840 113549 1 7 2}

>>> will be mapped to:

Content-Type: multipart/signed; protocol="application/pkcs7-signature";  
 mime-calg=sha1; boundary= .... >>> (with signature including certificate of signer in separate MIME body)

Content-Transfer-Encoding: binary >>> (only for signed MIME Content where the encoding „Binary“ will be used, otherwise no entry)

<Space>

This is a S/MIME signed message

<Space>

boundary=---...

<MIME Content>

Boundary=---...

Content-Type: application/pkcs7-signature; name="smime.p7s"

Content-Transfer-Encoding: binary/Base64 >>> (use binary or Base64)

Content-Disposition: attachment; filename="smime.p7s"

Boundary=---...--

#### 5. Rule when receiving messages with encrypted content:

Body Part 15 FTBP (FTAM Body Part)

Name= smime.p7m

OID = "id-envelopedData" {1 2 840 113549 1 7 3}

>>> or

Body Part 15 FTBP (FTAM Body Part)

Name= smime.p7m

OID = "id-signedAndEnvelopedData" {1 2 840 113549 1 7 4}

>>> will be mapped to:

Content-Type: application/pkcs7-mime; smime-type=enveloped-data;  
 name="smime.p7m"

Content-Transfer-Encoding: binary/Base64 >>> (binary or Base64 depends on parameter in partner profile)

Content-Disposition: attachment; filename="smime.p7m"

<Space>

<Encrypted MIME content>

#### 6. Rule when receiving messages with signed and encrypted content:

Body Part 15 FTBP (FTAM Body Part)

Name= smime.p7m

OID = "id-signedAndEnvelopedData" {1 2 840 113549 1 7 4}

>>> will be mapped to:

Content-Type: application/pkcs7-mime; smime-type=enveloped-data;  
name="smime.p7m"

Content-Transfer-Encoding: binary/Base64 >>> (binary or Base64 depends on parameter in partner profile)

Content-Disposition: attachment; filename="smime.p7m"

<Space>

<Encrypted signed MIME content>

>>> with wrapped signed content:

Content-Type: multipart/signed; protocol="application/pkcs7-signature"; micalg=shaxxx; boundary=---.... >>> (supported value for micalg SHA1, SHA256, SHA384, SHA512)

Content-Transfer-Encoding: binary >>> (only for signed MIME Content where the encoding „Binary” will be used, otherwise no entry)

<Space>

This is a S/MIME signed message

<Space>

boundary=---...

<MIME Content>

Boundary=---...

Content-Type: application/pkcs7-signature; name=smime.p7s

Content-Transfer-Encoding: binary/Base64 >>>( use binary or Base64)

Content-Disposition: attachment; filename="smime.p7s"

Boundary=---...--

## 2.4 The Forwarding of a message is not supported

The MessageGate File Interface does not support the forwarding of a delivered message. If a partner forwards a message to a MessageGate File Interface user, only the user data (payload body parts) will be provided in the message file. There will be no information about the forwarded message itself.

## 2.5 Transmission Set File Format

### 2.5.1 The central EDI functionality

When using the central *BusinessMail X.400* EDI function (must be ordered separately) it is not necessary to use the RFC2822 syntax to address your partner. You can simply upload one or several EDIFACT interchanges in one Transmission Set file via the file interface and the central EDI function will process this file.

MessageGate will use the information stored in the EDI partnership entries of the database to add the X.400 address of the partner. Now, the central EDI function only supports the sending or the receipt of one EDIFACT interchange per X.400 message in either a text body or a binary body part (BP14). Hence the central EDI function now will deliver only Transmission Set files that include one EDIFACT interchange to the user's directory. This restriction might be removed in one of the next versions of MessageGate, so any prospective applications should be designed to be able to handle received Transmission Set files that include more than one EDIFACT interchange.

**When the central EDI function is activated, all received messages (sent by partners), which include more than one EDIFACT interchange or one EDIFACT interchange and other documents, will be rejected. And a Non-Delivery Notification ("Invalid arguments" → Reason:1, Diagnostisic:11) will be sent to the partner's X.400 address. When checking the status of delivered messages the entries for such rejected messages will be also found.**

The Transmission Set files do not include a RFC2822 Header, so when sending an X.400 message, the UNB Interchange reference number will be used for the subject and the Message ID of this message:

Message ID: <INTERCHANGE CONTROL REFERENCE> (max. 14 char.)

Subject: EDIFACT <INTERCHANGE CONTROL REFERENCE> (max. 22 char.)

The EDIFACT interchange included in the Transmission Set file will be attached to an X.400 message based on the MessageGate parameter "Bodypart" configured in the host-based EDI trading relation profile. The default is text body (ISO-Latin-1). If the trading partner requests the document as a binary body part, the parameter must be change to "Bilateral Bodypart 14". The Content Type "EDI" should be used only if your partner does not support another message type. In this case, the same restriction of one EDIFACT interchange per received X.400 message applies; otherwise, the PEDI (X.435) message will be refused.

When using the central EDI function a MessageGate user must have at least one own EDI account/Identifier. This EDI account consists of an EDI ID/ Reference Number (e.g., ILN or GLN) and an optional EDI Qualifier (e.g., "14" for commerce or "65" for X.400). MessageGate will either check or use this Identifier in the UNB EDIFACT header:

- the EDI sender when sending an X.400 message
- the EDI receiver when delivering an X.400 message

By default, the central EDI function will deliver an EDIFACT interchange to a MessageGate user if the EDI Identifier in the UNB matches the EDI Identifier configured in the EDI account. In addition, it is possible to setup a Closed User Group (CUG) for each individual EDI Identifier/account. If this CUG is activated, only an EDIFACT document sent by a preconfigured EDI Trading partner will be delivered to the file interface.

An EDI Trading Relation entry must be configured for each partner that a MessageGate user wants to address via the central EDI function (e.g., send messages to). This entry includes the partner's EDI Identifier and his X.400 address, as well as a test flag entry. The test flag in the UNB header of an interchange can be used by a partner as routing information if he wants to differentiate between his live and his test environment (address) using the same EDI Identifier. If there is a single trading relation with enabled test flag parameter, the central EDI function will only send documents with test flag in UNB header to this EDI partner (see also the EDI Process User Guide available for download on the Service Website of *BusinessMail X.400* <https://www.service-viat.de> in section Patches / Downloads & Handbücher). It is also possible to assign an EDI Trading Relation to one or several of a MessageGate user's EDI accounts. The parameters in an EDI Trading Relation will define how to create the X.400 message.

Dependent on the settings of the host profile parameter "Map requested Report into" the MessageGate process will use the following notification settings in an X.400 message:

0 → Requests Non-Delivery Notification (NDN)

1 → Requests Delivery Notification (DN)

2 → Requests Receipt Notification and Delivery Notification (RN and DN)

For more information, please also see Chapter 2.9.2 X.400 Reports.

Please be aware that when using the central EDI Function the MessageGate process will not check the content of an EDI interchange, but it will check the syntax of the UNA, UNB and UNZ segments. If a syntax error is identified (e.g., no space in UNA or reference number in UNB and UNZ is different), the document will not be processed. In such a case MessageGate will send an NDN (Non-Delivery Notification) with Diagnostic Code 0x0B hex (11 decimal, Invalid Arguments) while processing this message. If MessageGate identifies a syntax error, but is not able to identify the EDIFACT content, it will deliver it as a message to the file interface. For uploaded interchanges, if MessageGate identifies a syntax error, it will set the status of this transaction to error and discard the message. If the X.400 originator of a message uses a wrong ILN or if there is no EDI Trading Relation (own EDI Identifier configured with Closed User Group) MessageGate will send an NDN (Non-Delivery Notification) with Diagnostic Code 0x11 hex (17 decimals, No bilateral Agreement).

To gather information about messages that have been rejected, a status report can be submitted via the file interface. Use the value "Both" or "Received" of the element "Direction:" to list entries of received (and rejected) messages. For more information, please also see Chapter 2.7 The Status Report.



## 2.5.2 The Content of Transmission Set Files

It is possible to upload two or more EDIFACT interchanges in one Transmission Set file. The interchanges can be separated by adding an empty line with <CR/LF> but this is not necessary.

The number of interchanges in one Transmission Set file is theoretically unlimited, but it is recommended not to exceed 100 interchanges. The size of a Transmission Set file may not exceed 100 MB.

MessageGate now delivers a Transmission Set file to the MessageGate file interface that only includes one EDIFACT interchange. A future version of MessageGate may support two and more EDIFACT interchanges per message. Anyone developing an application should consider this.

A parameter in the host profile defines whether MessageGate delivers an EDIFACT interchange in the form of a Transmission Set file or in the form of a message with MIME syntax to the file interface.

For more information, please also see Chapter 2.9 Communication and Trading Relation Profile.

## 2.6 Closed User Group (CUG)

Independent of central EDI funktion it is possible to enable a Closed User Group for any kind of partner, so that only configured MessageGate partner will be able to deliver any kind of messages/documents to the directory. As described in the last chapter a non configured sender will receive a NDN including Diagnostic Code 0x"11" (17 decimal, no bilateral agreement). The configuration of the CUG even allows the limitation of the message content to a single body part (any kind, text only, binary only) if your application only accepts certain content (e.g., in case of UTF-8 encoded contents a binary body part should be mandatory or in conjunction with Web Service profile v3). The limitation to allow only one body part is also possible without configuring a CUG. Please get in touch with BusinessMail X.400 administration to clarify if the limited delivery should be enabled for your MessageGate account and which kind of limitation you need.

## 2.7 The Status Report

### 2.7.1 General

MessageGate does not deliver X.400 reports to the file interface (only messages/data). However, it is possible to request a status report that provides information about the status of all messages processed by MessageGate (sent and delivered). The information for each transaction is stored in a database relation (Trace\_Tab) for a predefined period. The Default Purge Time is 240 hours (e.g., an entry will be deleted if it is older than 240 hours), but this value can be changed individually for each account. When uploading a status report request the MessageGate user receives information about the status of his messages in a structured file, which can be imported into an application or a tracking tool.

When sending a message to several recipients an entry will be made in the status report for each recipient. In such a case, the Message-ID and Order ID are only unambiguous in combination with the recipient address while processing the entry.

**Be aware that the logical sequence of these entries may be different to the sequence of recipients in the RFC2822 header.**

There will only be one entry for a delivered message, also if the originator sent the message to several recipients. Check the RFC2822 header of the delivered message file to get the recipient list.

The address displayed in the status report is identical to the address in the RFC2822 header of the message and the type of address will be unchanged. The type of address is displayed always in the same manner (first character: capital letter, subsequent characters: small letter) → “To:”, “Cc:” and “Bcc:”.

### 2.7.2 Request a Status Report

MessageGate provides status reports on request (S\_Order-ID.IN) or automatically if configured in the host profile (configuration via *WebConfig*, see Chapter 2.9.8 Configure automatically generated Status Report). *WebConfig* allows you to view status reports or to download those reports in structured files (CSV file), for more details please see Chapter 2.9.7 Request Status Report in WebConfig.

**Please note that after a status report request has been placed in the directory an exclusion period (default is 5 minutes) for subsequent status report requests is applied. This can be configured individually for each MessageGate account by the MessageGate administration. All other status report requests placed in the directory during this exclusion period will be renamed using the error code 9999 and will be not processed.**

To reduce the amount of information (number of entries) included in status reports MessageGate provides different selection criteria. One can select only one entry (using Order ID or Message-ID) and request to receive the whole message history (readable or CSV format) instead of only the present message status. MessageGate offers the following selection criteria/parameter in a status report request:

- whether the status of an entry has changed since the last status report or not
- whether a message was sent or delivered
- only defined Message-ID or part of it
- only defined Order ID or part of it
- date/time when a message was sent or delivered

An additional parameter defines whether to indicate the actual message status or the whole message status history.

Parameter: Value	Explanation:
Disposition: All	Select all messages, independent of whether the status has changed or not ( <b>Default</b> ).
Disposition: Modified	Select all messages where the status has changed since the last status report (e.g., received DN or RN).
Direction: Sent	Only select sent messages ( <b>Default</b> ).
Direction: Received	Only select received messages.
Direction: Both	Select sent and received messages.
Format: Actual	Show only the present status of messages and the timestamp of the last status change ( <b>Default</b> ).
Format: History	Show all changes of status and their respective timestamp (Sent/ received message, Received DN/NDN, sent/ received RN/NRN).
Format: CSV-S	Show all changes of status and their respective timestamp (Sent/ received message, Received DN/NDN, Sent/ received RN/NRN) using CSV semicolon separated format (equivalent History).
Format: CSV-C	Show all changes of status and their respective timestamp (Sent/ received message, Received DN/NDN, Sent/ received RN/NRN) using CSV colon separated format (equivalent History).
Message-ID: xxx*	Only show those messages where Message-ID starts with a predefined string. Always use "*" for a wildcard or partial wildcard search.
Order-ID: xxx*	Only show those messages where Order ID starts with predefined string. Always use "*" for a wildcard or partial wildcard search. The Order-ID is case sensitive.
Since: dd-mmm-yyyy hh:mm:ss	Select only those entries stored in the Trace_Tab after the defined date/time. When only a date is defined, all entries from this date (starting at 00:00) will be selected.

There are default values for the first two criteria/parameters and for the format that MessageGate will use if there is no definition in a 'status report request'. The number of returned entries can be limited further using additional criteria in the 'status report request' (logical "AND"- combination).

If MessageGate uses the three default values, all sent message entries stored in the database (Trace\_Tab, not deleted by the "Purger" process) will be written to the status report file, independent of whether the status has changed since the last status report request or not. Each of the entries will only display the status of the last message transaction, but not its history.

### 2.7.3 Status Report Syntax (readable structure)

If MessageGate has processed a status report request (S\_Order-ID.IN) it will select those entries based on the selection criteria and create a status report file. For this report file the same filename as the status report request file is used, but with the “.OUT” file extension.

**Please ensure that your application only uses a unique Order-ID in the name of the status report request file. The operating system on which the Message-Gate application is hosted on will not replace the existing status report request files. Instead, it will store the new file using a higher version number. WebDAV or Web Service clients will not exhibit these version numbers. These version numbers may cause problems in your application.**

The content of the status report file consists of header information and message entries. The first line in this header information shows the User-ID of the Message-Gate User and the timestamp of this report. The second line displays the filter criteria used to create this report.

Separated by an empty line (CR/LF) the message entries follow this header information.

Depending on the selected criterion “Format” a message entry may consist of 6 (Format:actual) or 5-7 lines (Format:history). The information regarding sender/recipient, Order-ID, Message-ID, and MTS/MTA ID are available in both formats. Only the status information differs between both formats.

Syntax of the first line of the header information

**Status Report for UserID xxxxx; generated dd-mmm-yyyy hh:mm:ss**

The UserID value field has a minimum length of 4 digits, but values of up to 6 digits are also possible.

The timestamp element “dd” defines the day of month (no leading zeros in case of one-digit values), “mmm” the name of month (English abbreviation), “yyyy” the year, “hh” the hours (24 hours schema, with a leading zero in case of one-digit values), “mm” the minutes (with a leading zero in case of one-digit values) and “ss” the seconds (with a leading zero in case of one-digit values). The timestamp is always in MET or MEST, (see also the syntax of the message entry).

**Example:**

Status Report for UserID 4911; generated 13-Sep-2010 13:43:32

Status Report for UserID 23423; generated 7-Sep-2010 07:12:01

Syntax of the second line of the header information

**Filters: Disposition=x, Direction=x, Format=x, Message-ID=x, OrderID=x, Since=dd-mmm-yyyy**

There are no spaces between the filter criteria and the corresponding value. The individual filter parameter are separated by the “,” (comma) and “ ” (space) character. MessageGate always adds the “Disposition=”, “Direction=” and “Format=” criteria fields. The other criteria fields are only displayed if they have been defined in the ‘status report request’.

The possible values of the selection criteria fields:

- “Disposition=” [“All” | “Modified”]
- “Direction=” [“Sent” | “Received” | “Both”]
- “Format=” [“Actual” | “History”]

- “Message-ID=” the selection criteria value used, or the selection criteria will be not displayed.
- “Order-ID=” the selection criteria value used, or the selection criteria will be not displayed
- “Since=” date in the form (dd-mmm-yyyy hh:mm:ss).used or the selection criteria will be not displayed. Please consider the “lifetime” of entries in the database (so called ‘Purge Time’ used in the Trace\_Tab).

**Example:**

Filters: Disposition=All, Direction=Both, Format=Actual, Since=1-Jan-2009

- ➔ All entries for sent/received messages since 01.01.2009 00:00:00 and which are still stored in the database are selected independent of whether the status has changed will be displayed with their actual status.

Filters: Disposition= Modified, Direction=Received, Format=History

- ➔ All entries for received messages where the status has changed since the last report will be displayed with detailed status information.

Filters: Disposition= Modified, Direction=Sent, Format=History, Order-ID=EDI\*

- ➔ All entries of sent messages where the status has changed since the last report and where Order ID starts with “EDI” will be displayed with detailed status information.

**Syntax of sent message entries with “Format” = “Actual”**

Each line consists of a field type followed by a semicolon “:”, a space “\_” and the field value that may also include spaces. The type of recipient address can be “To:”, “Cc:” or “Bcc:”.

<b>To:</b>	Recipient address (when using the central EDI function the address is always of the form "" <user-id@viat.de>)
<b>Cc:</b>	Carbon copy recipient address
<b>Bcc:</b>	Blind carbon copy recipient address
<b>Order-ID:</b>	Order number also defined in the file name (maximum 26 characters)
<b>Message-ID:</b>	Message number defined in the SMTP Message or when using central EDI function the Interchange Control Reference ID of the EDIFACT interchange (maximum 64 characters)
<b>MTS-ID:</b>	The MTS identification used by the MTA when the message was sent (maximum 32 characters)
<b>Status:</b>	The actual message status (for possible values see below)
<b>Date:</b>	Timestamp of last status change in the format dd-mmm-yyyy hh:mm:ss +xxxx, where “dd” is day of month (no leading zero in case of one-digit value), “mmm” is the month name (English abbreviation), “yyyy” is the year, “hh” is the hour (24 hours, with leading zero in case of one-digit value), “mm” is the minute (with leading zero in case of one-digit value) and “ss” is seconds (with leading zero in case of one-digit value). The value +xxxx shows the time difference to UTC/GMZ, so +0100 for MET or +0200 for MEST (daylight saving time).

Syntax of received message entries with "Format" = "Actual"

Each line consists of a field type followed by semicolon ":", a space and the value that may also include spaces.

<b>From:</b>	Originator address
<b>Order-ID:</b>	Order number assigned by MessageGate (maximum 26 characters)
<b>Message-ID:</b>	Message identification assigned by the message originator (maximum 64 characters)
<b>MTS-ID:</b>	Message identification assigned by originating MTA (maximum 32 characters). This will be mapped to the element X-MPDUID in the SMTP header.
<b>Status:</b>	The actual message status (for possible values see below)
<b>Date:</b>	Timestamp of last status change in the format dd-mmm-yyyy hh:mm:ss +xxxx, where "dd" is day of month (no leading zero in case of one-digit value), "mmm" is the month name (English abbreviation), "yyyy" is the year, "hh" is the hour (24 hours, with leading zero in case of one-digit value), "mm" is the minute (with leading zero in case of one-digit value) and "ss" is seconds (with leading zero in case of one-digit value). The value +xxxx shows the time difference to UTC/GMZ, so +0100 for MET or +0200 for MEST (daylight saving time).

Syntax of sent message entries with "Format" = "History"

Each line consists of a field type followed by colon ":", a space " " and the field value that may also include spaces. Please be aware that the number of status entries is variable and that the entries are interdependent (e.g., for a "READ" status it is necessary that there are entries for a "Sent" and "Delivered" status). A negative Report/ value will terminate the transaction so that there can be no more status entries for this message. The type of recipient address can be "To:", "Cc:" or "Bcc:".

**To:** Recipient address (when using the central EDI function the address is always of the form "" <user-id@viat.de>)

<b>Cc:</b>	Carbon copy recipient address
<b>Bcc:</b>	Blind carbon copy recipient address
<b>Order-ID:</b>	Order number also defined in the file name (maximum 26 characters)
<b>Message-ID:</b>	Message number defined in SMTP Message or when using the central EDI function the Interchange Control Reference ID of the EDIFACT interchange (maximum 64 characters)
<b>MTS-ID:</b>	The MTS identification used by the MTA when the message was sent (maximum 32 characters).
<b>Sent:</b>	Time stamp of sent message Format: dd-mmm-yyyy hh:mm:ss +xxxx, where "dd" is day of month (no leading zero in case of one-digit value), "mmm" is the month name (English abbreviation), "yyyy" is the year, "hh" is the hour (24 hours, with leading zero in case of one-digit value), "mm" is the minute (with leading zero in case of one-digit value) and "ss" is seconds (with leading zero in case of one-digit value). The value +xxxx

shows the time difference to UTC/GMZ, so +0100 for MET or +0200 for MEST (daylight saving time).

**Or**

**Error:** When MessageGate fails to process a message file or fails to send a message/document error information will be created and will be displayed with a respective timestamp in this entry

Format:

Error: dd-mmm-yyyy hh:mm:ss (Reason: nnnnnnnn, Diagnostic: n)

**Delivered:** Timestamp of when the message was delivered into recipient's mailbox (Format: see Sent:)

**Or**

**Failed:** Message was not delivered and the MTA created a Non Delivery Notification (NDN) that includes information about the problem

Format:

Failed: dd-mmm-yyyy hh:mm:ss (Reason: n, Diagnostic: n)

**Read:** Time stamp when recipient has processed/ read message (Format see Sent:)

**Or**

**Denied:** Message was deleted or discarded by recipient

Format:

Denied: dd-mmm-yyyy hh:mm:ss (Reason: n, Diagnostic: n)

#### Syntax of received message entry with "Format" = "History"

Each line consists of a field type followed by a colon ":", a space " " and the field value that may also include spaces. Please be aware that the number of status entries is variable and that the entries are interdependent (e.g., for "READ" it is necessary that there is an entry for a "Received" status). Negative Reports/ values will terminate the transaction so that there will be no more status entries for this message.

**From:** Originator Address

**Order-ID:** Order-ID set by MessageGate (maximum 26 characters)

**Message-ID:** Message-ID the originator sets in the message (maximum 64 characters)

**MTS-ID:** The MTS identification defined by the sending MTA (maximum 32 characters). The element X-MPDUID in the SMTP header will show same value.

**Received:** Time stamp Received message

Format:

dd-mmm-yyyy hh:mm:ss +xxxx, where "dd" is day of month (no leading zero in case of one-digit value), "mmm" is the month name (English abbreviation), "yyyy" is the year, "hh" is the hour (24 hours, with leading zero in case of one-digit value), "mm" is the minute (with leading zero in case of one-digit value) and "ss" is seconds (with leading zero in case of one-digit value). The value +xxxx shows the time difference to UTC/GMZ, so +0100 for MET or +0200 for MEST (daylight saving time).

**Or****Failed:** Message was not delivered

Format:

Failed: dd-mmm-yyyy hh:mm:ss (Reason: n, Diagnostic: n)

**Read:** Time stamp for Sent Receipt Notification

Format:

Read: dd-mmm-yyyy hh:mm:ss +xxxx, where “dd” is day of month (no leading zero in case of one-digit value), “mmm” is the month name (English abbreviation), “yyyy” is the year, “hh” is the hour (24 hours, with leading zero in case of one-digit value), “mm” is the minute (with leading zero in case of one-digit value) and “ss” is seconds (with leading zero in case of one-digit value). The value +xxxx shows the time difference to UTC/GMZ, so +0100 for MET or +0200 for MEST (daylight saving time).

**Or****Denied:** Information for sender that message was discarded

Format:

Denied: dd-mmm-yyyy hh:mm:ss (Reason: n, Diagnostic: n)

Status of sent messages:

Sent

Error: (Reason: nnnnnnnn, Diagnostic: n)

Delivered

Failed: (Reason: n, Diagnostic: n)

Read

Denied: (Reason: n, Diagnostic: n)

Status of received messages:

Received

Failed: (Reason: n, Diagnostic: n)

Read

Denied: (Reason: n, Diagnostic: n)

See also *Appendix C: Examples for Messages and Reports* for more examples.



## 2.7.4 Syntax of Status reports (CSV structure)

If MessageGate has processed a Status report request (S\_Order-ID.IN), it will select those entries defined in the selection criteria and create a report file. It will use the same file name but with the ".OUT" extension. Please be aware that the CSV format is optimized for automatic processing and so it is difficult to read.

The content of the status report file consists of header information and message entries. The header information in the first line contains the User-ID of the MessageGate User and a timestamp for this report. In the second line, the filter criteria used for this report are displayed.

Unlike the report format described in the last Chapter the header of the CSV format contains additional information. Separated by an empty line (CR/LF) there is another line containing the field identifier, the field name.

Field name:	Explanation:
From	In the received message the originator address (Alias/X.400 address + RFC2822 address, maximum 256 characters). Quotation marks will be used for the alias and also within the CSV structure for string declaration, so this field will start with three quotation marks, two quotation marks will follow the alias address and at the end of the field, another quotation mark will define the end of this string. (""G=test;S=tester1;O=testag;A=viaT; C=de""<95344@viaT.de>").
To	The recipient address used in the submitted message (Alias/X.400 address + RFC2822 address, maximum 256 characters). Format see "From:". Address type see Rcpt Type.
Order-ID	Order identifier defined in the file name (maximum 26 characters) within quotation marks.
Message-ID	Message identifier or reference number of EDIFACT interchanges in Transmission Set (maximum 64 characters) within quotation marks.
MTS-ID	Identifier (=MPDU ID, maximum 32 characters) generated by the MTA when sending a message within quotation marks.
Received	Timestamp (UTC/GMT) of receiving message (format yyyy/mm/dd hh:mm:ss used for File Interface, dd.mm.yyyy hh:mm within WebConfig) without quotation marks or "Failed" if message could not be delivered (Error code see Reason und Diagnostic).

Sent	Timestamp (UTC/GMT) of sent message (format yyyy/mm/dd hh:mm:ss used for File Interface, dd.mm.yyyy hh:mm within WebConfig) without quotation marks or “Error” if message could not be sent (Error code see Reason und Diagnostic).
Delivered	Timestamp (UTC/GMT) of delivery time of a sent message (format yyyy/mm/dd hh:mm:ss used for File Interface, dd.mm.yyyy hh:mm within WebConfig) without quotation marks or “Failed” if message could not be delivered (Error code see Reason und Diagnostic).
Read	Timestamp (UTC/GMT) when a sent message was processed (format yyyy/mm/dd hh:mm:ss used for File Interface, dd.mm.yyyy hh:mm within WebConfig) without quotation marks or “Denied” if message had been discarded (Error code see Reason und Diagnostic).
Reason	Reason code for a failed action (for details see Appendix B) without quotation marks.
Diagnostic	Diagnostic code for a failed action (for details see Appendix B) without quotation marks.
Errordate	Timestamp (UTC/GMT) for a failed action (format yyyy/mm/dd hh:mm:ss used for File Interface, dd.mm.yyyy hh:mm within WebConfig) without quotation marks.
Rcpt Type	Recipient address type: To, Cc or Bcc, when using central EDI function the address type is EDI (equal To).

## 2.8 Send Receipt Notification

It is possible to send a positive or negative Receipt Notification (RN) for each delivered message if the originator has requested this in his message. MessageGate will display this request in the header element "Disposition-Notification-To: <X.400 Originator Address>" of delivered SMTP message.

It is also possible to configure in the host profile that a requested RN will not be mapped into an SMTP message so that the structure of the header does not change.

To send a Receipt Notification, create a file where file name includes the Order-ID of delivered message.

The format of file name is: „R\_<Order-ID of delivered message>.IN".

MessageGate will use the following values defined in this file to create the X.400 reports:

Processed	→	Receipt Report/Notification (RN)
Failed	→	Non-Receipt Report/Notification (NRN) with Reason Code "0" (Discarded)

Other values in the file will cause MessageGate to refuse this request and to add an error code to the file name.

When using MessageGate with enabled Closed User Group, you must be aware, that the originator's User-ID added to the filename of delivered message file is not part of the Order-ID. So do not add it to the name of receipt notification's „in" file.

**Please be aware that the sending of Receipt-Notifications is a chargeable item within the BusinessMail X.400 and other X.400 services!**

## 2.9 Communication and Trading Relation Profile

### 2.9.1 General

One must configure a communication profile for each MessageGate file interface user. This profile defines the rules of how to map the content of a SMTP message into an X.400 message and how to build the header of the X.400 message.

With an existing communication profile, a MessageGate user can send to and receive messages from X.400 partners and is also able to use the *BusinessMail X.400* Gateway services (Fax-Gateway and the Internet/SMTP-Gateway). For these cases, MessageGate will use the definitions stored in the communication profile to create X.400 messages.

The configuration of separate Trading Relation Profiles is only necessary if the trading partner requires a special message type/content or if non-default Notifications is required for this Trading Relation. The *BusinessMail X.400* central Administration will configure a Trading Relation Profile on request, but this service is chargeable. Hence, we recommend the use of *WebConfig*, a *BusinessMail X.400* Web-based application that allows, amongst other services, the administration of Trading Relations.

All standard Web browsers (enable Java-script to facilitate the use of all the offered functions) are supported for the access to the *WebConfig* application. To use *WebConfig* (URL <https://webconfig.viat.de/webconfig>) one must first download a Client certificate (check information provided on the *BusinessMail X.400* Service URL: <https://www.service-viat.de>) and import this certificate into the selected Web browser before trying to access the *WebConfig* application. At this stage, the use of cookies must be enabled. When accessing the *WebConfig* application the *BusinessMail X.400* Proxy Server will request this certificate for each connection.

The *WebConfig* application itself will prompt for Username and Password to authenticate the *WebConfig* user. This Login data is configured when your MessageGate account is provisioned. If you do not receive this information, please contact the *BusinessMail X.400* Helpdesk. After the login into your *WebConfig* account, you may download a personalized Client certificate and modify the properties of your *WebConfig* account so that this Client certificate is used to authorize the access to your *WebConfig* account. Use the Service URL: <https://www.service-viat.de> to find the certificate of the WebConfig CA (Certificate Authority) which signed this personalized Client certificate and import it also into the certificate store of your browser.

The main *WebConfig* menu contains the following menu items dependent on the subscribed features:

- User administration to change the *WebConfig* password, verify the change log, download data (certificates, parameter files, configuration files) and modify the properties of the *WebConfig* Account (time zone and separator for CSV files, Web browser language selection and cookie lifetime).
- Administration of certificate used for WebDAV or Web Service access
- Administration of MessageGate Properties, Trading Relations and Status Reports

- Administration of EDI Trading Relations (own EDI Identifier, Partner's EDI Identifier), if central EDI functionality is activated
- Administration of SMTP/Internet Trading Relations (SMTP Filter)
- Access to extended service information

The following screenshot shows the menu items of a MessageGate user where the central EDI function is activated.

...T...Systems· Business flexibility



BusinessMail X.400 :: WebConfig mgate (49603)

**MessageGate Relation :: Properties** ⓘ

**Properties** ⓘ

When sending X.400 Messages map a requested Notification into Delivery Notification (DN)

Receipt Notification requested in X.400 messages should be   
☐ ignored   
☒ send, if client had sent notification

Message Expiration 1440 Minutes

X.400 Content Type IPM84  
IPM88

Bodypart IA5 Text  
Bilateral (Bodypart 14)  
ISO Latin 1  
Depends on context (variable)

Encode binary data binary  
base64

Format of Output ⓘ SMTP  
TS (TransmissionSet)

Purge time ⓘ 240 Hours

Ok Cancel

In the menu item “MessageGate Relation - Properties” the default communication parameters for all partners will be available and in the “MessageGate Relation - Create, Show/Modify or Delete” menu items the communication parameters of special Relations can be configured. The values configured in menu item “MessageGate Relation - Properties” will be used as the initial values when starting to configure new Relations. *WebConfig* will ask you whether you also want to modify the existing relations or not when changing some of these parameters.

## 2.9.2 X.400 Reports

The X.400 standard offers different report types (Delivery or Receipt Notifications) to verify the status of a message / a transaction. By default, MessageGate will request a Delivery Notification (DN) within an X.400 message when the header element “Disposition-Notification-To” is defined in the SMTP message. However, this can be modified in the communication profile or in a Trading Relation profile:

- Non-Delivery Notification → Send a Report only if the message cannot be delivered to the recipient's mailbox
- Delivery Notification → Send Report if the message was delivered to the recipient's mailbox (also implies Non-Delivery Notification)
- Receipt Notification → Send a Receipt Notification Message if the recipient had processed the message (read/fetched). In this case, a Delivery Notification will also be requested and the receipt of a DN will be displayed in the status report.

If “Disposition-Notification-To” is not defined in the SMTP message an X.400 report will not be requested and the MessageGate user will not receive any information if the delivery of a message has failed.

When sending documents via the central EDI function an X.400 report will always be requested based on the parameters configured in the host profile.

It depends on the acknowledgement mechanisms agreed between trading partners and the transferred data whether it is necessary to request X.400 reports or not. If there is already an acknowledgement mechanism defined at the application layer, it may not be necessary to monitor the transport (X.400) layer and hence to request X.400 reports. However, we recommend monitoring the delivery of a message into the partner's mailbox. Please be aware when requesting a Receipt Notification that the partner must agree to send this type of report because the X.400 standard offers the recipient of a message the choice to send a Receipt Notification or not. The mapping of a Receipt Notification to a sent message is only possible if the “Purger” process has not already deleted the entry of this transaction in the Trace\_Tab.

**Please be aware that the sending of Receipt Notifications is a chargeable item within the context of the *BusinessMail X.400* service as well as with other X.400 service providers.**

Select in the menu item “When sending X.400 Messages map requested Notification into” the report mechanism (Default is Delivery Notification DN) that fits to your processes.

The menu item “Receipt Notifications requested in X.400 Messages should be” defines whether a request for a Receipt Notification within a X.400 message should be mapped into “Disposition-Notification-To” of a delivered SMTP message or not. By default, the request will be mapped, but this setting may cause problems in certain applications, so it is possible to suppress this in the host profile.

### 2.9.3 X.400 Header Information

The menu item “Message Expiration” defines the expiration time of the X.400 message. The default value of the item “Message Expiration” is 1440 minutes (= 24 hours). This means the MTA or set of MTAs will try to deliver a message to a correctly addressed recipient for 24 hours before a Non-Delivery Notification (NDN, if requested!) is created and sent back to the originator of the message.

The item “X.400 Content Type” defines the type (structure) of X.400 message that will be sent to a partner. Possible values are “IPM84” and “IPM88” (Default).

When using “IPM84” MessageGate will create an X.400 message of type P2 (X.400 Standard 1984) and it will support IA5 text body part and the binary body part (Bilaterally defined body part BP14 → Binary data without file information). In this case, it will also not add the “Common name” address element to the originator’s and recipient’s address. Use the value “IPM84” only if your partner has problems processing messages that MessageGate has sent to his mailbox (for example the partner is a customer of a mail system that only supports the 1984 X.400 Standard or has problems processing the X.400 “Common name” address attribute).

When using “IPM88” (default value for this item) MessageGate will create an X.400 message of type P22 (X.400 Standard 1988/92) and will also support a BP15 FTAM body part (FTBP), that includes the binary data and additional file information (e.g., file name). MessageGate will also add the X.400 “Common name” address attribute to the originator’s and to the (local) recipient’s address in the X.400 message.

If you are using the central EDI functionality to send messages, the message type will be defined in the EDI Trading Relation. The central EDI functionality offers in addition the message type PEDI (X.435) to send the data (EDIFACT document).

### 2.9.4 X.400 Body parts

The item “Bodypart” defines the mapping of MIME content into X.400 body parts when sending an X.400 message. It is possible to use the same body part type mapping (values: IA5-Text, Bilateral body part 14, ISO-Latin-1) or to map the MIME Content into equivalent X.400 body part (value: Variable). Variable is the default value for this item and should only be changed in a Trading Relation:

- if your partner always requests the same X.400 body part and your application is not able to provide the equivalent MIME content type in all cases
- if your application will send MIME text content including a file name, that must be sent as an X.400 text body part.

Be aware when sending an X.400 message to different recipients, the MessageGate process will ignore existing Trading Relations and will use the mapping rules configured in the property’s entries instead.

When sending documents via the central EDI function, the mapping of an EDIFACT document to an X.400 body part will be defined in the EDI Trading Relation.

## 2.9.5 Encoding of binary data

This item defines whether MessageGate should use MIME Content-Transfer-Encoding Binary or Base64 when mapping an X.400 binary body part (BP14 or BP15/FTBP) into an equivalent MIME Content Type. All E-mail clients should be able to process Base64 encoded messages. This parameter will also be used for encrypted S/MIME content but ignored for signed S/MIME content to map a valid signature. When sending a message, the MessageGate process will accept both types of Content-Transfer-Encoding.

## 2.9.6 File format (Format of output)

If the central EDI function is activated this item will define whether an EDIFACT document sent by a partner will be delivered in an RFC2822 structured file or in a Transmissions Set file. This menu item is only available in the base host profile and not in the Trading Relation profile.

## 2.9.7 Request Status Report in WebConfig

The menu item “Status Report” enables you to request the status of transactions (sent or received messages) and view it directly in the Web browser or to download it in a CSV file or to open this file directly using an appropriate program (for example Microsoft Excel). You can restrict the number of displayed/stored entries by entering a date, by checking the box “Only failed messages” or using a search string filter when selecting reports for viewing directly via the Web browser. The character used as a separator in the CSV file will be defined in the properties of *WebConfig* Management. The default separator is the “;” semicolon.

The screenshot shows a web browser window with the address bar displaying 'BusinessMail X.400 :: WebConfig' and the user 'mgate (49603)'. The main content area is titled 'MessageGate Relation :: Request Status Report' with an information icon. Below the title, there is a 'Records since' input field with a date-time format '(Format: DD-MMM-YYYY hh:mm:ss)'. A checkbox labeled 'Only failed messages' is present. Two buttons, 'Show Status Report' and 'Download CSV file', are displayed. A 'Filter:' input field with a search icon is also visible. At the bottom, a status message reads: 'Status Report for UserID 49603; generated 18-Feb-2009 09:20:42 Disposition=all, Direction=Both, Format=History'.

## 2.9.8 Configure automatically generated Status Report

You can receive status reports in your MessageGate directory by requesting these via the MessageGate API or the periodic generation of these status reports can be configured via *WebConfig*. This menu item defines the time when a status report should be generated (day of week, start time, end time) and the schedule (minimum is 30 minutes). A file name prefix can be defined for this file and MessageGate will add a timestamp to the file name to create a unique file name.

The content of these reports is defined by the selection criteria used (see also Chapter 2.6 The Status Report).



BusinessMail X.400 :: WebConfig mgatetester (49640)

### MessageGate Relation :: Automatically generated Status Report

☒ Enable automatically generated Status Report

**Properties**

Prefix of file name: Status\_49640

Only failed messages: ☐

Days of the week: ☒ Monday, ☒ Tuesday, ☒ Wednesday, ☒ Thursday, ☒ Friday, ☒ Saturday, ☒ Sunday

Daily start date: 0:00 (MET/MEST, Format: hh:mm)

Daily end date: 24:00 (MET/MEST, Format: hh:mm)

Schedule: 30 Minutes (0=Only one time at daily start date) 30

Disposition: All

Direction: Both

Format: Actual

Ok Cancel

While enabling the option “Only failed messages” you may reduce the number of entries in the report.

## 2.9.9 EDI Relation

If the central EDI functionality has been enabled for the MessageGate account a main menu item “EDI Relation” should be displayed where you can configure your EDI accounts and your EDI Relations. If you want to use the central EDI functionality, you must configure a minimum of one own EDI account (Identifier). If no Closed User Group (CUG) has been enabled for this own EDI account, all your trading partners can send EDIFACT documents (one document/ interchange per message) to this account.

BusinessMail X.400 :: WebConfig mgate (49603)

### EDI Relation :: Create EDI Account

**EDI Account**

EDI ID:

EDI Qualifier:

Closed User Group: ☐ (only configured partner were able to send messages to this EDI Account)

Ok Cancel

If you plan to send EDI documents to your partners, you must configure an EDI Relation for each partner. In this EDI Relation, you must add the X.400 address or a

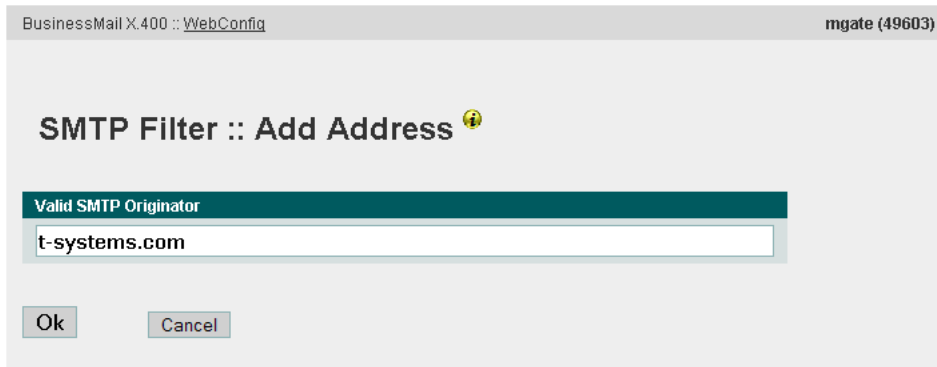
User ID (Mailbox X.400 internal identifier) to a partner's EDIFACT Identifier (EDI ID, for example ILN/GLN and optionally an EDI Qualifier).

For an EDI Relation you may use the optional test flag in the UNB Header of the EDIFACT document to differentiate between your partner's live and test system. If you enable the "test flag" within a partner relation and there is no second relation for production data (without test flag in the UNB header) this data will not be sent to this partner. In this menu section, you can define the format of an X.400 message and the type of body part sent to your partner.

## 2.9.10 SMTP Filter

By default, there are no restrictions about Internet E-Mail users sending mails to your MessageGate account. However, it is possible to setup a filter in the SMTP-Gateway to prevent the delivery of mails sent by Internet E-Mail users or restrict it to configured partner/ domains.

Using the "SMTP Filter - Add Address" menu item complete E-Mail addresses or only parts thereof (for example the domain part) may be configured. If adding partial addresses, no wild card characters should be used (see screenshot).



Please be aware that your configured address filter rules will only be effective if the "SMTP Filter - Set Status" menu item is set to "partly disabled".

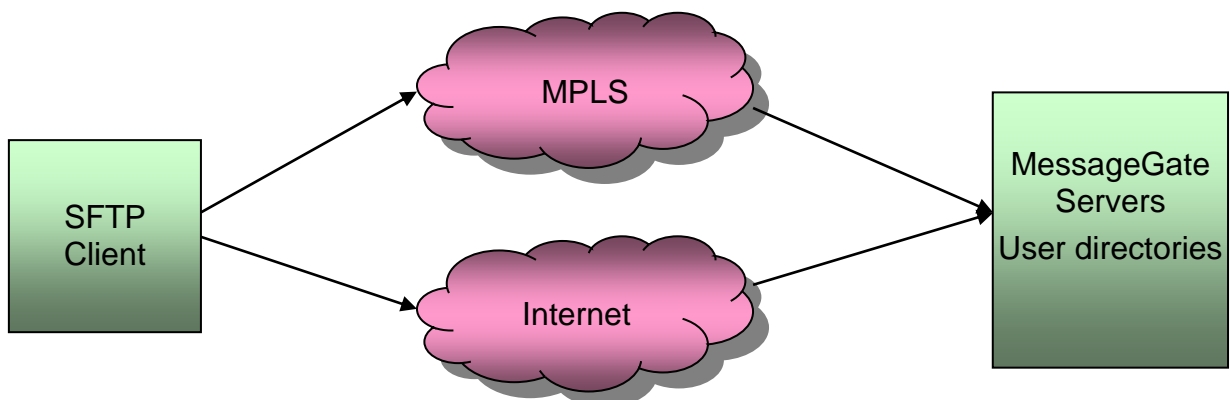
### 2.9.11 Web Service of WebConfig

When using the Web Service of WebConfig you may change the values of the parameters in the properties of the MessageGate account, in the MessageGate or EDI relations and manage your own EDI Identifiers directly within your application. You must use a CSV structure while uploading the necessary data and will receive data in same way. Details regarding this Web Service you will find in the quick-guide available on the Service URL of BusinessMail X.400 (<https://www.service-viat.de> in section Patches, Download und Handbücher).

## 3 Access via SFTP (SSH)

### 3.1 General Information

For network administrators the use of active FTP (together with BUA → old file interface) may cause problems because FTP requires two TCP/IP sessions. As an alternative *BusinessMail X.400* offers the use of SFTP (part of SSH suite using Port 22) for secure data transfer. Each MessageGate user will be configured on the Application server (OpenVMS) and the MessageGate directory will be the login directory. This means that it is possible to directly upload and download data from the Application server after login into the SFTP account.



You can use the MPLS network of Deutsche Telekom or the Internet to access the *BusinessMail X.400* Application servers.

To access your directory via the Internet, use the logical address "sftp.viat.de", for MPLS use the IP address 164.31.4.145.

### 3.2 Features to note

A client using SFTP when accessing *BusinessMail X.400* will authenticate itself via a username and private key against the public key stored on the SFTP host, and in turn, the host will authenticate itself to the client with its own host public key. Different SFTP clients may use different types of public keys (to convert the different types we recommend the use of *puTTYgen*, which is part of *putty* and the *WinSCP* product set). A potential SFTP user must provide his public key (format SSH2 with End of Line character LF, if necessary, modify the line end using an appropriate Tool, e.g., Notepad ++ when using Windows OS) to the *BusinessMail X.400* administration, who will store this key on the respective SFTP server. A user may provide more than one public key. On the SFTP server system each SFTP user has a user specific subdirectory named SSH2 containing his public keys and the file AUTHORIZATION. All the valid keys are defined in this file. We recommend a minimum RSA key length of 3072 bit when using SFTP communication. A SFTP user may copy new public keys directly into the SSH2 subdirectory, but only the *BusinessMail X.400* administration can modify the AUTHORIZATION file. Please ask the *BusinessMail X.400* Helpdesk to initiate this change.

In parallel (using a separate Port definition) to the existing SFTP server a new (OpenSSH) SFTP server will be introduced where the valid keys will be managed in

the `authorized_keys` file located in the same SSH2 directory. This new SFTP server will also support ed25519 signed public keys.

## 3.3 Recommended SFTP Communication modules

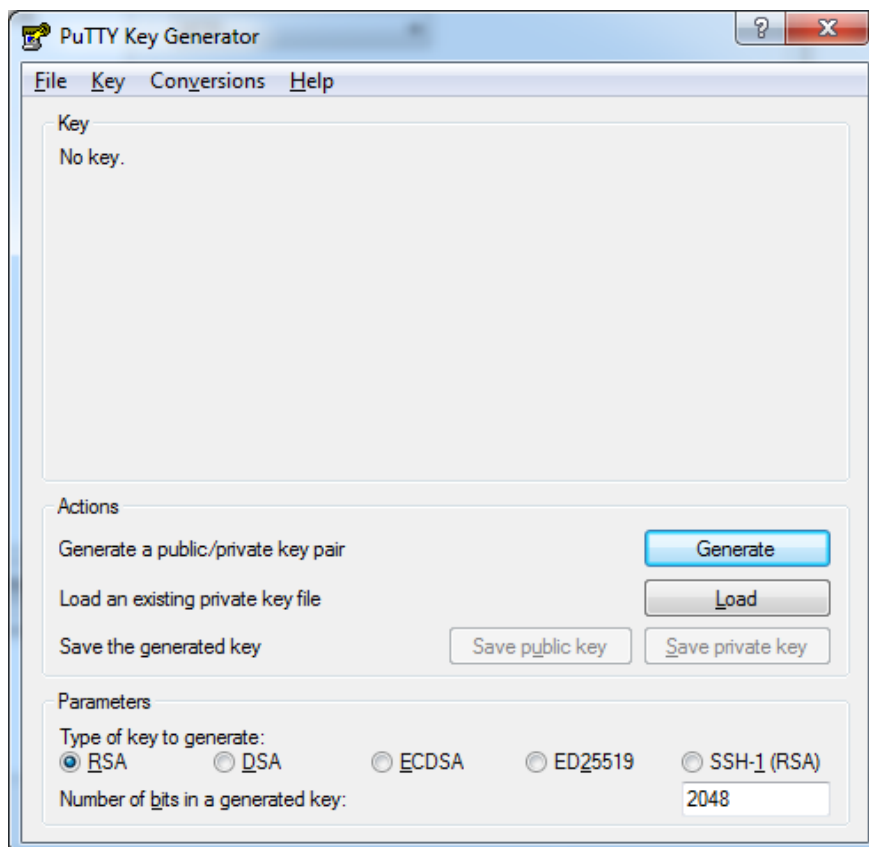
### 3.3.1 Using Microsoft® Windows 32 Bit Operating systems

#### ■ WinSCP

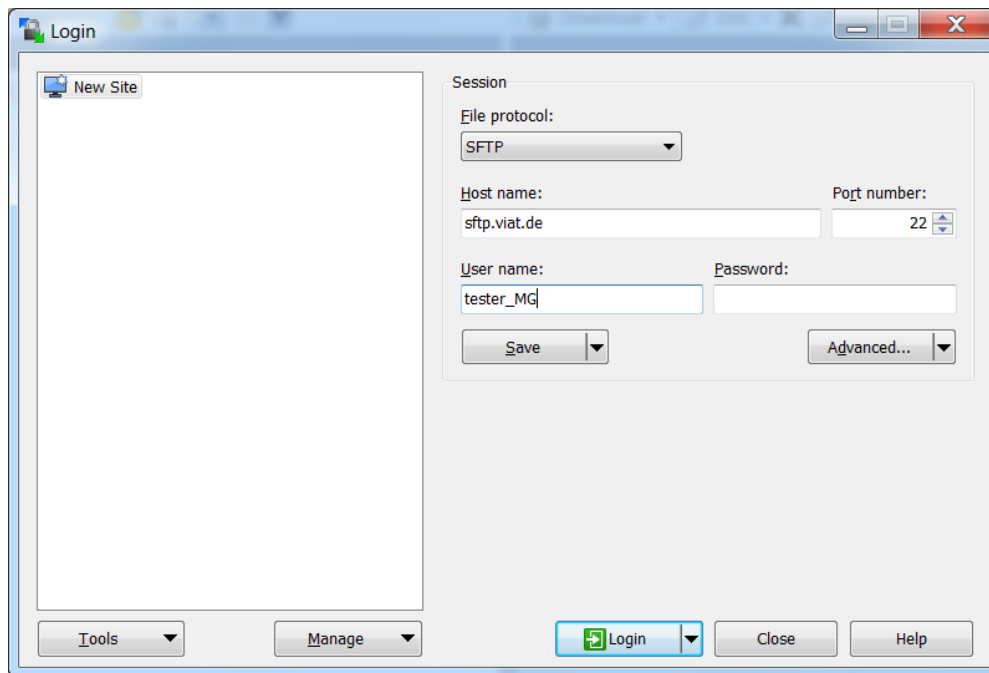
*WinSCP* is an Open-Source client providing SFTP capabilities (and supports https/ WebDAV and FTP) in combination with a Graphical User Interface to transfer to and download from data from an SSH server. *WinSCP* also offers Batch mode and a command line interface running in a DOS window with a command set comparable to those of a standard FTP client to transfer data automatically.

#### Configuration:

Create a key pair while running *WinSCP* using the menu item “Run *puTTYgen*” and send the public key to the *BusinessMail X.400* Administration who will store this key on the SFTP server systems (please use the line ending character LF, if necessary, modify the line end using an appropriate Tool, e.g., Notepad ++ when using Windows OS). Store the private key locally to use with *WinSCP*.



Start *WinSCP*, create a “New Site” and select the protocol “SFTP”. Enter the Host name (SFTP server name - DNS name) or the numerical IP address (provided by the *BusinessMail X.400* Administration) and the provided Username. Select Button “Advanced.../Advanced.../SSH/Authentication” to define the file including the private key as a value of the parameter “Private key file”. Store this configuration when exiting the menu item “Advanced...”.



When this configuration has been completed, one should connect to the SFTP server for the first time. When connecting to the SFTP server *WinSCP* will prompt for the passphrase of the private key, if configured (can be imported via file in batch mode). When establishing the first session the SFTP server will send its public key and you will be prompted to accept it to continue the session.

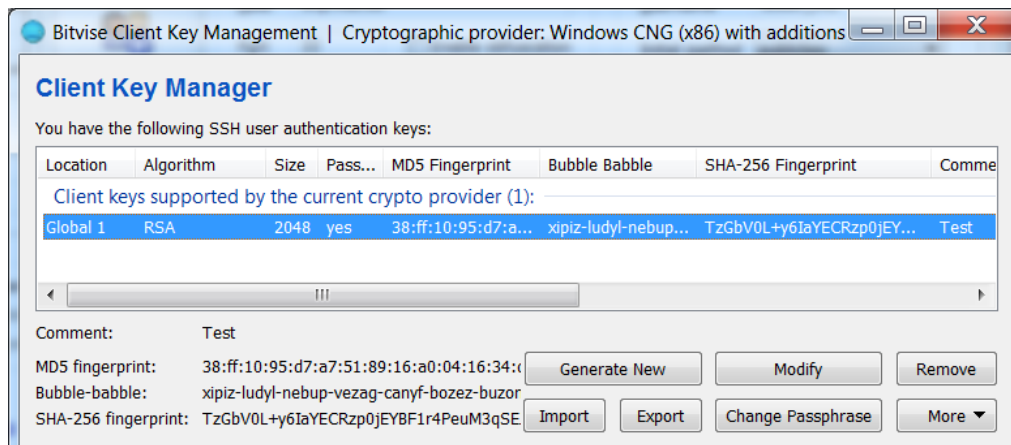
The menu item “Advanced...” allows you to tailor *WinSCP* to your requirements.

#### ▪ Bitwise SSH Client (formerly Tunnelier)

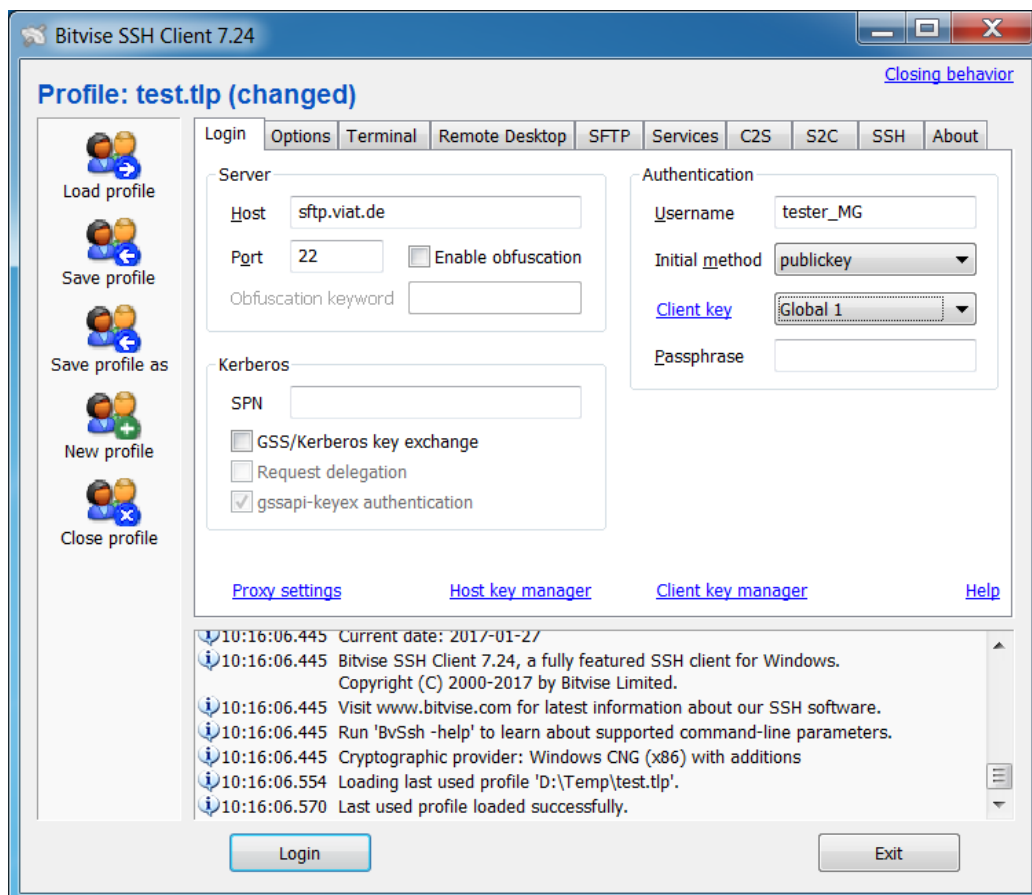
This is a freeware SFTP client also offering a Graphical User Interface, Batch mode and a DOS command line mode.

#### Configuration:

Create a key pair using the program Keypair Management (you can also start the program in the Bitwise GUI) and send the public part of the key to the *BusinessMail X.400* Administration where the public key will be stored on the SFTP server systems (please use the End of Line character LF, if necessary, modify the line end using an appropriate Tool, e.g., Notepad ++ when using Windows OS). Please verify that no space character is used in the value of the comment field when creating the key pair, because the Keypair Manager will export the comment field without quotation marks into the public key file. The SFTP Server systems will not be able to import this type of key file.



Start *Bitvise SSH Client* to create a communication profile. Enter the name of the SFTP server system (the DNS name) or the numerical IP address in the “Host” field. In the field “Username” use the username provided by the *BusinessMail X.400* Administration. In the field, “Initial method” select the slot of the stored key and in the field “Passphrase” enter the password of the key.



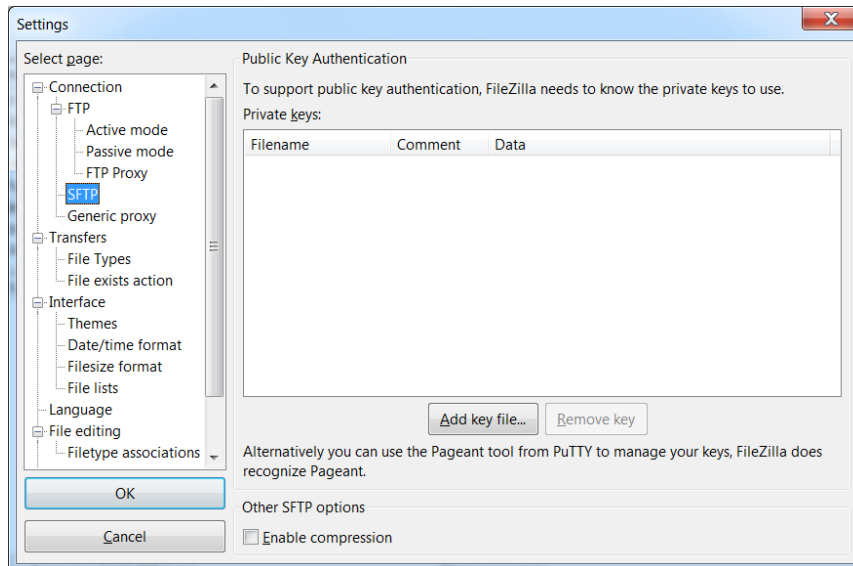
Within the “SFTP” tab, the local directory displayed in the GUI can be configured. A SFTP session can now be established. When first establishing a SFTP session the public key of the SFTP server system must be accepted.

#### ■ FileZilla

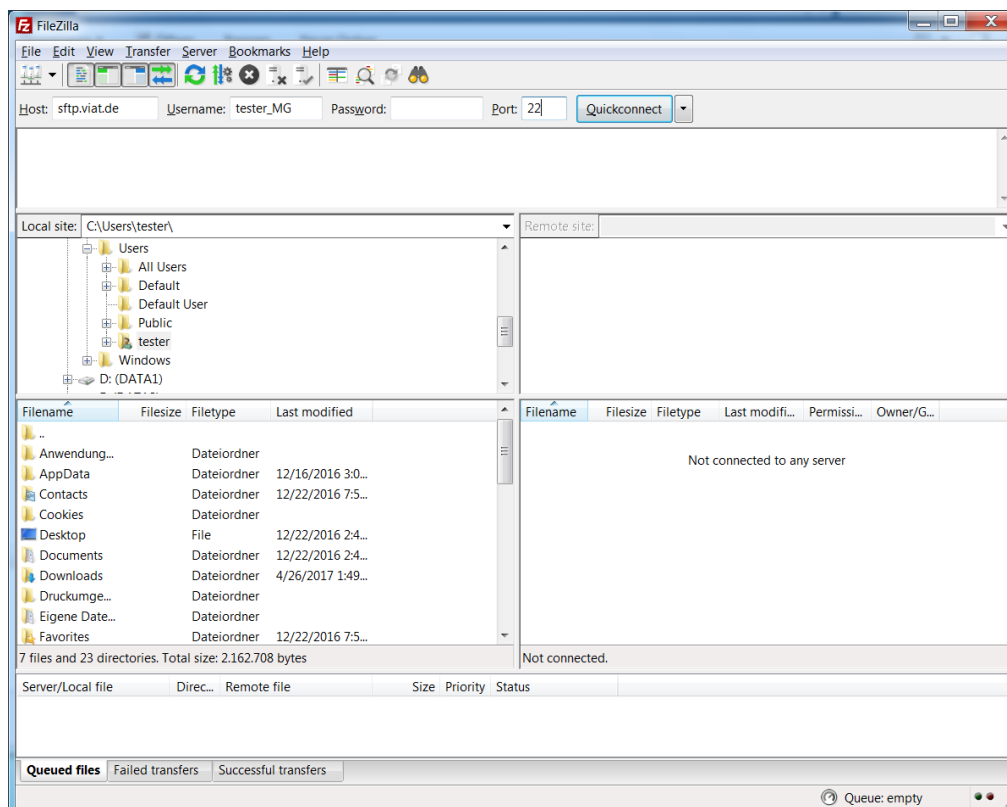
*FileZilla* is an Open-Source client for FTP and SFTP comparable to *WinSCP*.

Configuration:

An appropriate tool (for example *puTTYgen*) is required to create a key pair (Generate). Send the public part of the key to the *BusinessMail X.400* Administration who will store this key on the SFTP server systems (please use the line ending character LF, if necessary, modify the line end using an appropriate Tool, e.g., Notepad ++ when using Windows OS). The private key must be stored locally (Settings → SFTP) for use in *FileZilla*.



Access to the *BusinessMail X.400* MessageGate directory can now be configured. Enter the name of the SFTP server (DNS name) or the numerical IP address in the field "Host" and the username (provided by *BusinessMail X.400* administration) in the field "Username". During the first session, you must accept the public key sent by the SFTP server system.





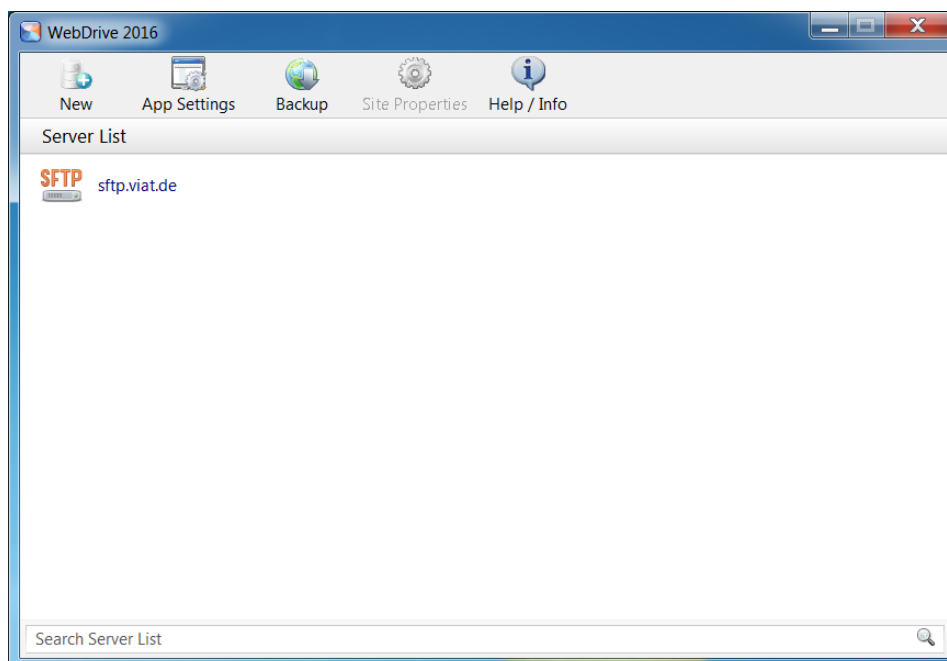
### ▪ WebDrive produced by South River Technologies

Creates a network drive and assigns a local drive via SSH/SFTP (this is also possible via WebDAV, see Chapter 4 Access via HTTPS/WebDAV) and behaves as if it is a local directory.

#### Configuration (in WebDrive 10):

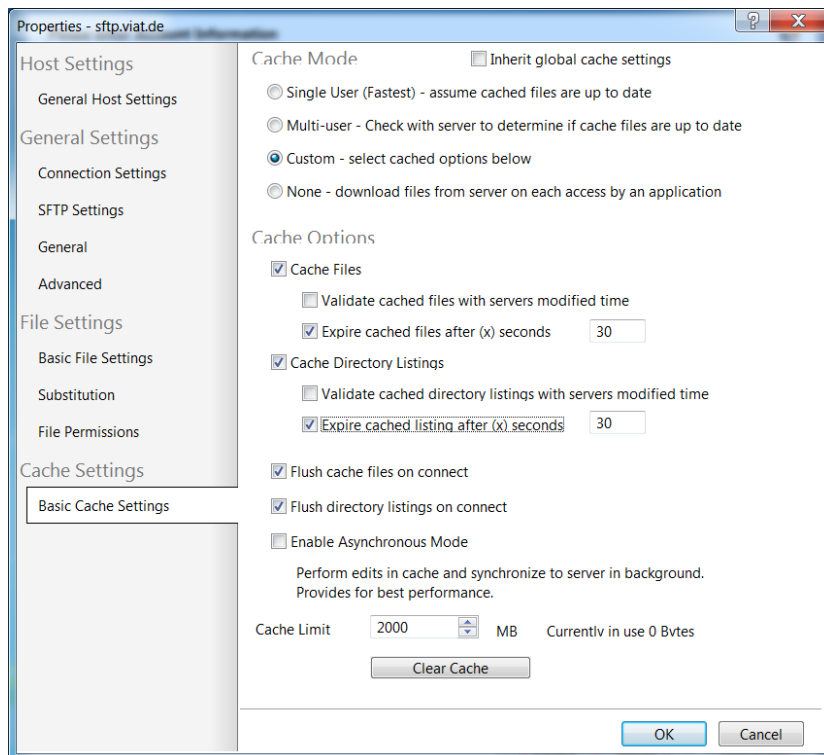
Create a key pair using “Host Key Manager” available in the menu item “App Settings → Security” for example with RSA 3072 bits and store the public part of the key in a file and send this to the *BusinessMail X.400* Administration who will store this key on the SFTP server systems (please use the line ending character LF).

Configure connection → New, select Type “SFTP” and enter the address of the MessageGate directory (sftp.viat.de) into parameter Address/URL and the username (provided by *BusinessMail X.400* administration).

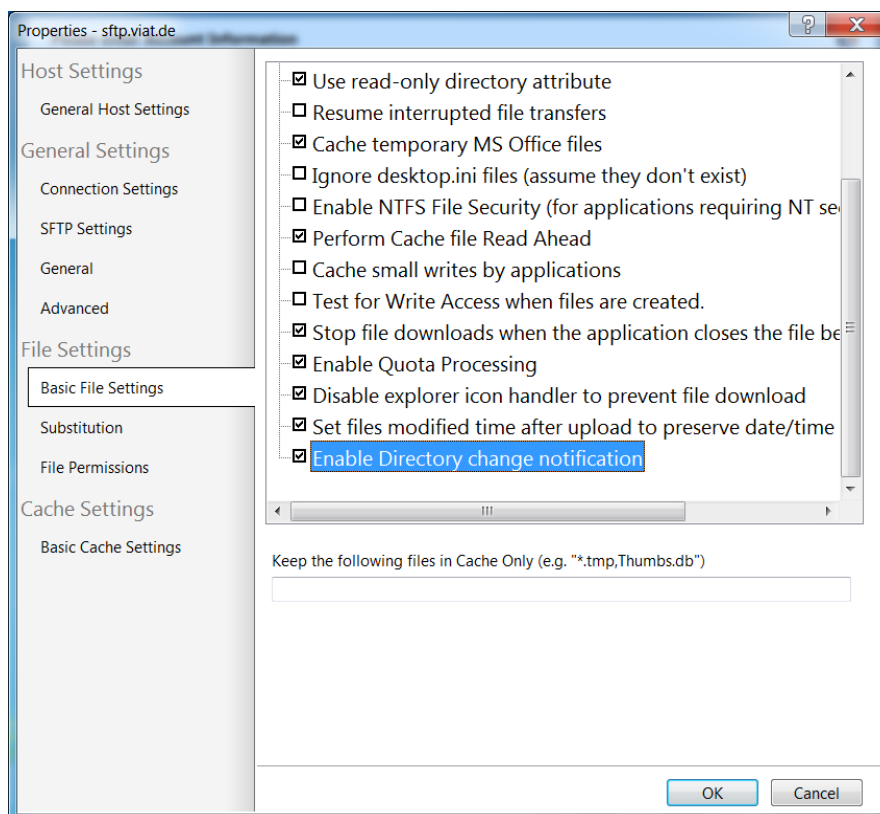


Assign certificate → Site Properties → SFTP Settings → SFTP HostKey → choose “Client Host Key” and enter the password.

Configuration Cache → Site Properties → Cache Settings → Basic Cache Settings  
Site Properties → Cache Settings → “Basic Cache Settings” enable the parameter “Expire cached files after 30 seconds” and “Expire cached listings after 30 seconds”.



Enable the option “Enable Directory change notification” available in “Site Properties → Cache Settings → Basic Cache Settings” so that the client displays the files that MessageGate stores in the directory.

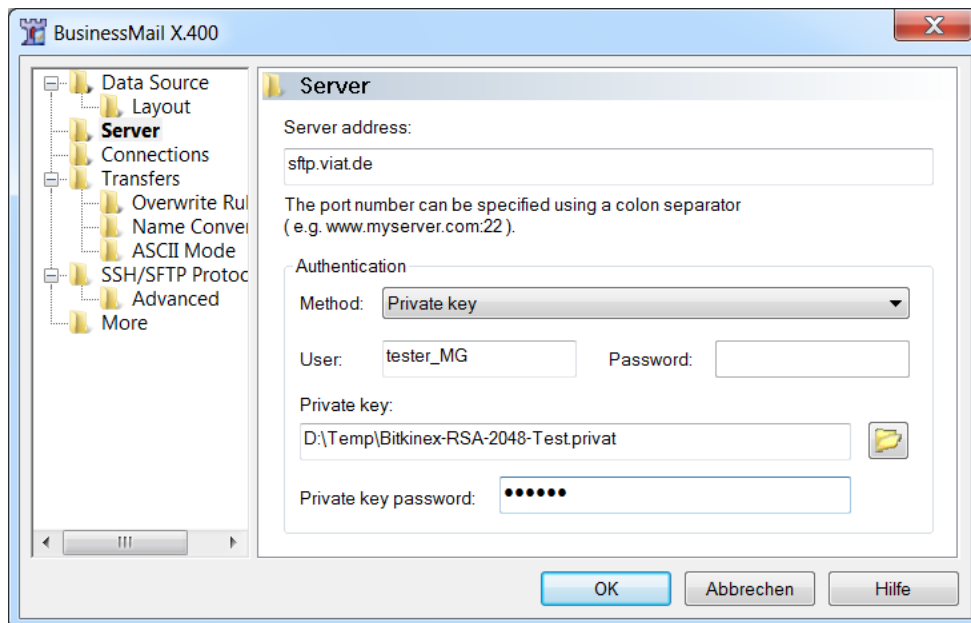


### ▪ BitKinex von BARAD-DUR, LLC

The SFTP (and WebDAV) Client offers a Graphical User Interface and Batch mode, and it can also be run as a Windows service

#### Configuration:

Configure profile → Select http/WebDAV → Right mouse button or Data Source → New → SFTP → Define name → Properties → Server



Server address: sftp.viat.de

Authentication Method: Private key

User: Provided by *BusinessMail X.400* administration

Private key and Private key password: An appropriate tool (for example *puTTYgen*) is required to create a key pair (Generate). Send the public part of the key to the *BusinessMail X.400* Administration (please use the line ending character LF, if necessary, modify the line end using an appropriate Tool, e.g., Notepad ++ when using Windows OS) who will store this key on the SFTP server systems. The private key must be stored locally and add the path of this private key to parameter "Private key". Add the password to parameter "Private key password".

Double-click on the Host Entry of the *BitKinex* File explorer.

BitKinex also has a Command Line Mode (without a GUI) and can be integrated into other applications.

## 3.3.2 Using Microsoft® Windows 64 Bit Operating systems

All products described in the last Chapter will also run-on Windows 64-bit systems. **South River Technologies** will offer newer versions of *WebDrive* only as a native 64bit version (current version is WebDrive NextGen 1.1.16).

Be aware, that when using WebDrive Next Generation (tested in 1.1.13, 1.1.14 and 1.1.16), unlike the older versions (2019, 2016 etc.) you must now enter the value "/" for the configuration option "Default directory" (normally this is not necessary, because the login is the home directory). South River plan to fix this in a later version.

### 3.3.3 Using Linux and UNIX Operating systems

- **SFTP as part of the OpenSSH suite**

A key pair using the program `ssh-keygen` needs to be created. For example, the command “`ssh-keygen -t rsa -b 3072`” will generate a 3072-Bit RSA key to use with SSH V2. The key files `id_rsa` (private key) and `id_rsa.pub` (public key) are then stored in the hidden subdirectory “`/.ssh`” of the respective user directory. Before the user’s public key is sent to the *BusinessMail X.400* administration, it needs to be converted using the following command:

```
# ssh-keygen -e -f ~/.ssh/id_rsa.pub > ~/.ssh/ssh_XXXXXX.pub
```

where `XXXXXX` is the User-ID of the MessageGate account. To connect to the SFTP host the following command is used:

```
# sftp username@ssh_host_name" (username of the VMS account!).
```

During the first session the key of the SSH host needs to be accepted.

- **FileZilla**

*FileZilla* is an Open-Source client for FTP and SFTP comparable to *WinSCP*.

Configuration:

The configuration of this client is equivalent to the Windows version (see Chapter 3.3.1 Using Microsoft® Windows 32 Bit Operating systems). The newer versions of this client also include a key generator called `fzputtygen`.

### 3.3.4 Using Apple Mac OS X

- **SFTP as part of the OpenSSH suite using the terminal program**

The OpenSSH suite and the included SFTP client can only be run in the context of a terminal session. A key pair using the program `ssh-keygen` needs to be created. For example, the command “`ssh-keygen -t rsa -b 3072`” will generate a 3072-Bit RSA key to use with SSH V2. The key files `id_rsa` (private key) and `id_rsa.pub` (public key) are then stored in the hidden subdirectory “`/.ssh`” of the respective user directory. Before the user’s public key is sent to the *BusinessMail X.400* Administration it needs to be converted (please use the line ending character LF) using the following command:

```
# ssh-keygen -e -f ~/.ssh/id_rsa.pub > ~/.ssh/ssh_XXXXXX.pub
```

where `XXXXXX` is the User-ID of your MessageGate account. To connect to the SFTP host use the command

```
# sftp username@ssh_host_name" (username of the VMS account!).
```

During the first session the key of the SSH host needs to be accepted.

- **FileZilla**

*FileZilla* is an Open-Source client for FTP and SFTP comparable to *WinSCP*.

Configuration:

The configuration of this client is equivalent to the Windows version (see Chapter 3.3.1 Using Microsoft® Windows 32 Bit Operating systems). The newer versions of this client also include a key generator called `fzputtygen`.

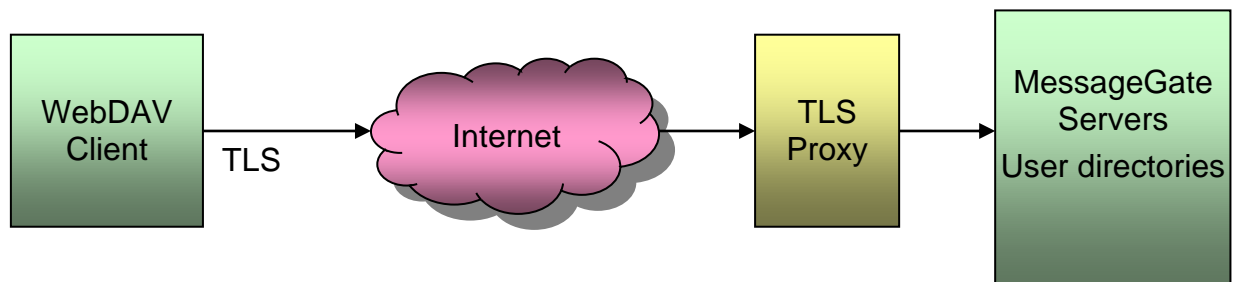
### 3.3.5 Using other operating systems

There are ports of the OpenSSH suite available for the IBM operating systems *i5/OS* now called *IBM i* for Business and *zOS*, so the description in Chapter 3.3.3 Using Linux and UNIX Operating system should also apply to these operating systems. However, these assumptions cannot be verified, as these operating systems are not available in our test environment.

## 4 Access via HTTPS/WebDAV

### 4.1 General Information

An alternative to the SFTP access described in the last Chapter is the access via HTTPS using WebDAV extension. HTTPS/WebDAV is the recommended communication protocol for those customers using Internet to access their MessageGate directory. Only connections secured by TLS V1.2 (min. 128 Bit key length, server supports 256 Bit AES) and higher via the Internet will be accepted. The access via HTTPS/WebDAV access provides comparable functionality to the FTP command set but with the inherent advantage that it does not require two TCP/IP (management and data) channels for a session. In addition, several WebDAV modules provides the inherent advantage that remote disks on the WebDAV server system appear to an application as local disks. Hence, the application does not have to be specially adapted to provide the communication environment.



To access the directory, use the following URL:

<https://webdav.viat.de/~00000nnnnn/>

where nnnnn is the User-ID (*BusinessMail X.400* internal identifier of the X.400 user-account). The forward slash “/” at the end of URL is necessary, otherwise the connection will fail.

To establish a TLS connection to the application server it is necessary that the WebDAV client sends a client certificate in response to the TLS proxy request. You may generate the necessary certificate with private key using the CA of *WebConfig* (download the WebConfig CA certificate using Service URL: <https://www.service-viat.de> in Section “WebConfig & X.400-App”) in menu item “Certificate Management – Create” and download the PKCS12 file in menu item “Certificate Management – View/Download”. Be aware that this Client certificate will be available on the proxy server the next day. If your application requires a separate certificate and key file, you must extract them out of the PKCS12 file using a suitable tool, e.g., OpenSSL. Here are examples when using OpenSSL:

Export key without password: `openssl pkcs12 -in <name>.p12 -out <name>_key.pem -nodes -nocerts`

Export key with password: `openssl pkcs12 -in <name>.p12 -out <name>_key.pem -nodes`

Export certificate: `openssl pkcs12 -in <name>.p12 -out <name>_key.pem -nodes -nokeys -clcerts`

If your security policy requests the use of a certificate signed by an official CA it is also possible to provide you such a certificate (ZIP file including PKCS12 file and a separate certificate and key file using PEM format). Use *WebConfig* to download this ZIP file in the menu item “WebConfig Management - Downloads”.

The downloaded Client certificate and the private key must be imported into the certificate store of your WebDAV application. By default, there is a passphrase to secure the private key in the P12/PFX file (see the text file associated with the certificate) because most existing WebDAV clients request a passphrase. If you are using a WebDAV solution that has problems using passphrases, then please make *BusinessMail X.400* administration aware of this issue when you request your new account. In such cases, a file can be generated without a passphrase. Please be aware that not all WebDAV solutions support the use of Client certificates. Chapter 4.3 contains information about successfully tested client solutions and associated libraries, which support Client certificates.

When configuring the MessageGate directory there is a choice to use this Client certificate also for the authentication to directly log into the Web Server of the WebDAV access or additionally a username and password will be prompted and verified.

## 4.2 Features to note

Customers who only want to receive/fetch messages (for example applications that only receive electronic invoices sent by telecommunication companies) do not need a communication module that supports the WebDAV protocol. These files can be downloaded using a standard Web browser. The messages/data delivered to the file interface are purged automatically based on a preconfigured purge time. Customers who need to send messages require a full function WebDAV solution.

A customer who orders MessageGate with reduced functionality will only recognize that a message has been rejected if the automatically generated status report is configured in the host profile or if he requests a status report within *WebConfig* GUI.

If you upload message files, please consider that the MessageGate process will assume, that the content is ISO-Latin-1/ANSI encoded. If your OS uses for example UTF-8 encoding (e.g., Windows 10 starting with Rev. 1903) while storing files, there might be a mapping problem when sending some special characters (e.g., German umlauts) in the subject of the message (X.400 uses T.61 character set).

The following sections include information about a selection of communication modules with which MessageGate access has been successfully tested.

## 4.3 Recommended WebDAV Clients

### 4.3.1 Using Microsoft® Windows 32 Bit Operating systems

- **Microsoft® Windows Explorer (Windows 2000 and newer)**

Configure a new Web Folder in my Network place.

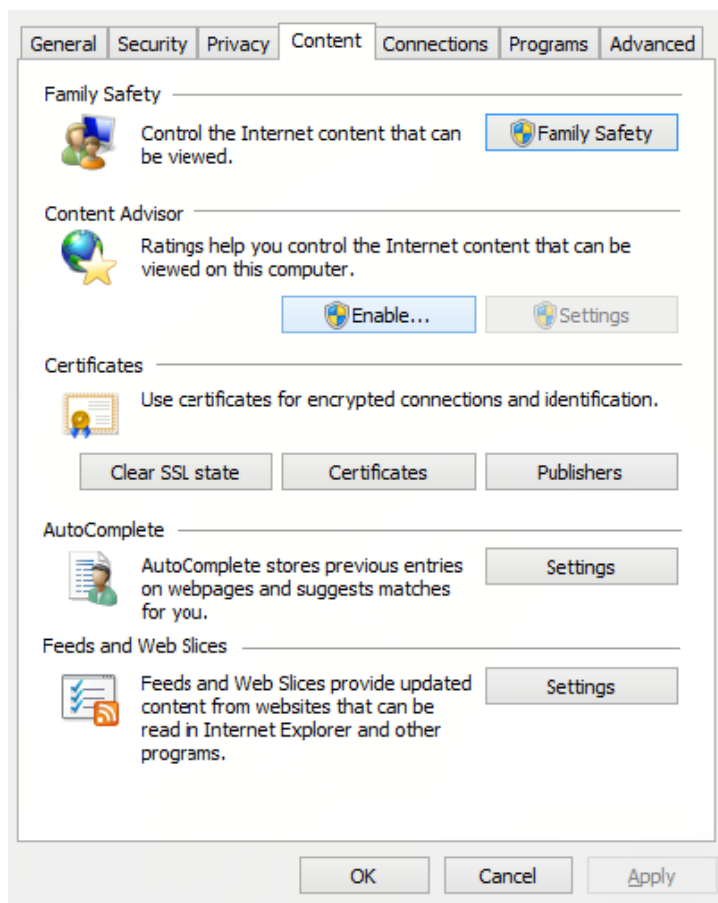
When using Windows 2003 Server and IE7.0 or Windows Vista you may need to install the Patch KB907306 otherwise the connection can fail. In Windows 2003 Server, you may also have to start Web client service because this service is set by default to “manually”. For the configuration of net drives in Windows 7 and newer Windows OS, it is necessary to use the authentication method client certificate. Please request this authentication method when ordering your WebDAV account.

Configuration:

My Network place → Add a network place → [https://webdav.viat.de/~00000... /](https://webdav.viat.de/~00000.../)  
or when using newer Windows OS (Windows 7 or newer):

Map Network Drive → Connect to a Web site that you can use to store your documents and pictures → Choose a custom network location → enter in field Internet or network address the URI <https://webdav.viat.de/~00000...>

Requirement: Import certificate (\*.p12) in Windows cert store for example open certificate file in explorer or using IE Explorer → Tools → Internet options → Content → Certificates → Import or using mmc (run mmc.exe) and add Snap-Ins → Certificates → My User account (or Computer account), start this Snap-in and select Personal → All tasks → Import.



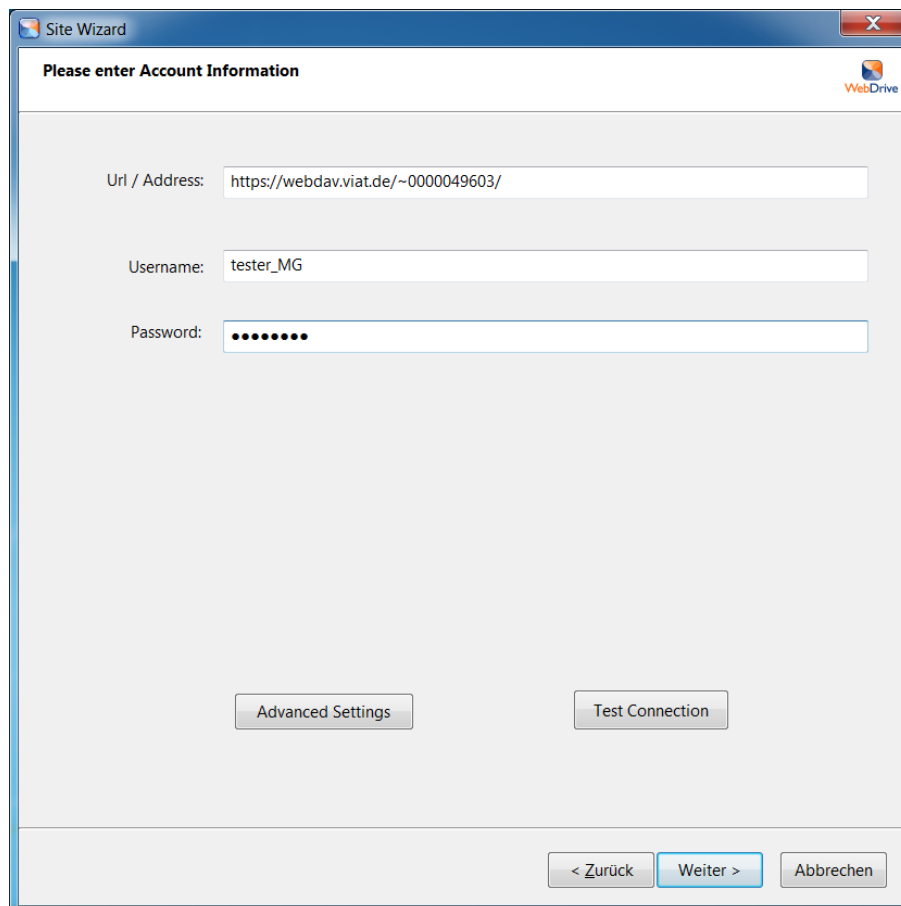


- **WebDrive produced by South River Technologies**

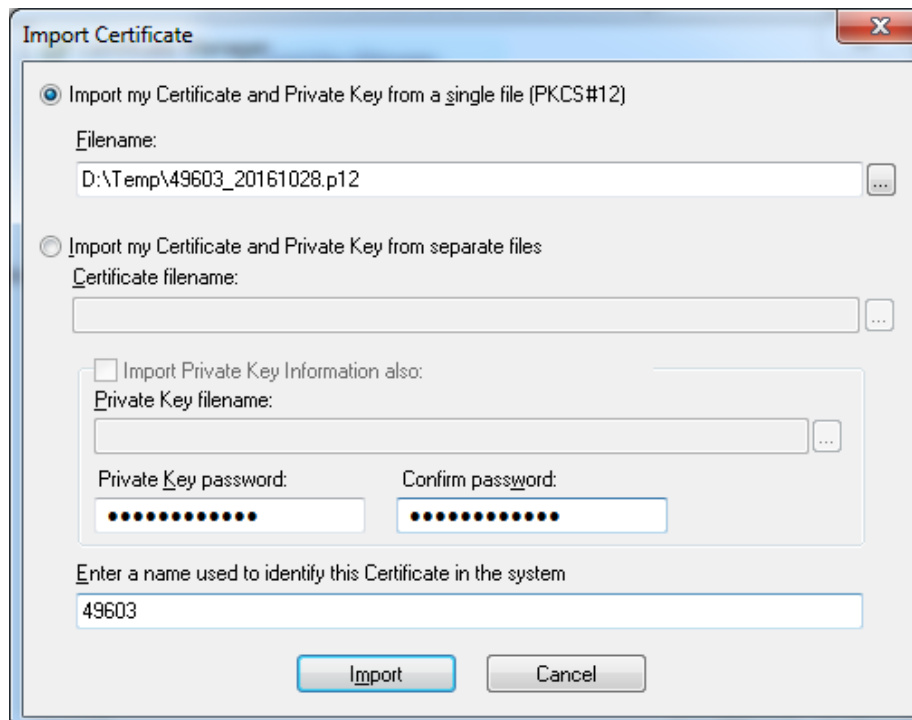
Creates a network drive and assigns a local drive via WebDAV (and via SFTP) to make the remote directory appear as a local directory.

Configuration (in WebDrive 10):

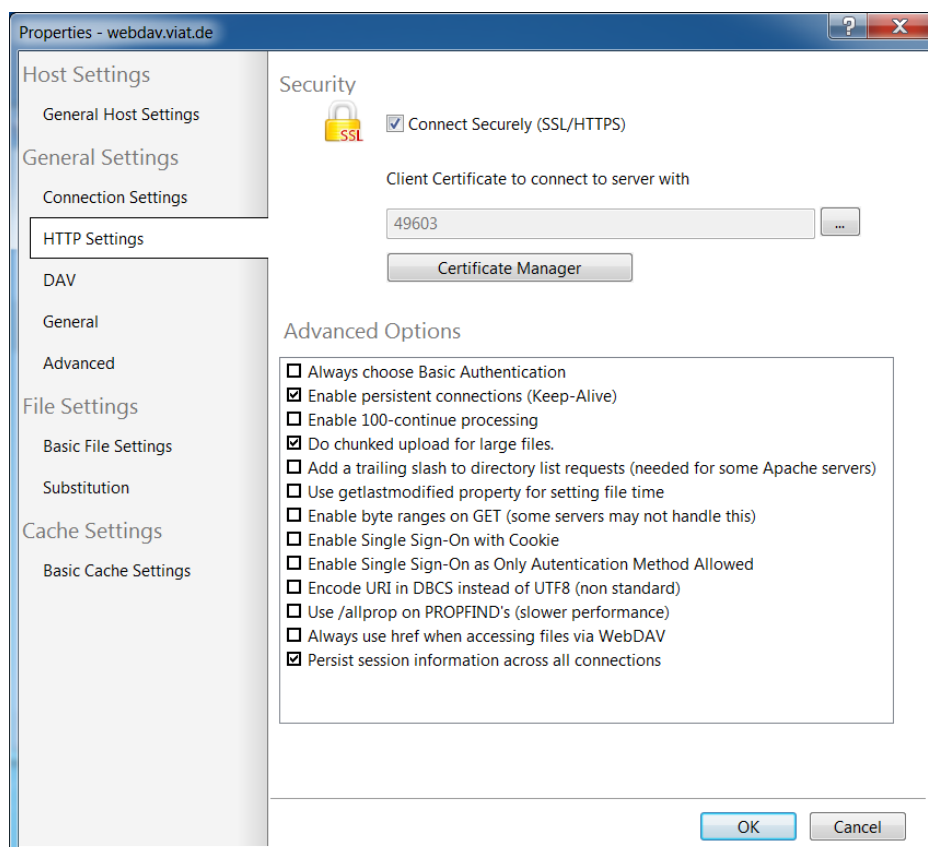
Configure connection → New, select Type “Secure WebDAV” and enter the address of the MessageGate directory (“https://webdav.viat.de/~00000xxxxx/”, where xxxxx is the User-ID) into parameter Address/URL and the Username /Password (provided by *BusinessMail X.400* administration).

The image shows a screenshot of a 'Site Wizard' window titled 'Please enter Account Information'. The window has a blue header bar with the title and a close button. Below the header, there is a 'WebDrive' logo in the top right corner. The main area contains three input fields: 'Url / Address:' with the value 'https://webdav.viat.de/~0000049603/', 'Username:' with the value 'tester\_MG', and 'Password:' with a masked password represented by dots. At the bottom of the main area, there are two buttons: 'Advanced Settings' and 'Test Connection'. At the very bottom of the window, there are three navigation buttons: '< Zurück', 'Weiter >', and 'Abbrechen'.

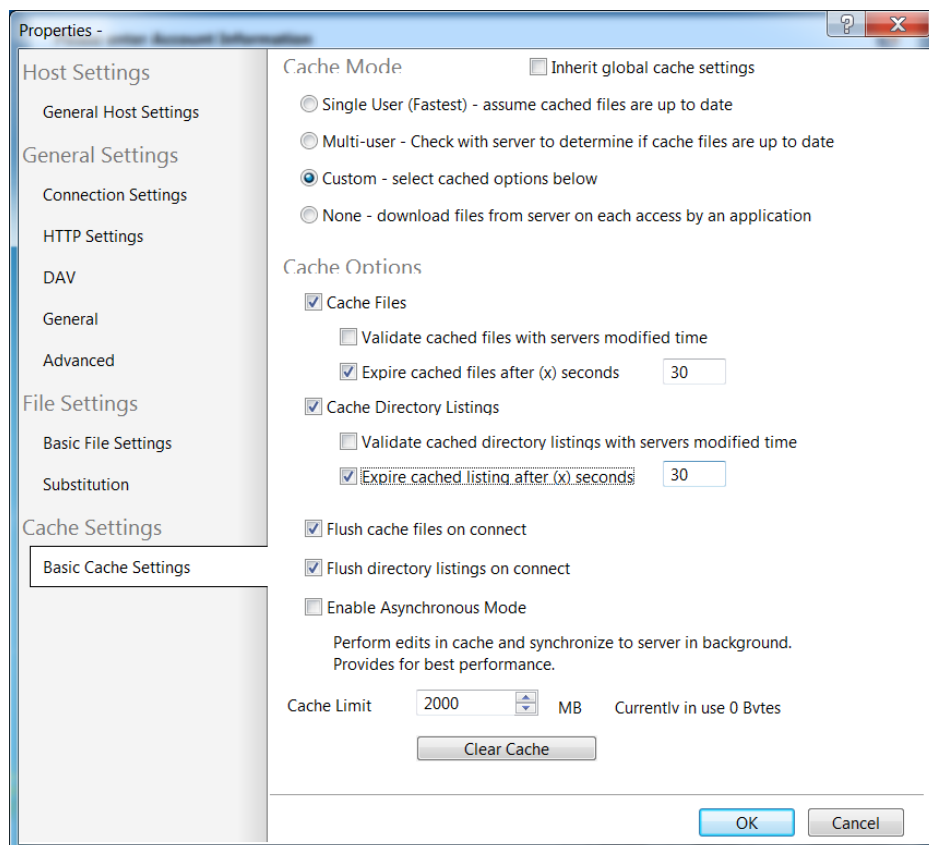
Import of certificate: “→ Advanced Settings → HTTP Settings → Certificate Manager → Import → Import ... from single file (PKCS#12):” and enter the PKCS12 file, the password of the file and assign a name for this certificate.



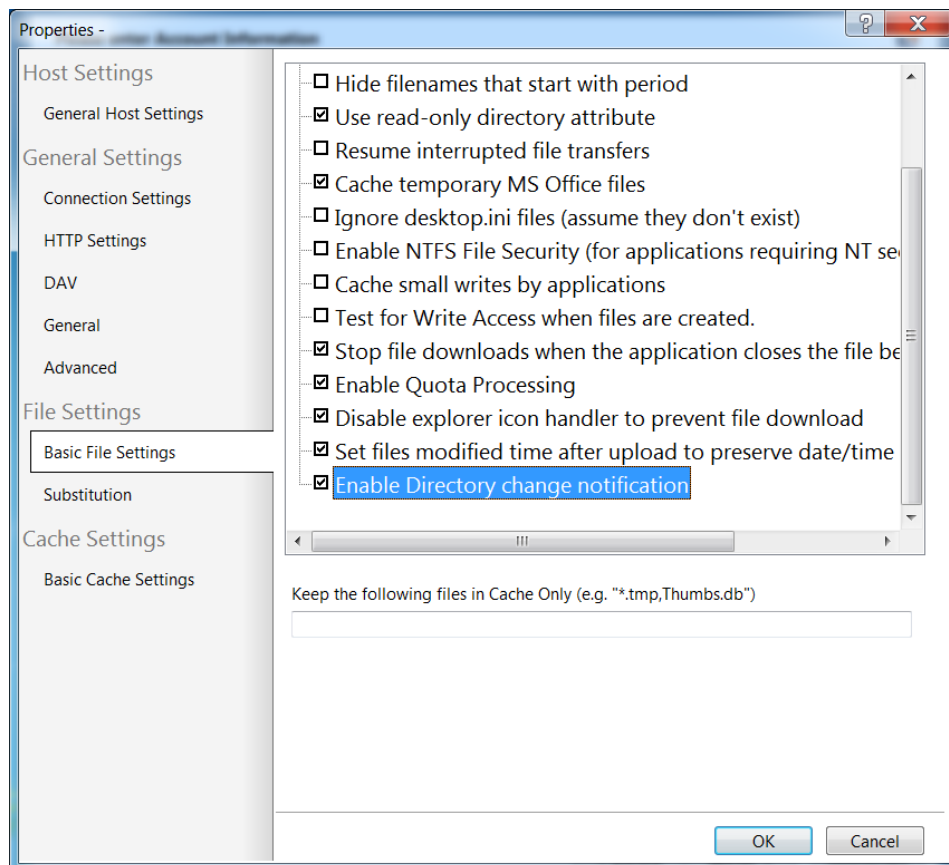
Choose HTTP settings in Properties of site



Configuration Cache → Program Setting → Cache Settings → Options → Custom →  
Expire cached files after 30 seconds und Expire cached listings after 30 seconds



Enable the option “Enable Directory change notification” so that the client automatically displays the files that MessageGate places in the directory.

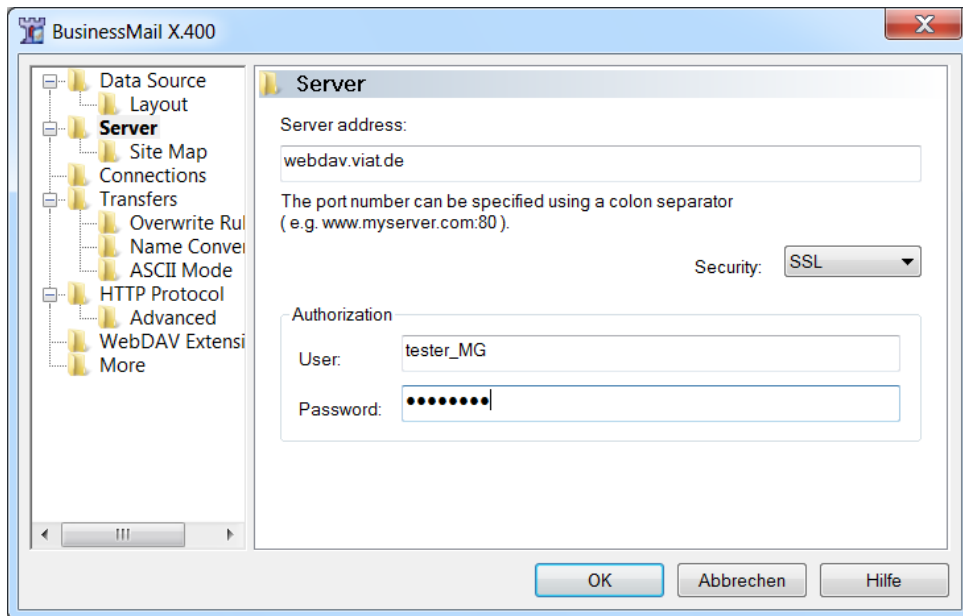


▪ **BitKinx produced by BARAD-DUR, LLC.**

WebDAV Client (also supports SFTP) provides Graphical User Interface and a Batch mode option, may also run as a Windows service.

Configuration:

Configure connection → Select http → Right mouse click or Data Source → New → http/webdav → Assign name → Properties → Server

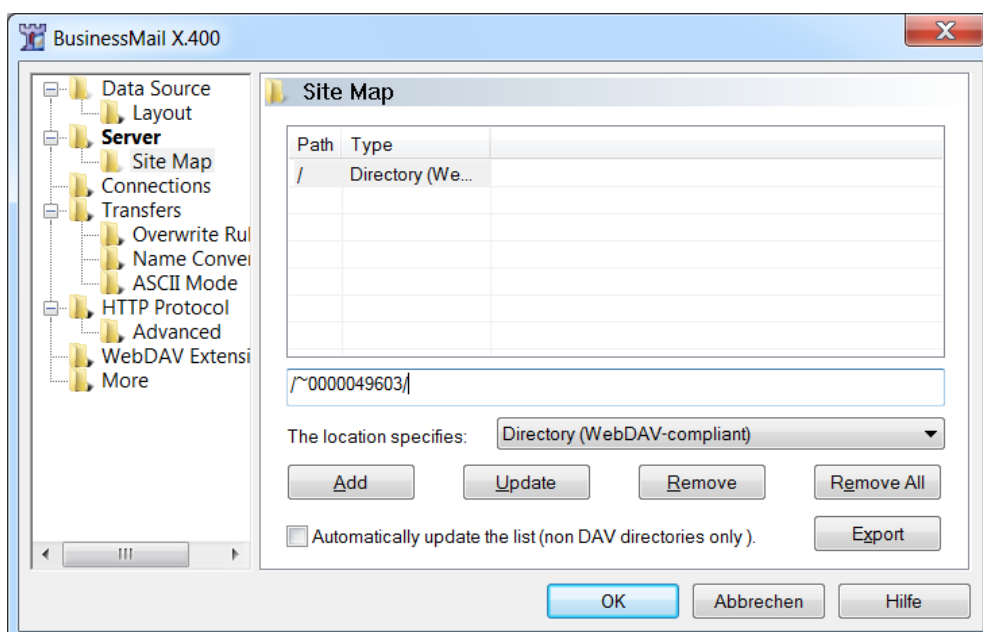


Server address: webdav.viat.de

Security: SSL

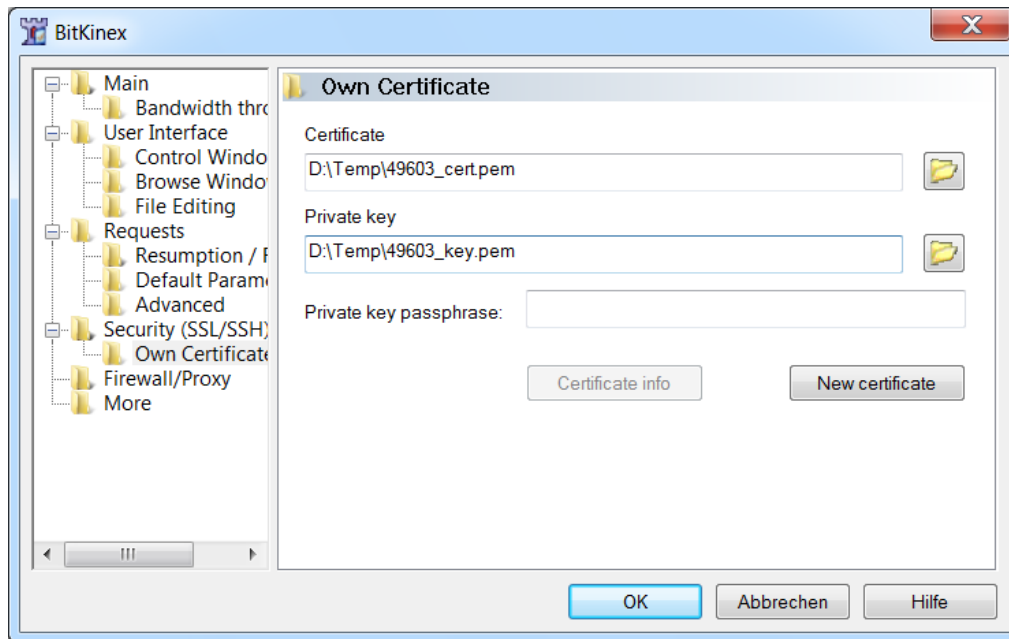
User + Password: Provide login data

→ Site Map



Path: /~00000nnnnn/ → (nnnnn is User-ID)

Configure certificate → File → Option → Security → Own certificate



Extract certificate and key file from the ZIP archive or PKCS12 file and enter the path for both files and the passphrase for the key file.

Double-click on the host entry to open *BitKinex* FileManager.

BitKinex also provides a command line mode (no GUI) so it would be possible to integrate the module into existing EDI solutions.

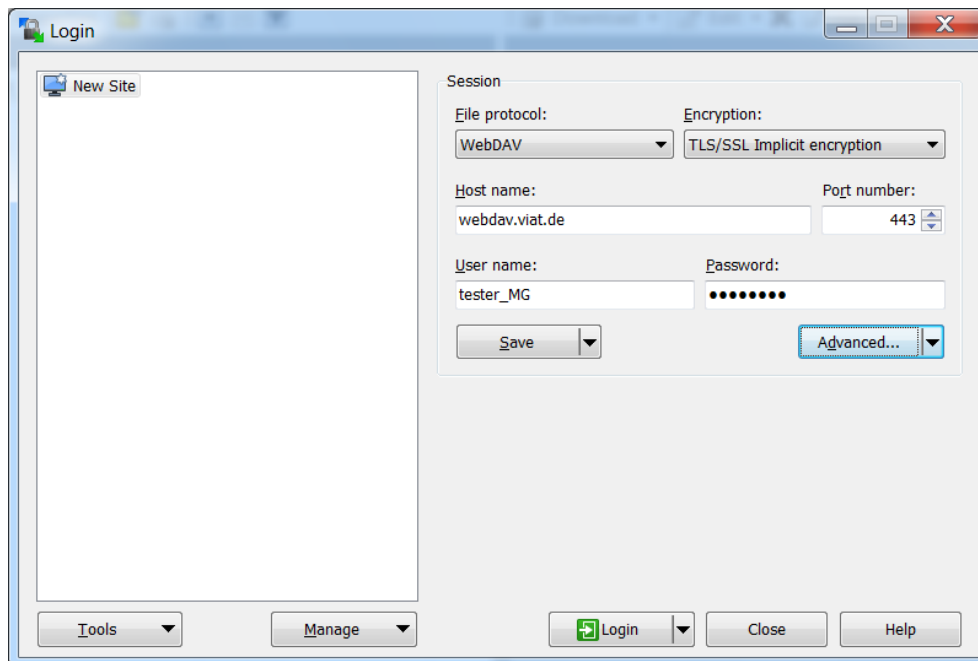
#### ■ WinSCP

*WinSCP* is an Open-Source client providing https/WebDAV capabilities (also SFTP and FTP) in combination with a Graphical User Interface to transfer and to download data to a Web server.

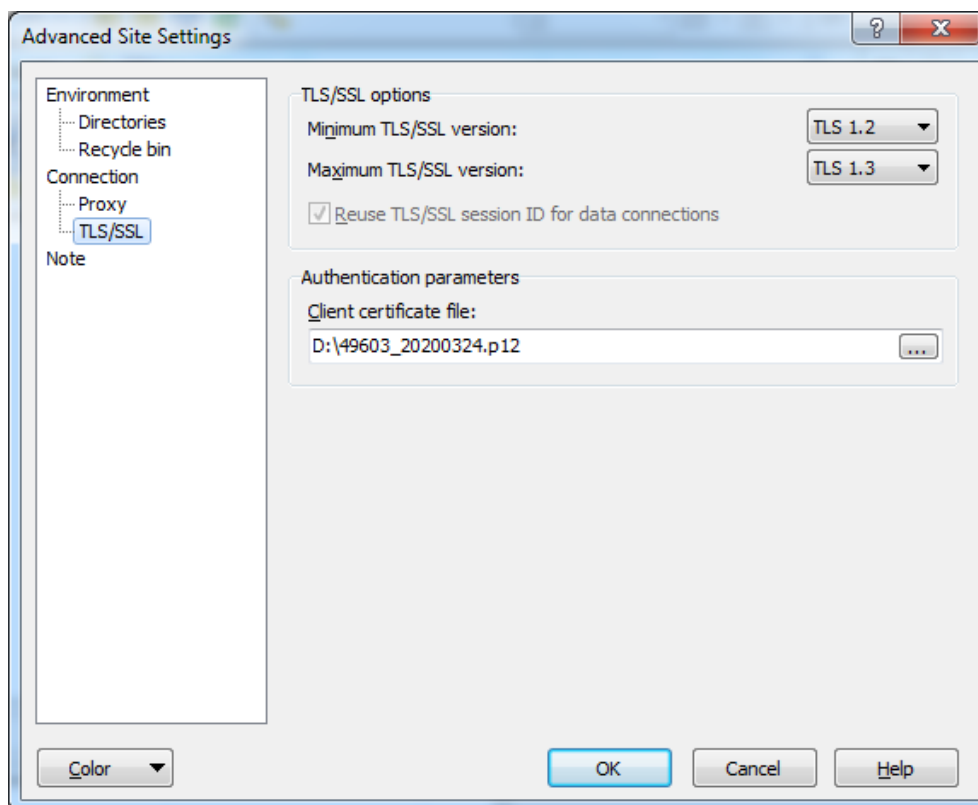
*WinSCP* also offers a Batch mode and a command line interface running in a DOS window with a command set comparable to those of a standard FTP client to transfer data automatically.

#### Configuration:

Start *WinSCP* and create a “New Site” while selecting File protocol “WebDAV”. Enter the Host name (WebDAV server name - DNS name) or the numerical IP address (provided by the *BusinessMail X.400* administration) and the provided Username and password.



Use “Advanced... /Advanced...” to select in the menu item “Connection / TLS/SSL” the provided PKCS12 file and store this change when leaving the menu “Advanced...”.



After finishing this configuration, you may log into the https server for the first time. While connecting *WinSCP* will ask for the password of the PKCS12 file (can be imported via file in batch mode).

In menu item “Advanced...” you can tailor *WinSCP* to your requirements.

- **Mozilla Firefox for Windows with Add-on “Open as Webfolder 0.22”**

The Add-On “Open as Webfolder” uses the Windows Explorer functionality to upload files to MessageGate and to delete files in the directory. The better approach would be to use Windows Explorer directly. There is no WebDAV Add-On for the Linux version of Firefox.

- **Onion (WebDAV C++ Library)**

There is no active support for this library at the present time.

- **Neon (WebDAV C Library, most WebDAV Clients were developed based on this library)**

Library, last update September 2021 (see <https://notroj.github.io/neon/>).

### 4.3.2 Using Microsoft® Windows 64 Bit Operating systems

All products described in last sections will also run on the 64-bit Windows. South River Technologies offers newer versions of WebDrive only as a native 64-bit version (current version is WebDrive NextGen 1.1.16).

### 4.3.3 Using Linux and UNIX Operating systems

- **Cadaver**

*Cadaver* is command line-based Client which provides almost the same functionality (commands, syntax) as provided in FTP command line-based Clients for Linux and Microsoft® Windows.

One has to first import the Client certificate using the command “set client-cert certificate-name.p12”. When connecting to MessageGate the *Cadaver* Client initially prompts for the certificate passphrase. Consider obtaining a certificate without a passphrase from *BusinessMail X.400* administration.

- **Konqueror/Dolphin (Web und Directory Browser for Linux)**

Assign address “webdavs://webdav.viat.de:443/~xxxxxx”.

In older Linux versions, you may encounter problems with IP V6 support. If this is the case, you should disable this feature.

Configuration:

Requirement: Import Client certificate in Setting → Configure Konqueror → Encryption → Your certificate.

You now must configure this certificate to be the default or select that *Konqueror* always prompts for a client certificate.

Newer Linux distributions (SuSE and 9.3 and newer) include features to configure a web folder using Konqueror.

In SuSE Linux, select Network browser → Add network folder → Web folder:

- Enter a name for this folder
- Enter the username of your WebDAV Accounts
- Enter server “webdav.viat.de”
- Enter Port 443

- Enter folder “~00000nnnnn” where nnnnn is User-ID of your MessageGate Account
- Encrypt session

- **davfs2 (File system for Web folders based on WebDAV protocol)**

Installation kit for Debian and Ubuntu but after installation the required libraries davfs2 will also run on SuSE Linux.

Required libraries davfs2 will also run on SuSE Linux.

For a separate installation description see <https://www.service-viat.de>

**Please be aware that the file system must be unmounted before a server shut-down!**

- **Sitecopy (Program to map the content of local directory into MessageGate directory and vice versa)**

Starting Version 0.16.3 the configuration might be:

```
***
site webdav
server webdav.viat.de
protocol webdav
username <Username>
password <Password>
client-cert </path/to/cert.p12>          #in man-page not listed!
remote /~<User-ID>/
local /<local path>/
http secure
```

- **Onion (WebDAV C++ Library)**

There is no active support for this library at the present time.

- **Neon (WebDAV C Library, most WebDAV Clients are developed based on this library)**

Library, last update September 2021 (see <https://notroj.github.io/neon/>).

#### 4.3.4 Apple iOS

- **WebDAV Nav+ (commercial App for power user, the free version WebDAV Nav does not support the use of a client certificate)**

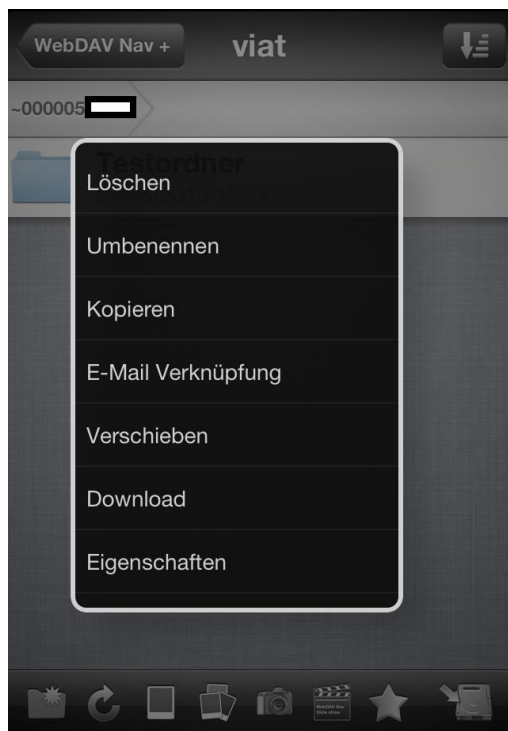
Use the iPhone App Store to download and install the app. A new profile using the login values supplied by the *BusinessMail X.400* administration needs to be created.





Next, the PKCS12 file for the authorization of the connection for the SSL proxy needs to be placed in the root directory using the name of the profile. In our example, this would be the file name "viat.p12".

Now you should be able to log into your MessageGate directory and to upload or download files. The following picture shows the available options.



### 4.3.5 Using other operating systems

- **Neon (WebDAV C Library, most WebDAV Clients were developed based on this library) available for Unix OS**

Library, last update September 2021 (see <https://notroj.github.io/neon/>).

- **Sitecopy (Program to map the content of local directory into MessageGate directory and vice versa)**

Starting Version 0.16.3 the configuration might be:

\*\*\*

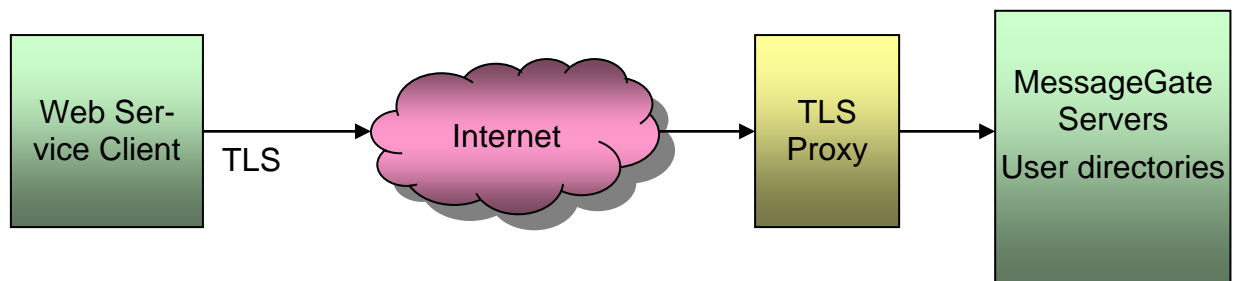
```
site webdav
server webdav.viat.de
protocol webdav
username <Username>
password <Password>
client-cert </path/to/cert.p12>          #in man-page not listed!
remote /~<User-ID>/
local /<local path>/
http secure
```

For editorial reasons this page is empty!

## 5 Access via HTTPS/Web Services

### 5.1 General Information

In addition to the access mechanisms described in the last two Chapters, a REST based Web Service has been introduced in 2017 as a new method to access the MessageGate File Interface. HTTPS is the recommended communication protocol for those customers using the Internet to access their MessageGate directory. Only TLS1.2 connections (min. 128 Bit key length, server supports 256 Bit AES) and higher via the Internet will be accepted. Beside the advantages of HTTPS in conjunction with WebDAV (providing a better network security integration than FTP) Web Services has the potential of providing better and easier application integration if needed.



To access the MessageGate user directory use the following URL:

<https://webdav.viat.de/mgate/vn/~00000nnnnn/>

where “nnnnn” is the User-ID (*BusinessMail X.400* internal identifier of the X.400 user-account) and the “n” in vn identifies the Web Service API version. All profiles support the requests PUT (upload a message file, a Transmissions Set file, a receipt notification file, or a status request file), GET (download a file list, a message file, a Transmissions Set file or a status report) und DELETE (delete a message file, a Transmissions Set file or a status report). The forward slash “/” at the end of URL is necessary, otherwise the connection will fail.

To establish a TLS connection to the application server it is necessary that the Web Service client sends a client certificate in response to the TLS proxy request. You may generate the necessary certificate with a private key using the CA of *WebConfig* (download the WebConfig CA certificate using Service URL: <https://www.service-viat.de> in Section “WebConfig & X.400-App”) in the menu item “Certificate Management – Create” and download it in menu item “Certificate Management – View/Download”. Be aware that this Client certificate will be available on the proxy server the following day.

If your security policy requires the use of a certificate signed by an official CA it is also possible to provide you with such a certificate with appropriate private key (ZIP file including PKCS12 file and a separate certificate and key file using PEM format). Use *WebConfig* to download this file in the menu item “WebConfig Management - Downloads”.

The downloaded Client certificate and the private key must be imported into the certificate store of your Web Service application. By default, there is a passphrase to secure the private key in the P12/PFX file (see the text file associated with the certificate) because most existing WebDAV clients request a passphrase. If you are

using a Web Service solution that has problems using passphrases, then please make *BusinessMail X.400* administration aware of this issue when you request your new account. In such cases, a file can be generated without a passphrase. When configuring the MessageGate directory you have a choice whether this Client certificate is also used for WebDAV authentication or whether username and password authentication should be used.

## 5.2 Features to note

The Web Service access was designed to handle different profiles based on scripts providing a high level of flexibility to fit customer requirements. At present, the Web Service profiles 1 (v1) and 2 (v2/v2a) are available. The profile v1 is a 1:1 mapping of existing MIME structures (content types) into access directory, while the profile v2 (v2a) will use JSON structure to transfer the data.

If you upload message files, please consider that the MessageGate process will assume, that the content is ISO-Latin-1/ANSI encoded. If your OS uses for example UTF-8 encoding (e.g., Windows 10 starting with Rev. 1903) while storing files, there might be a mapping problem when sending some special characters (e.g., German umlauts) in the subject of the message (X.400 uses T.61 character set).

## 5.3 Web Service API

Method	Request <a href="https://webdav.viat.de/mgate/">https://webdav.viat.de/mgate/</a>	Description
Upload file	PUT .../v1/~00000nnnnn/<file name> Content-Type: message/rfc822 PUT .../v1/~00000nnnnn/<file name> Content-Type: text/plain PUT .../v2/~00000nnnnn/<file name> Content-Type: application/json PUT .../v3/~00000nnnnn/<file name> Content-Type: application/json	Upload file: <ul style="list-style-type: none"> <li>• Message</li> <li>• Transmission Set</li> <li>• Status Report Request</li> <li>• Receipt Notification</li> </ul>
List dir	GET .../v1/~00000nnnnn/*.out Content-Type: text/plain GET .../v1/~00000nnnnn/M_* Content-Type: text/plain GET .../v1/~00000nnnnn/*.in Content-Type: text/plain GET .../v2/~00000nnnnn/*.out Content-Type: application/json GET .../v2/~00000nnnnn/M_* Content-Type: application/json GET .../v2/~00000nnnnn/*.in Content-Type: application/json GET .../v3/~00000nnnnn/*.out Content-Type: application/json	A List of all files <ul style="list-style-type: none"> <li>• with extension “.out”</li> <li>• starting with „M_”</li> <li>• with extension “.in”</li> </ul>

	GET .../v3/~00000nnnnn/M_* Content-Type: application/json GET .../v3/~00000nnnnn/*.in Content-Type: application/json	
Get file	GET .../v1/~00000nnnnn/<file name> Msg: Content-Type: message/rfc822 TS: Content-Type: text/plain SR: Content-Type: text/plain, text/csv-c, or text/csv-s GET .../v2/~00000nnnnn/<file name> Msg: Content-Type: application/json TS: Content-Type: application/json SR: Content-Type: application/json GET .../v3/~00000nnnnn/<file name> Msg: Content-Type: application/json TS: Content-Type: application/json SR: Content-Type: application/json	Download file: <ul style="list-style-type: none"> <li>• Msg: Message</li> <li>• TS: Transmission Set</li> <li>• SR: Status Report</li> </ul>
Delete file	DELETE .../v1/~00000nnnnn/<file name> DELETE .../v2/~00000nnnnn/<file name> DELETE .../v3/~00000nnnnn/<file name>	Delete a file

### 5.3.1 The Web Service profile v1

The profile offers the following features:

#### PUT - Upload a file

The method "PUT" is used to upload a new file to the MessageGate directory. The file content is included in the content of the PUT request and the (unambiguous) file name must start with "M", "T", "R" or "S" followed by an underscore "\_" character. The file extension must be ".in" or if there is no file name extension, the ".in" will be added automatically. The file name is case insensitive. The status "HTTP 404 Not found" with the extended information "Invalid filename format" will be returned if those requirements are not fulfilled.

If the HTTP header information "Content-Type" is used within the PUT request it must start with "text" when uploading "T\_", "R\_" and "S\_" files. When uploading a "M\_" file the "Content-Type" should be "message/rfc822" or start with "application/" (e.g., application/octet-stream). The status "HTTP 406 Not Acceptable" with extended information "Illegal combination of filename and Content-Type" will be returned if those requirements are not fulfilled.

Binary data is not allowed when uploading files with content-Type "text/...". If a file with a defined file name does not exist, it will be created, and the status "HTTP 201 Created" will be returned. The HTTP header information "Location" will show the URL of the file. If a file(name) already exists, the upload will be rejected with the status

“HTTP 423 Locked” with the extended information “File currently locked” will be returned. Only “complete” files can be uploaded, so the HTTP header information “Content-Range” cannot be used and will be rejected with the status “HTTP 400 Bad Request” with extended information “Content-Range not allowed” will be returned.

**Example a message file (M\_) upload using curl:**

```
curl https://webdav.viat.de/mgate/v1/~0000049640/ -X PUT -T M_161031_001.in -H "Content-Type: message/rfc822" --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

The “/” at the end of the URL is mandatory.

## GET – List files

The method "GET" is used to receive a list of the files stored in the directory including the information about the file creation date and the files size in bytes. A GET request to the URL of the MessageGate directory will return a file list structured in csv-c-format (“,” comma separator) including the file names of all files, their creation date (yyyy-mm-dd hh:mm:ss) and the file size in bytes.

**Example for GET using curl:**

```
curl https://webdav.viat.de/mgate/v1/~0000049640/ --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

**returns the following csv structure**

```
"Filename","Datetime","Filesize"
"m_1610.in_err0007","2016-10-31 15:02:22",6756
"s_161024.out","2016-10-31 15:16:17",7138
"m_d45f427672e345.out", 2016-10-31 15:16:59",204217
```

To receive a file list with a csv-s structure (“;” semicolon separator) the HTTP header information “Accept: text/csv-s” must be add to the request. To receive a file list with a plain text file structure without header line and different format of the creation date (dd mmm yyyy hh:mm:ss, while mmm is the shortcut of the English month) the header information “Accept: text/plain” has to be added.

The list does not show subdirectories. The amount of provided entries may be reduced while adding a selection criterion (e.g., “\*.out” or “\*.in”, if not already processed by MessageGate) to URL.

**Here is an example using curl:**

```
curl https://webdav.viat.de/mgate/v1/~0000049640/*.in --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

Wildcards (“\*”) might be also used to reduce the size of the file list, for example only show files of a certain type (“m\_\*”).

## GET - Download a file

The method "GET" is also used to download a single file. The following curl example using Curl to request the download of a message file, the default content type is message/rfc822:

```
curl https://webdav.viat.de/mgate/v1/~0000049640/m_53786626568.out --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

The file name extension ".out" is optional and will be added automatically if missing.

The HTTP content type of the response depends on the value of the HTTP request “Accept” header field and on type (file name) of file that is to be downloaded.

If the file name does not start with “M\_”, “T\_” or “S\_” or the extension is not “.out” or not defined, the content type will be “application/octet-stream” or if necessary “text/plain”.

If the file name starts with “M\_”, the content type will be “message/rfc822” or if necessary “application/octet-stream”.

If the file name starts with “T\_”, the content type will be “text/plain”.

If the file name starts with “S\_”, the content type will be “text/csv-c”, “text/csv-s” or “text/plain” depending on the format of the existing file in the directory and the HTTP request header field “Accept”.

The following HTTP status codes can be returned:

200 OK (request was successfully processed; file will be downloaded)

404 Not found (file does not exist)

406 Not Acceptable (file type incompatible to value in HTTP request “Accept” header field)

423 Locked (requested file is locked by another process)

### **DELETE – Delete a file**

The method “DELETE” will be used to delete a file in the directory.

Here an example using curl:

```
curl https://webdav.viat.de/mgate/v1/~0000049640//s_161031.out -X DELETE --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

When deleting a file, the file name extension “.out” is optional and will be added automatically if missing. If the request is successful, the status “HTTP 204 No content” will be returned. This status will also be returned if no file with this file name is present in the directory.

If the file is locked by another process, the status “HTTP 423 Locked” with the extended information “File currently locked” will be returned.

The use of wildcards in the file name is prohibited and the status “HTTP 500 Internal Server Error” will be returned.

## **5.3.2 The Web Service profile v2 (2a)**

In this profile (v2 will provide the downloaded data including CR/LF, spaces and tabs for improved readability, v2a without any additional characters) the default MessageGate file structures (Text, Message RFC2822, CSV...) will be mapped into JSON structures. When uploading data independent of called profile (v2 or v2a) additional characters might be used or not. The JSON field names in the files and the values will be mostly equivalent to those defined in the chapters 2.3 (The Message File), 2.5 (Transmission Set File Format), 2.7.2 (Request a Status Report) and 2.8 (Send Receipt Notification).

The hyphens (minus) “-” used in several RFC2822 parameter names (eg. Message-ID) may cause problems in Java script so when downloading a message file, it will be replaced by underscore “\_” and that is why the full stop in X.400 (X400\_Address) will also be removed. A field name will start (also behind an underscore “\_”) with a capital letter and the remainder will be in small letters. You will find other exceptions when checking the examples in this chapter.

In a message there can only be one originator (From:) but MessageGate allows you to add several recipients to a message, so the profile will reflect this option. There will be a separate array for To:, Cc: and Bcc: (the last two only if used) recipients including the address entries (consists of X400\_Address, User\_Id). An array structure will also be used although there is only one recipient address in this array.



The binary attachment of a message should be encoded in BASE64. Please configure this format in the properties of your MessageGate account in WebConfig (deliver binary data BASE64 encoded to file interface), also. When receiving signed messages request the sender to also encode the binary attachment(s) in BASE64.

When requesting a Status report use format "CSV-C" or "CSV-S" so the mapping into JSON structure will work correctly.

The profile offers the following functions:

### PUT - Upload a file

The method "PUT" is used to upload a new file to the MessageGate directory. The file content is included in the content of the PUT request and the (unambiguous) file name must start with "M", "T", "R" or "S" followed by an underscore "\_" character. The file extension must be ".in" or if there is no file name extension, the ".in" will be added automatically. The file name is case insensitive. The status "HTTP 404 Not found" with the extended information "Invalid filename format" will be returned if these requirements are not fulfilled.

If the HTTP header information "Content-Type" is used within the PUT request for all the files it must have the value "application/json". The status "HTTP 406 Not Acceptable" with extended information "Illegal combination of filename and Content-Type" will be returned if this requirement is not fulfilled.

If a file with a defined file name does not exist, it will be created, and the status "HTTP 201 Created" will be returned. The HTTP header information "Location" will show the URL of the file. If a file(name) already exists, the upload will be rejected with the status "HTTP 423 Locked" and the extended information "File currently locked" will be returned. Only "complete" files can be uploaded, so the HTTP header information "Content-Range" cannot be used and will be rejected with the status "HTTP 400 Bad Request" and the extended information "Content-Range not allowed" will be returned.

**Example of a message file (M\_) (with two body parts, example with one see GET) upload using Curl:**

```
Curl https://webdav.viat.de/mgate/v2/~0000049640/ -X PUT -T M_161031_001.in -H "Content-Type:application/json" --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

**with JSON structure (including CR/LF, spaces and tabs for an improved readability as defined in profile v2):**

```
{
  "TO": [
    {
      "X400_Address": "G=ipm;S=tester;O=testag;A=viaT;C=de",
      "User_Id": "49637"
    }
  ],
  "From": {
    "X400_Address": "G=MG1;S=MGATE;CN=MG1 MGATE;O=TESTAG;P=MGATE;A=VIAT;C=DE",
    "User_ID": "49603"
  },
  "Message_Id": "0001-01-03-2018",
  "Date": "01 Mar 2018 09:11:30 +0100",
  "Subject": "testout",
  "Mime_Version": "1.0",
  "Content_Type": {
```

```

    "Mime_Type": "multipart/mixed"
  },
  "Multipart": [
    {
      "Content_Type": {
        "Mime_Type": "text/plain"
      },
      "Content_Transfer-Encoding": "8bit",
      "Data": "Test with BASE64 encoded file attachment"
    },
    {
      "Content_Type": {
        "Mime_Type": "application/octet-stream"
      },
      "Content_Transfer-Encoding": "BASE64",
      "Content_Disposition": {
        "Disposition_Type": "attachment",
        "Filename": "webdav.p12"
      },
      "Data": "AQcBMwGCiqSlb3DQEMAYwDgQlZn9gMIQvocsCAggAgIIILQGaNZr0IW6bN6jEdpt
hjnBzmCv8W9ipvE8wmpVxzUEwj5Mh226vHaBp2WtMBaHPSomsXFMPEJJj9JFnF2SgPxDZVjUe5ImUB2EQDA
opQEYLxJSX0YXh8uqnSD5Se4vuex+kunnb6o2nGXT8+Y9m3/uNCD9MEb6CGIA0JExtmWQJXkDeHDZjLjYiVC
pclt-
NeMNC7EGH842jRGzS1umfOeSWb8+TcA2/uZtzaE9uIL7ILfD7dfIJz/4uawC+LstCfO984pFKR8vOxKIAdbOn1Cpu
SQFHHdgCZYVy1EHODllmQbml+bJ2GwxUPKDdUGdyK6G45JHZjuj4zDUSHZhUpRQwAPYyQo6zxhdd7NsdXP
u7mDisNE/p6p0DNPTf97j/AiPWVMEwz0nsfITqF+4L0NXVKia7Mp8o7Zzn5XpwJ0/LP+47/+ZyCaClqB/qYtGlbXgl
04DFbS6xaoUu7iNh7ZSqnXNMRJREtBx/WVoMChpYHuvVqitPWdsBpawNpUHs5uEXUopa0UlyXOn9ALfLE0t9v
5FP4NE3xSHMGAc5iisH7Fys8g5Z+SGp3n9ynM8Jw97JhZfjKoQMqrFzNL5FIZUBVwNYOtUNXxKJ3L+1WtRXSE
QgmfhptKZicCZKHoGZQ4Z8F4r9sA7wmS9CbLjiNQlmWlrvaMWE3fi6dzhrUOFIdu2LE7TI7+1Qmh/AcP3NVIUSU
ZIGJqqGc511BUmpMP3CJPo25xJ7zAek/YECJmQ5p9"
    }
  ]
}

```

#### Example of a Transmission Set file (T\_) upload using curl:

```
curl https://webdav.viat.de/mgate/v2/~0000049640/ -X PUT -T T_181116_0001.in -H "Content-Type:application/json" --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

providing a JSON array structure with two EDIFACT interchanges (the segment terminator Apostrophe „“ has been „escaped“, also the Web Service is able to handle this character, and CR/LF, spaces and tabs for an improved readability as defined in profile v2 have been added). Independent of the number of interchanges always use a JSON array structure.

```

[
  {
    "Interchange": "UNA:+.? \u0027
UNB+UNOA:2+MGATE1:65+TESTER:65+020508:1413+0709210008\u0027
UNH+EVA0000001+INVOIC:D:95A:UN:ETEIB++0\u0027
...
UNT+37+EVA0000001\u0027
UNZ+1+0709210008\u0027"
  },
  {
    "Interchange": "UNA:+.? \u0027

```

```

    UNB+UNOA:2+MGATE1:65+TESTER:65+020508:1413+0709210009\u0027
    UNH+EVA0000001+INVOIC:D:95A:UN:ETEIB++0\u0027
    ...
    UNT+37+EVA0000001\u0027
    UNZ+1+0709210009\u0027"
  }
]

```

#### Example of a Status Report Request (S\_) upload using curl:

```
curl https://webdav.viat.de/mgate/v2/~0000049640/ -X PUT -T S_12002.in -H "Content-Type:application/json" --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

**providing a JSON structure (CR/LF, spaces and tabs for an improved readability as defined in profile v2 have been added):**

```

{
  "Since": "13-Nov-2018",
  "Format": "CSV-C",
  "Direction": "both"
}

```

#### Example of a Receipt Notification (R\_) upload using curl:

```
curl https://webdav.viat.de/mgate/v2/~0000049640/ -X PUT -T R_12002.in -H "Content-Type:application/json" --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

**providing a JSON structure (CR/LF, spaces and tabs for an improved readability as defined in profile v2 have been added):**

```

{
  "Status": "Processed"
}

```

The "/" at the end of the URL is mandatory.

## GET – List files

The method "GET" is used to receive a list of the files stored in the directory including the information about the file creation date and the files size in bytes. A GET request to the URL of the MessageGate directory will return a list of entries within a JSON array ("," comma separator) including the file names of all files, their creation date (yyyy-mm-dd hh:mm:ss) and the file size in bytes.

#### Example for GET using curl:

```
curl https://webdav.viat.de/mgate/v2/~0000049640/ --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

**returns the following JSON array structure (for an improved readability, the profile v2 with CR/LF, spaces and tabs will be used)**

```

[
  {
    "Filename": "m_1610.in_err0007",
    "Datetime": "2016-10-31 15:02:22",
    "Filesize": "6756"
  },
  {
    "Filename": "s_161024.out",
    "Datetime": "2016-10-31 15:16:17",
    "Filesize": "7138"
  },
  {
    "Filename": "m_d45f427672e345.out",

```

```

        "Datetime": "2016-10-31 15:16:59",
        "Filesize": "204217"
    }
]

```

The list does not show subdirectories. The amount of provided entries may be reduced while adding a selection criterion (e.g., "\*.out" or "\*.in", if not already processed by MessageGate) to URL.

**Here is an example using curl:**

```
curl https://webdav.viat.de/mgate/v1/~0000049640/*.in --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

Wildcards ("\*") might be also used to reduce the size of the file list, for example only show files of a certain type ("M\_\*").

### GET - Download a file

The method "GET" is also used to download a single file while adding the file name to the URL. While downloading EDIFACT interchanges within message text or Transmission Set file the segment terminator Apostrophe "'" will be replaced using "\u0027" because some parsers have problems to process this character properly.

**Here is an example using curl, where an unsecured message file including one body part will be downloaded and the Content Type application/json will be used:**

```
curl https://webdav.viat.de/mgate/v2/~0000049640/m_53786626568.out --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

**returns the following JSON structure (for an improved readability, the profile v2 with CR/LF, spaces and tabs will be used, please be aware that in difference to profile v1 instead of "X-Mpduid" the field name "Mts\_Id" will be used equivalent to Status report):**

```

{
  "To": [
    {
      "X400_Address": "G=MG1;S=MGATE;O=TESTAG;P=MGATE;A=VIAT;C=DE",
      "User_ID": "49603"
    }
  ],
  "From": {
    "X400_Address": "G=ipm;S=tester;O=testag;A=viaT;C=de",
    "User_Id": "49637"
  },
  "Message_Id": "124 01-03-2018",
  "Mts-Id": "16067AF811E8E7211E0018B9",
  "Date": "01 Mar 2018 12:10:59 +0100",
  "Subject": "test001",
  "Mime_Version": "1.0",
  "Content_Type": {
    "Mime_Type": "text/plain"
  },
  "Content_Transfer-Encoding": "8bit",
  "Data": "test without binary attachment"
}

```

**Here is an example using curl, where an encrypted and signed message file including one body part will be downloaded and the Content Type application/json will be used:**

```
curl https://webdav.viat.de/mgate/v2/~0000049640/m_d7ae2u7s1oh0t3ch.out --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

returns the following JSON structure (for an improved readability, the profile v2 with CR/LF, spaces and tabs will be used, please be aware that in difference to profile v1 instead of “X-Mpduid” the field name “Mts\_Id” will be used equivalent to Status report):

```
{
  "To": [
    {
      "X400_Address": "G=MG1;S=MGATE;O=TESTAG;P=MGATE;A=VIAT;C=DE",
      "User_Id": "49603"
    }
  ],
  "From": {
    "X400_Address": "G=ipm;S=tester;O=testag;A=viaT;C=de",
    "User_Id": "49637"
  },
  "Message_Id": "8552 19/03/12",
  "Mts-Id": "0172375E11E944A185D454AE",
  "Date": "12 Mar 2019 08:29:51 +0100",
  "Subject": "test Web Service with encrypted message",
  "Mime_Version": "1.0",
  "Content_Type": {
    "Mime_Type": "application/pkcs7-mime ",
    "Smime_Type": "enveloped-data",
    "Name": "smime.p7m"
  },
  "Content_Transfer-Encoding": "base64",
  "Content_Disposition": {
    "Disposition_Type": "attachment",
    "Filename": "smime.p7m"
  },
  "Data": "MIAGCSqGSIlb3DQEHA6CAMIACAQAxggKSMI...AAAAAANCg"
}
```

Here is an example using curl, where a signed message file including two text body parts and a binary body part will be downloaded and the Content Type application/json will be used:

```
curl https://webdav.viat.de/mgate/v2/~0000049640/m_d7ae2u7s1oh0t3ch.out --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

returns the following JSON structure (for an improved readability, the profile v2 with CR/LF, spaces and tabs will be used, please be aware that in difference to profile v1 instead of “X-Mpduid” the field name “Mts\_Id” will be used equivalent to Status report):

```
{
  "To": [
    {
      "X400_Address": "G=MG1;S=MGATE;O=TESTAG;P=MGATE;A=VIAT;C=DE",
      "User_Id": "49603"
    }
  ],
  "From": {
    "X400_Address": "G=ipm;S=tester;O=testag;A=viaT;C=de",
    "User_Id": "49637"
  },
  "Message_Id": "8553 19/03/25",
  "Mts_Id": "F38BA8C211E94EF185D4DABC",
}
```

```

    "Date": "25 Mar 2019 11:34:29 +0100",
    "Subject": "test Web Service mit signierter Mitteilung",
    "Disposition_Notification_To": "\"G=ipm;S=tester;O=testag;A=viaT;C=de\"",
    "Mime_Version": "1.0",
    "Content_Type": {
        "Mime_Type": "multipart/signed",
        "Protocol": "application/pkcs7-signature",
        "Micalg": "sha256"
    },
    "Multipart": [
        {
            "Content_Type": {
                "Mime_Type": "multipart/mixed"
            },
            "Multipart": [
                {
                    "Content_Type": {
                        "Mime_Type": "text/plain",
                        "Charset": "ISO-8859-1"
                    },
                    "Content_Transfer-Encoding": "quoted-printable",
                    "Data": "test"
                },
                {
                    "Content_Type": {
                        "Mime_Type": "text/plain",
                        "Charset": "ISO-8859-1"
                    },
                    "Content_Transfer-Encoding": "quoted-printable",
                    "Data": "\\r\\nC:\\\\Users\\tester\\Documents>C:\\OpenS....."
                }
            ],
            "Content_Type": {
                "Mime_Type": "application/octet-stream"
            },
            "Content_Transfer-Encoding": "base64",
            "Content_Disposition": {
                "Disposition_Type": "attachment",
                "Filename": "Modem_cfos.txt",
                "Modification_Date": "Wed, 02 Sep 2015 09:04:32 +0100"
            },
            "Data": "DQoJICAgICAgeysrfSBkZW5vdGV....."
        },
        {
            "Content_Type": {
                "Mime_Type": "application/pkcs7-signature",
                "Name": "smime.p7s"
            },
            "Content_Transfer-Encoding": "base64",
            "Content_Disposition": {

```

```

        "Disposition_Type": "attachment",
        "Filename": "smime.p7s"
    },
    "Data": "MIIJ5gYJKoZIhvcNAQcCollJ1zCCCdMCAQExDTALBgIlg.....CX"
}
]
}

```

Here is an example using curl, where a Transmission Set file including one EDIFACT interchange will be downloaded and the Content Type application/json will be used:

```
curl https://webdav.viat.de/mgate/v2/~0000049640/T_657832112362.out --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

returns the following JSON array structure (the segment terminator Apostrophe “'” has been “escaped” and for an improved readability, the profile v2 with CR/LF, spaces and tabs will be used). Independent of the number of interchanges a JSON array structure will be used:

```

[
  {
    "Interchange": "UNA:+.? \u0027
    UNB+UNOA:2+TESTER:65+ MGATE1:65+020508:1413+0709210009\u0027
    UNH+EVA0000001+INVOIC:D:95A:UN:ETEIB++0\u0027
    ...
    UNT+37+EVA0000001\u0027
    UNZ+1+0709210009\u0027"
  }
]

```

Here is an example using curl, where a Status report including the entry of a sent and received message will be downloaded and the Content Type application/json will be used:

```
curl https://webdav.viat.de/mgate/v2/~0000049640/R_10023.out --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

returns a JSON structure (for an improved readability, the profile v2 with CR/LF, spaces and tabs will be used and in contrary to profile v1 and equivalent to message structure for the addresses the field names “X400\_Address” and “User\_Id” will be used. If the User-ID is not available, a "" will be provided.):

```

{
  "Type": "Status Report",
  "User_Id": "49603",
  "Date": "2019/02/15 08:33:29",
  "Timezone": "GMT",
  "Filter": {
    "Disposition": "All",
    "Direction": "Both",
    "Format": "CSV-C",
    "Since": "15-Feb-2019"
    "Message-Id": "",
    "Order-Id": ""
  },
  "Report": [
    {
      "From": {
        "X400_Address": "G=ipm;S=tester;O=testag;A=viat;C=de",
        "User-ID": "49637"
      },
      "To": {}
      "Order_Id": "657832112376",
    }
  ]
}

```

```

    "Message_Id": "126 13-11-2018",
    "Mts_Id": "16067AF777E8E7211E0018B9",
    "Received": "13.11.2018 12:34",
    "Sent": "",
    "Delivered": "",
    "Read": "",
    "Reason": "",
    "Diagnostic": "",
    "Errordate": "",
    "Rcpt_Type": "To"
  },
  {
    "From": {}
    "To": {
      "X400_Address": "G=ipm;S=tester;O=testag;A=viat;C=de",
      "User_Id": "49637"
    },
    "Order_Id": "131118-0001",
    "Message_Id": "131118-0001",
    "Mts_Id": "19127AF777E8E7211E002312",
    "Received": "",
    "Sent": "13.11.2018 12:36",
    "Delivered": "13.11.2018 12:36",
    "Read": "13.11.2018 12:38",
    "Reason": "",
    "Diagnostic": "",
    "Errordate": "",
    "Rcpt_Type": "To"
  }
]
}

```

The file name extension “.out” is optional and will be added automatically if missing. The HTTP content type of the response is “application/json”.

The following HTTP status codes can be returned:

200 OK (request was successfully processed; file will be downloaded)

404 Not found (file does not exist)

406 Not Acceptable (file type incompatible to value in HTTP request "Accept" header field)

423 Locked (requested file is locked by another process)

## DELETE – Delete a file

The method "DELETE" will be used to delete a file in the directory.

Here an example using curl:

```
curl https://webdav.viat.de/mgate/v2/~0000049640/s_161031.out -X DELETE --cacert ca-bundle.crt --cert
tester_cert.pem --key tester_key.pem -v
```

When deleting a file, the file name extension “.out” is optional and will be added automatically if missing. If the request is successful, the status “HTTP 204 No content” will be returned. This status will also be returned if no file with this file name will be present in the directory.



If the file is locked by another process, the status “HTTP 423 Locked” with the extended information “File currently locked” will be returned.

The use of wildcards in the file name is prohibited and the status “HTTP 500 Internal Server Error” will be returned.

### 5.3.3 The Web Service profile v3 (3a)

This profile is very similar to profile v2/v2a (v3 will provide the downloaded data including CR/LF, spaces and tabs for improved readability, v3a without any additional characters). It is designed to be used in conjunction with Closed User Group (CUG) and the TEDIS P2 approach used for EDI documents (one recipient and only one EDIFACT document sent in one text or binary body part). So, all functions/commands described in Web Service profile v2 (v2a) will be also available in v3 and the results are mostly equal.

Different is the (minimalized) structure (only the Recipient's User-ID, Message ID, Subject, only one Text or Binary body part) to store a message file in the MessageGate directory using a PUT command for the upload. Also, when downloading a message using GET there will be that minimalized structure but here there will be only the Sender's User-ID and in addition also the Order-ID included. That is necessary because in profile v3 the option NEXTFILE will be available. When using this option, it is no longer necessary to list the directory using GET command to receive the file-name of a message and after that use the GET command in conjunction with file-name to download it explicitly. A GET with option NEXTFILE will provide in the result the structure of the oldest (delivered) message in the MessageGate directory. At the end of the download the Web Service will move the message file into the subdirectory “ARCHIVE” so that NEXTFILE would not provide it any longer. In the subdirectory “ARCHIVE” the message file will be available for an explicitly download (use GET to list directory and in conjunction with file name to download message) until the purger process (configured maximum age of file in directory) would delete it. In conjunction with CUG it is possible to use the option NEXTFILE together with Sender's User-ID (add User-ID in URL and receive the oldest message sent by this sender) and the archived message files will be moved into a sender specific subdirectory of “ARCHIVE”.

#### PUT - Upload a message file

The method "PUT" is used to upload a new message to the MessageGate directory. The file content is included in the content of the PUT request and the (unambiguous) file name must start with “M” followed by an underscore “\_” character. The file extension must be “.in” or if there is no file name extension, the “.in” will be added automatically. The file name is case insensitive. The status “HTTP 404 Not found” with the extended information “Invalid filename format” will be returned if these requirements are not fulfilled.

If the HTTP header information “Content-Type” is used within the PUT requests for all the files it must have the value “application/json”. The status “HTTP 406 Not Acceptable” with extended information “Illegal combination of filename and Content-Type” will be returned if this requirement is not fulfilled.

If a file with a defined file name does not exist, it will be created, and the status “HTTP 201 Created” will be returned. The HTTP header information “Location” will show the URL of the file. If a file(name) already exists, the upload will be rejected with the status “HTTP 423 Locked” and the extended information “File currently locked” will be returned. Only “complete” files can be uploaded, so the HTTP header information “Content-Range” cannot be used and will be rejected with the status

“HTTP 400 Bad Request” and the extended information “Content-Range not allowed” will be returned.

In difference to the profile v2/v2a only a minimalized header (Recipient’s User-ID, Message-ID, Subject) will be used and only one body part (a Text body encode in ISO-Latin 1 or a binary body → BP14 without filename) will be supported.

**Example of a message file (M\_) upload with Text Body Part (example with Binary Body Part see GET) using curl:**

```
curl https://webdav.viat.de/mgate/v3/~0000049640/ -X PUT -T M_211020_001.in -H "Content-Type:application/json" --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

**with JSON structure (including CR/LF, spaces and tabs for an improved readability as defined in profile v3):**

```
{
  "User_Id": "49637",
  "Message_Id": "0001-01-11-2021",
  "Subject": "testin",
  "TEXT": "Test mit BASE64 encoded file attachment"
}
```

## GET - Download a file

The method "GET" is also used to download a single message file while adding the file name to the URL.

**Here is an example using curl, where an unsecured message file including one binary body part (text of example PUT encoded in BAS64) will be downloaded and the Content Type application/json will be used:**

```
curl https://webdav.viat.de/mgate/v3/~0000049640/m_53786626568.out --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

**returns the following JSON structure (for an improved readability, the profile v3 with CR/LF, spaces and tabs will be used):**

```
{
  "Order_Id": "53786626568",
  "User_Id": "49637",
  "Message_Id": "0002-01-11-2021",
  "Subject": "testout",
  "DATA": " VGVzdCBtaXQgQkFTRTY0IGVuY29kZWQgZmlsZSBhdHRhY2htZW50"
}
```

When using the Option NEXTFILE instead of file name the Web Service will provide the oldest message in MessageGate directory using a (minimalized) JSON structure within http result. Here is an example:

```
curl https://webdav.viat.de/mgate/v3/~0000049640/NEXTFILE --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

At the end of the download the message file will be moved into the subdirectory “ARCHIVE” and it is possible to download it once more. Here’s an example:

```
curl https://webdav.viat.de/mgate/v3/~0000049640/archive/m_53786626569.out --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

The subdirectory will be created while moving a message file the first time.

If the Closed User Group of a MessageGate account is enabled, the User-ID of the sender will be added to the name of the message file and now it is possible to receive in conjunction with NEXTFILE the oldest message sent by this User-ID. Here is an example:

```
curl https://webdav.viat.de/mgate/v3/~0000049640/NEXTFILE/042788/ --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

At the end of the download the message file will be moved not only into the subdirectory "ARCHIVE" but there into the subdirectory defined by sender's User-ID and it is possible to download it once more. Here's an example:

```
curl https://webdav.viat.de/mgate/v3/~0000049640/archive/042788/m_53786626579_042788.out --cacert ca-bundle.crt --cert tester_cert.pem --key tester_key.pem -v
```

The subdirectory defined by sender's User-ID will be created while moving a message file the first time.

If the v3/v3a profile is not used in conjunction with Closed User Group, there might be the possibility that a partner had sent a message including several body parts. In such a case the Multi Part of the Message file would be mapped unchanged into JSON file. Here's an example:

```
{
  "Order_Id": "051B3MQV3UH0W39N",
  "User_Id": "49637",
  "Message_Id": "238 22/03/09",
  "Subject": "test 2xFTAM",
  "Multipart": "--MG=_D8563E2211EC9FB6DC9C1895_=MG\r\nContent-Type: application/octet-stream\r\nContent-Disposition: attachment; filename=\"test.zip\"\r\nContent-Transfer-Encoding: base64\r\n\r\nUESDBBQAAAA ... IFydHMNCiAgKD4xTUlpIGZhaWxpbmcuDQo=\r\n--MG=_D8563E2211EC9FB6DC9C1895_=MG--\r\n"
}
```

The upload of a message including a Multi Part is not supported.

## 5.4 REST based modules used for test purposes

For the Web Service access tests, the following utilities were used:

- Curl (see <https://curl.haxx.se/download.html>), a Command line Interface and available for different operating systems (Windows 32 und 64 Bit, Linux, Unix, Mac OS X, OpenVMS...).
- Postman, in the past an add-on for the Google Chrome Web browser for Windows, now a stand-alone App available for Windows, MacOS and Linux (see <https://www.postman.com/downloads/>).

## 6 AS2 and MessageGate

### 6.1 General Information

The AS2 Gateway users are not aware that they are using a MessageGate process because the AS2 communication is serviced by a module called ComAS2 (Java communication module offered by Compinia GmbH & Co. KG for different Operating systems, e.g., Windows and Linux). The MessageGate process (MGAS2X) accepts AS2 documents received via ComAS2, creates X.400 messages, and send these to the respective partners. If the “End-to-End” security is not enabled (default) in the trading relation, the ComAS2 module will extract the user data out of the secured content of the AS2 message so that the X.400 partner is able to process it without any restrictions. If this option is enabled ComAS2 will forward the S/MIME content within a single X.400 Body Part 15/FTAM body part (see also Chapter 2.3.4 S/MIME secured content) to the X.400 partner. In such a case, the partner’s application must extract the user data from the secured message content. To decrypt and check the signature of received messages the AS2 user must store in his solution the certificate provided by the X.400 partner and not the default certificate of *BusinessMail X.400* AS2 service. Hence, it is necessary to verify whether the X.400 partner’s application can handle secured content before enabling this option. For version 5.2 and newer versions of the P7 Clients FileWork and UA-FI, it is necessary to add the certificate of the signer to the signature of the secured content.

X.400 messages destined for an AS2 user are forwarded directly by MessageGate to ComAS2. ComAS2 in turn transfers these documents using EDIINT AS2 protocol to the configured URL of the user’s application and adds, depending on the value of the parameter End-to-End security and other partner relation specific parameters, a signature and encrypts the AS2 message. MessageGate also maps X.400 reports to AS2 MDN and vice versa. Depending on the configured gateway mode, the AS2 MDN will be mapped into an X.400 Receipt Notification or an X.400 Delivery Notification.

#### Gateway-Modus “Agent” or “Transfer”

If the gateway mode is “Agent” (Default) MessageGate will send a requested X.400 Delivery Notification (DN) when message was provided to the AS2 communication module (ComAS2) for the delivery to the customer’s AS2 solution. An MDN sent for this delivery would be mapped into a Receipt Notification (Read Notification → Status in Report is “Read”) if requested by the X.400 partner and enabled by the AS2 user. If the gateway mode is “Agent”, the AS2 user has the access to several useful features in *WebConfig*, e.g., Message Management and automatically generated Status report.

The gateway mode “Transfer” should be used if the AS2 user is not a customer of *BusinessMail X.400*, and such does not have the ability to use *WebConfig* and the extended features of the AS2 gateway. In this case, the X.400 user must order and pay for the AS2 connection. For this X.400 user, it is important that the MDN sent by the AS2 partner will be mapped into a Delivery Notification (Status in Report is “Delivered”). A requested RN will be ignored, and the Message Management functionality is restricted. While using gateway mode “Transfer” the X.400 Standards will define the message expiry time-out values and an NDN with an appropriate error code will be sent for expired messages.

Both ComAS2 and MessageGate support the MIME Content-Type “Multipart mixed”. When using the *BusinessMail X.400* AS2 Gateway an AS2 user’s X.400 partners always appear as an AS2 partner and vice versa, an X.400 user’s AS2 partners appear as X.400 partners without any restrictions regarding the respective transport protocol used.

EDIINT AS2 (RFC4130) is a Peer-to-Peer protocol so that the AS2 user must assign a separate AS2 ID for each of his partners. If he wants to communicate with X.400 partners via the AS2 Gateway, he also must assign AS2 IDs for these partners. This AS2 ID will be mapped in the AS2 Gateway based on the host trading profiles to the X.400 address of a partner and this address will be used when sending X.400 messages.

An AS2 user can also use the *BusinessMail X.400* central EDI functionality to send EDIFACT documents to X.400 partners or to receive EDIFACT documents when configuring a trading relation and assigning an AS2 ID for the central EDI function.


BusinessMail X.400 :: [WebConfig](#)
as2tester (49639)  
AS2-ID: AS2Tester001


## AS2-X.400 Relation :: Create new Relation

☒ Use Central EDI Functionality

### AS2 Properties

IDs

AS2 ID: AS2 User AS2Tester001  
AS2 ID: X.400 Partner   
☒ Add User ID of X.400 partner to given AS2 ID  
Enable End-to-End Security  ☐

Properties: X.400 → AS2 

Default URL   
Altern. URL   
Compress ☒  
Sign   
Encrypt   
Request MDN ☒  
Sign MDN ☒  
MDN Transfer

In such a case, the AS2 user will only need one AS2 ID to communicate with several X.400 partners because the central EDI function will use the header information in the EDIFACT documents to find the partner’s X.400 address. This option can help the AS2 user to reduce the costs if the AS2 application has a per partner license model. The End-to-End security is not available in conjunction with central EDI functionality, because here the TEDIS P2 definition will be used where only one recipient and only one document (EDIFACT) sent as a text or binary body part (BP14) is allowed.

**Please be aware that there is a feature to note in the central EDI functionality of the AS2 Gateway causing a difference to the MessageGate or EDIBOX (special mailbox type) behavior. It is possible to send a Transmission Set file including several EDIFACT interchanges, but in this case, the AS2 Gateway is not able to send an MDN based on DN or RN. Each interchange will result in a separate X.400 message so that the mapping of reports is not possible. The AS2-Gateway will send the MDN after the EDI function has processed all interchanges (sent the messages via MTA or refuse them in case of missing relations or wrong parameters). So, use a Transmission Set file that includes only one EDIFACT interchange if you need the correct mapping of reports.**

The MessageGate process also adds the X.400 addresses in the TO: and FROM: elements of the AS2 RFC2822 Header. These elements are not necessary in the AS2 communication where AS2-TO: and AS2-FROM: are used to define recipient and sender. However, in conjunction with the central EDI functionality it is these elements that provide the AS2 users with information about the sender of the EDIFACT document.

## 6.2 Differences between File Interface and AS2 users

The following pages describe an AS2 account configured to run in the MTA mode “Agent” where the AS2 user has access to *WebConfig*. In the MTA mode “Transfer” the menu items EDI relations, Message Management and automatically generated Status report will not be available, or the access will be restricted. By default, in the MTA mode “Transfer” there will be only one relation between an X.400 user who is a *BusinessMail X.400* customer and his partner, using an application that supports AS2 only, and who is not a *BusinessMail X.400* customer.

For the mapping between MDN and X.400 Reports and vice versa, the MessageGate process uses the same database relation (Trace\_Tab) to store the message status information. An AS2 user can also request a Status Report in the *WebConfig* menu (View/Download) or configure the delivery of Status Reports via the AS2 protocol sent with a preconfigured AS2 ID (for more details about the format of the entries see the end of this Chapter).

The MessageGate process will not delete the database entries once the transaction (data transferred, and report sent) is completed. The entries will remain in the database until the end of the entry lifetime (determined by the purge time). The “Purger” process runs several times a day and deletes the entries that have reached the end of their lifetime (default is 240 hours = 10 days, this can be configured individually on customer request).

If the “Purger” process deletes the entry of a transaction that has not yet completed (for example a requested RN has not yet been received and ComAS2 has not been able to send an asynchronous MDN) the final status will be set to “failed”.

### Properties of AS2 communication

There are some other additional options configured in “Default Properties”.


The mapping between X.400 and AS2 messages (data compression, signing with SHA1, SHA256, SHA384 or SHA512, encrypt message with 3DES or AES256-CBC, request MDN) and between the reports is configured in the *WebConfig* menu item “AS2 – X.400 Relation: Default Properties” or in the individual Trading Relations.








Additional menu items are the AS2 ID and up to three certificate aliases (Signature, Encryption, TLS → including each up to two certificates managed in the menu X.509 certificates or via the Certificate Exchange Management CEM using a special kind of AS2 message).


BusinessMail X.400 :: [WebConfig](#)
as2tester (49639)  
AS2-ID: AS2Tester001

## AS2-X.400 Relation :: Default Properties

**User**  
X.400 Address cn=as2 tester; g=as2; s=tester; o=testag; n-id=2049639; a=VIAT-AS2; c=DE

**Properties of AS2 User **

MTA Mode	Agent 
AS2 ID	AS2Tester001
URL	<a href="http://as2.testag.de:4080/">http://as2.testag.de:4080/</a>
Certificate Alias	AS2TESTPC1
Alias Verschlüsselung	AS2TESTPC1_ENC
Alias TLS	AS2TESTPC1_TLS
Email Address	as2@testag.de
Inactive	<input checked="" type="checkbox"/>
Enable AS2 Bypass	<input checked="" type="checkbox"/>
Duplicate Check	<input checked="" type="checkbox"/>
Enable End-to-End Security 	<input type="checkbox"/>
Purge time 	240 Hours
Send Timeout Message 	1440 Minutes (0-65535, 0=Unlimited Retries)
Send Timeout MDN 	1440 Minutes (0-65535, 0=Unlimited Retries)
Receive Timeout MDN 	0 Minutes (0-65535, 0=no Retries)
Max. number of retries 	0 (0-127, 0=no Retry)

**Default Properties: X.400 ⇒ AS2 **

## Send Timeout of an AS2 message

Apart from the configuration of the AS2 ID and the URL of his AS2 application the customer can define how the AS2 Gateway should react if problems occur when delivering messages or reports to the customer's application. One can configure how long the AS2 Gateway should try to deliver a message before moving the data of this transaction into the Message Management for manual intervention. A user can select an entry in Message Management to download the message user data, to delete it or to reactive the transaction.

## Message Management

**AS2-X.400 Relation :: Message Management** ⓘ

Records since  (Format: DD-MMM-YYYY hh:mm:ss)

Records till  (Format: DD-MMM-YYYY hh:mm:ss)

Filter:  ⓘ

Time	Message	Status	Download	Action items
13.07.2022 10:54:10	04224KMTRRI0W7DI	bypassed		resend delete
13.07.2022 10:54:22	14224KMTRRI0W7DI	bypassed		resend delete
13.07.2022 13:11:05	24224KMTRRI0W7DL	bypassed		Send RN resend delete

From: "G=ipm;S=tester;O=testag;A=viat-test;C=de" 49637@viaT.de  
 Order-ID: 24224KMTRRI0W7DL  
 Message-ID: 83 22/07/13  
 MTS-ID: 405B062611ED02ADDC9C87A5  
 Content-Type: text/plain  
 Encoding: 8bit  
 Received: 13-Jul-2022 13:11:05 +0200  
 AS2-ID: BM400\_49637  
 AS2-Status: bypassed

The entries in the Message Management are available until the “Purger” process deletes them in the database relation (Trace\_Tab). If a message listed in the Message Management has been reactivated, the AS2 Gateway will try to deliver it, also the function “AS2 Bypass” is still enabled. In such a case, the message will only be listed in the Message Management if the send timer has expired. When sending a requested Receipt Notification (RN), the transaction will be finalised ie. completed, and it is not longer possible to resend the message via AS2.

## Send Timeout MDN and retransmit AS2 message

The parameter “Send Timeout MDN” defines how long the AS2 Gateway will try to deliver an asynchronous MDN to a user’s AS2 application before it will set this transaction status to “failed”.

The next two parameters define whether the AS2 Gateway should retransmit a message if the requested asynchronous MDN has not been received in time (parameter “Receive Timeout MDN”) and how often the AS2 Gateway should retransmit the message. If the parameter “Max number of retries” is set to “0” the AS2 Gateway will directly list the message in the Message Management after the Receive Timeout MDN timer has expired. If the parameter “Max number of retries” value is not “0” the AS2 Gateway will resend message as often as defined and only then will the message be listed in the Message Management.

We recommend requesting a synchronous MDN when configuring the submission of AS2 messages (X.400 ⇒ AS2, see screen shot on next page).

## Management AS2 communication in case of a problem

“Inactive” an AS2 user can temporarily interrupt the delivery of messages and reports (for example when replacing certificates). If there is a bigger problem or a longer downtime of a user’s application is expected, setting the option “Enable AS2 Bypass” means that all the messages will be listed in the Message Management for manual processing.



When the parameter “Duplicate Check” is set, the AS2 Gateway will not forward messages to the X.400 partner that a user’s AS2 application has sent twice.

## Mapping an AS2 into X.400 message

There are some AS2 specific parameters in the AS2 account properties but most parameters for the mapping between AS2 documents and X.400 messages are equivalent to those of the MessageGate file interface users.

The screenshot displays the 'Default Properties: X.400 → AS2' configuration window. It is divided into two main sections: 'Default Properties' and 'X.400 Properties'.

**Default Properties: X.400 → AS2**

- Compress: ☒
- Sign: SHA384
- Encrypt: AES
- Request MDN: ☒
- Sign MDN: SHA512
- MDN Transfer: asynchronous: HTTP

**X.400 Properties**

**Properties**

- Send requested asynchronous MDN: when Message was delivered (DN)
- Receipt Notification requested in X.400 messages should be:
  - ☐ ignored
  - ☒ send, if client had sent notification
- Message Expiration: 1440 Minutes
- X.400 Content Type: IPM84, IPM88
- Bodypart: IA5 Text, Bilateral (Bodypart 14), ISO Latin 1, Depends on context (variable)
- Encode binary data: binary, base64
- Comment: test

At the bottom, there are 'Ok' and 'Cancel' buttons.

To conform to the AS2 Standard there are four possible values for the menu item “Send requested asynchronous MDN”:

- Immediate → Send asynchronous MDN as soon as the message is delivered to the MessageGate process
- When Message has been sent → Send asynchronous MDN if the X.400 message has been sent and the MTA has generated a Message-ID. This Message-ID will be set within the MDN (X-MPDUID or MTS-ID).
- When Message has been delivered (DN) → Send asynchronous MDN if the X.400 message has been delivered to the partner’s mailbox. This also implies the request for a Non-Delivery Notification!
- When Message has been processed (RN) → Send asynchronous MDN if the message has been processed by the partner’s application (read/fetched). This implies the request for a Delivery Notification, but a DN is only visible in a status report.

When choosing the value “Immediate” the AS2 user only has the information that the documents were accepted by the AS2 Gateway, but he cannot deduce whether there were problems sending or transferring the X.400 message to the recipient’s mailbox.

Attention, this restriction also exists when requesting a synchronous MDN. So, we explicitly do **not** recommend requesting a synchronous MDN in conjunction with AS2 gateway. In case of a X.400 transfer failure only a status report will offer the AS2 user (e.g., in WebConfig or receiving a automatically generated status report using option “failed only”) extended transaction information.

Using the setting “when message was sent” an AS2 sender only knows that the respective X.400 message has been sent, as he will receive the MTA message identifier X-MPDUID, but he does not know whether the X.400 recipient has received the message or not.

We recommend using the setting “when message was delivered (DN)” (default for this parameter). In such a case, the receipt of a positive MDN by the AS2 Gateway implies that the message has arrived in the partner’s mailbox. Choosing the setting “when message was processed (RN)”, the AS2 user can receive the information that the partner has processed the message (read/fetched), but the X.400 partner needs to create and send the Receipt Notification in time. Otherwise, the AS2 application must wait too long causing alarms or unnecessary retransmissions.

## Using the central EDI functionality

When using the central EDI functionality, it is possible to send several X.400 messages while sending a Transmission Set file including the appropriate number of EDIFACT interchanges via AS2. In such a case, it is not possible to map the X.400 reports into an asynchronous MDN because each X.400 message may result in the receipt of a report. If the Option “Immediate” is not used the AS2 gateway will send an MDN as soon as the MTA has processed all messages. In the MDN there will be an entry for each EDIFACT interchange ID with the corresponding MTS ID for messages successfully sent or error information (wrong address, syntax error, no EDI relation...) if the message submission failed. If the exact mapping of an X.400 report to an MDN is required, then only one EDIFACT interchange should be included into a Transmission Set file.

Example for the text in MDN:

\*\*\* IC(s) failed \*\*\*

AS2TEST3	0815	:11	NOTEXIST	:65	Receiving Partner not found
AS2TEST4	NOTEXIST	:11	2001005	:65	Sending Partner not found

\*\*\* IC(s) submitted \*\*\*

AS2TEST1	0815	:11	2001005	:65	X-MPDUID: C8D72CFB11E17CCE85D40FBA
AS2TEST2	0815	:11	2001005	:65	X-MPDUID: C8EDE94B11E17CCE85D413BA

When an X.400 user sends messages to his AS2 partner the MessageGate process will send a requested Delivery Notification to the X.400 originator after the message has been transferred to ComAS2. ComAS2 will try to send the document via the AS2 protocol. If the customer’s application is not available, ComAS2 will retry transmission dependent on the configured timers (Send timeout, purge time) and increase the interval for every retry if there is still an entry in the Trace\_Tab. The maximum retry interval is 1 hour.

To give X.400 users the possibility to check the delivery of documents to AS2 partner they should request a receipt notification (RN). In the partner profile of the AS2 user

the item “Receipt Notifications requested in X.400 messages should be” should be set to “Send” so the MessageGate process will create the RN based on the MDN send by Customer’s AS2 application.

**Please be aware that the sending of Receipt-Notifications is a chargeable item (added to transfer volume)!**

In contrast to the File Interface, where the UNB reference number will be used for the Message ID and the subject of the X.400 message, the AS2 Gateway will map an internal process ID into the Message ID and for the subject of the X.400 message the corresponding subject of the AS2 message will be used.

## Automatically generated Status Report

As has already been mentioned the AS2 user can use *WebConfig* to request status reports (view or download) or to configure the delivery of status reports via AS2.

To receive status reports via AS2 the AS2 user must enable this feature in the menu item “Automatically generated Status Report”.

### AS2-X.400 Relation :: Automatically generated Status Report

☒ Enable automatically generated Status Report

#### AS2 Properties

##### IDs

AS2 ID: AS2 User AS2Tester001

AS2 ID: Status Report

##### Properties: X.400 ⇒ AS2

Default URL

Altern. URL

Compress ☒

Sign

Encrypt

#### Automatically generated Status Report

##### Properties

Only failed messages ☐

Days of the week

- ☒ Monday
- ☒ Tuesday
- ☒ Wednesday
- ☒ Thursday
- ☒ Friday
- ☒ Saturday
- ☒ Sunday

Daily start date  (MET/MEST, Format: hh:mm)

Daily end date  (MET/MEST, Format: hh:mm)

Schedule  Minutes (0=Only one time at daily start date) 30

Disposition

Direction

Format

In addition to the options described in Chapter 2.9.8 Configure automatically generated Status Report had to be assigned for the reporting process and like all other AS2 trading relations it is possible to define whether compression should be used and the data should be signed or encrypted. If necessary, an alternative URL will be configured, but an MDN will be not requested.

The first parameters of the status reports entries are equivalent to those of the file interface. However, there are some AS2 specific parameters added to these entries to describe the status of AS2 transaction:

Here are the details of those additional parameters:

AS2-ID:	AS2 Identifier of X.400 Partner
AS2-MIC:	Message Integrity Check identifier sent in Message Header when requesting signed MDN
AS2-Status:	<p>Possible values are:</p> <p>(MDN) not yet send → temporary status of message (Transaction X.400 → AS2) or asynchronous MDN (Transaction AS2 → X.400) which MessageGate transferred to ComAS2</p> <p>(MDN) still sending → Message (Transaction X.400 → AS2) or asynchronous MDN (Transaction AS2 → X.400) still in ComAS2 retry queue</p> <p>(async MDN) sent → Message (Transaction X.400 → AS2) or asynchronous MDN (Transaction AS2 → X.400) has been sent</p> <p>sync MDN sent → synchronous MDN has been sent (Transaction AS2 → X.400)</p> <p>sync MDN received → Message has been sent (Transaction X.400 → AS2) and synchronous MDN has been received</p> <p>async MDN requested → Message has been sent (Transaction X.400 → AS2) and asynchronous MDN has been requested</p> <p>async MDN received → Message has been sent (Transaction X.400 → AS2) and asynchronous MDN has been received</p> <p>deleted by order → Message (Transaction X.400 → AS2) has been deleted in Message Management by user request (in <i>WebConfig</i>).</p> <p>bypassed → Message (Transaction X.400 → AS2) has been moved directly into Message Management because “AS2 Bypass” had been enabled.</p> <p>send error – bypassed → Message (Transaction X.400 → AS2) has been moved to Message Management because send timer had been expired.</p> <p>send error – discarded → Message (Transaction X.400 → AS2) has been discarded in case of time out (only in Transfermode)</p> <p>async MDN missing - bypassed → Message (Transaction X.400 → AS2) was sent via AS2 but requested asynchronous MDN did not arrive in time. The message will be moved to Message Management. Depending on the setting of the parameter “Retries” the message transmission may be attempted several times (see also Sentcounter).</p> <p>async MDN missing - discarded → Message (Transaction X.400 → AS2) was sent via AS2 but requested asynchronous MDN did not</p>

arrive in time. The message has been discarded (only in Transfer-mode). Depending on the setting of the parameter “Retries” the message transmission may be attempted several times (see also Sentcounter).

send again requested → temporary status when reactivating a message (Transaction X.400 → AS2) transmission that was listed in Message Management.

error → temporary status if timer has expired but the message has not been moved yet into the Message Management

Message received → AS2 message received but there was no MDN request (Transaction AS2 → X.400)

AS2-Lastsent: Timestamp of the AS2 submission

Format: dd-mmm-yyyy hh-mm-ss +0100 (+0200 MEST)

Sentcounter: Shows how often AS2 Gateway has sent this message (Retries if timer for requested asynchronous MDN has expired)

MDN\_expected Date on which the requested asynchronous MDN is expected.

Format: dd-mmm-yyyy hh-mm-ss +0100 (+0200 MEST)

When requesting format “CSV-C” or “CSV-S”, the CSV file will show the following additional fields (in addition to those already defined in the Chapter 2.7.4 Syntax of Status reports (CSV structure)):

Field name:	Explanation:
AS2-ID	AS2 Identifier of X.400 Partner (within quotation marks)
AS2-MIC	Message Integrity Check identifier within quotation marks sent in Message Header when requesting signed MDN.
AS2-Status	Possible values see above (within quotation marks)
AS2-Lastsent	Timestamp (UTC/GMT) of the AS2 submission (dd.mm.yyyy hh:mm) (without quotation marks)
Send_counter	Shows how often AS2 Gateway has sent this message (without quotation marks)
MDN_expected	Timestamp (UTC/GMT) the AS2 Gateway expects requested asynchronous MDN (dd.mm.yyyy hh:mm) (without quotation marks)

## 7 SMTP MTA and MessageGate

### 7.1 General Information

An increasing number of X.400 domain administrators have requested a secure SMTP MTA access to the MailGate X.400 service, but do not want to change the X.400 address (Global Domain Identifier ⇒ GDI) of their domain. In order to satisfy this requirement, the MailGate X.400 service will now provide a SMTP MTA in addition to the existing X.400 MTA.

For this new SMTP MTA a modified MessageGate process will provide the mapping between RFC2822 (MIME or S/MIME) and X.400 messages. This new process will connect to the API of a local SMTP MTA and to the API of a local X.400 MTA to allow the message transfer between those two messaging environments.

This new MessageGate process (MGPMDF) will also store transaction logs in a database, but the status reports will be only available for the administrators of a customer's E-Mail service and not for the end user of this service. This process will also map SMTP notifications into X.400 reports and vice versa and so provide status information to the end users.

The local SMTP MTA is based on the PMDF MTA (Process Software) and available on OpenVMS.

The current version of the SMTP MTA will not support all mapping rules defined in RFC 2156/2157 (MIXER). Please read the following sections to find more details about these mapping rules.

### 7.2 Difference between File Interface and SMTP MTA users

A SMTP MTA user will not be connected directly to the MessageGate process because an E-Mail client is used for the data exchange with partners. Messages are sent to an appropriate MTA of BusinessMail X.400 using the SMTP protocol. The MessageGate process maps the RFC2822 messages into X.400 messages and records each transaction in a database relation (Trace\_Tab). Reports/Notifications received for sent messages will cause an update of the respective transaction entry. The E-Mail service administrator will see this change when requesting a status report, but the MessageGate process will also send the appropriate SMTP Reports (DSN or MDN) to the originator of the message.

We recommend using a SMTP MTA on customer side that supports DSN (Delivery Status Notification, see RFC 1891 or 3461). If this is the case, X.400 Delivery Notifications (DN) will be mapped to DSN and X.400 Receipt Notifications (RN) to MDN (Message Disposition Notifications, see RFC 3798) and vice versa.

The MessageGate process also allows you to configure the mapping of X.400 DN requests to MDN requests, but this approach is problematic as this relies on the E-Mail client/user to create and send an MDN. The administrator of the E-Mail service cannot control this. The X.400 standard requires that if a delivery notification has been requested, that an DN or NDN in the case, where a message has not been delivered, is sent within a specified time limit. MGPMDF will send an NDN with reason

code “Time expired” if an MDN is not received in time. The “delayed” MDN will be ignored.

There is also a difference in the behavior between the File Interface and the SMTP MTA concerning the message content. File Interface users have the choice to define, whether a Binary or Base64 format will be used within the binary MIME content. The SMTP MTA only supports Base64 format within the binary MIME content, but this is the recommended format for RFC2822 messages. For messages sent to X.400 the Base64 coded binary content will be decoded and attached as a BP14 without file name information or as a BP15/ FTAM Body Part. Also, S/MIME content will be unchanged and mapped into a single BP15/FTAM body part (see also 2.3.4 S/MIME secured content). Additional information can be found in section X.400 Message Structure.

This is an example of the default properties of an X.400 Domain using SMTP MTA and MGPMDF.

**MessageGate SMTP Relation :: Properties**

**User**

X.400 Address n-id=2060036; p=PMDF-TEST; a=VIAT-TEST; c=DE

**Properties**

If MDN requested in SMTP message request a X.400 Receipt Notification (RN)

If DN/NDN requested in X.400 message request a DSN in SMTP message

If RN/NRN requested in X.400 message request a MDN in SMTP message

Message Expiration 1440 Minutes

X.400 Content Type IPM84 IPM88

Bodypart IA5 Text  
Bilateral (Bodypart 14)  
ISO Latin 1  
Depends on context (variable, flat)  
Depends on context (variable, nested)

Mapping into SMTP address Use X.400 address syntax  
Mapping of all X.400 address elements  
Only mapping into natural address

Encode binary data base64

Purge time 120 Hours

Domain pmdftest.de

Comment

Ok Cancel

In the menu item “MessageGate SMTP Relation - Properties” new rules for the mapping of X.400 and SMTP report requests, additional X.400 body part mapping rules and a new option when mapping X.400 into a RFC2822 address can be found.

The MGPMDF will always map the request for a DSN into a request for an X.400 DN. MGPMDF will always request an X.400 NDN (Non-Delivery Notification), even though no DSN request has been made. Therefore, in case of a problem the E-Mail user will receive a DSN with an appropriate error code. For more details see B5. Mapping rules NDN to DSN.

Be aware that the X.400 standard allows the request of different report types for each individual recipient of an X.400 message. It is not possible to map this feature into an RFC2822 message. Therefore, if there is a message sent to different E-Mail recipients and in the X.400 message there is a request for a receipt notification for only some of the recipients, the SMTP MTA will not request an MDN.

## Partner entries

It is also possible to configure specific partner entries for the domain, but those changed properties will be applied to all SMTP MTA users of that particular domain when sending mails to this X.400 address. Be aware that the SMTP MTA only uses the mapping rules configured in the partner entries if the message has only one recipient. If there is more than one recipient in a message the SMTP MTA will use the rules configured in the domain properties.

## Address Mapping

In the menu item described above you will also find the domain name used for the mapping of RFC2822 domain elements to X.400 ORaddress elements. For all the other X.400 address elements MGPMDF will use the mapping rules defined in RFC 2156/2157 (MIXER), where the given name and surname will be added to the Name part of the address field (e.g., to the left of the “@”) and the organization/ organization-units to the domain part (e.g., right of the “@”).

### Example: Address mapping (using the domain in the example)

X.400 address: c=de;a=viat;p=pmdf-test;o=testag;ou1=entwicklung;s=tester; g=erster

RFC2822 address: [erster.test@entwicklung.testag.pmdf-test.de](mailto:erster.test@entwicklung.testag.pmdf-test.de)

MGPMDF also offers an extended mapping for migration purposes only (replace existing X.400 MTA and use a SMTP MTA), where in addition to the GDI address elements (Country, ADMD, PRMD) the organization and organization units will be mapped to the domain part. Therefore, the following mapping roles are also possible:

X.400 address filter: c=de;a=viat;p=pmdf-test;o=testag;ou1=entwicklung

Internet mail domain: entwicklung.de

The other X.400 address elements will be added to the left and right hand of the “@” sign.

So when the SMTP MTA end user receives a message the recipient address will be in the “natural” RFC2822 address form.

The format of the originator address and if present, those of the other recipients will be defined in the configuration option “Mapping into SMTP address” available in the properties and partner relation entries.

By default the rule “Use X.400 address syntax” will force the SMTP MTA to place the whole X.400 originator address in the name part of the address field (ie. to the left of the “@” sign). The X.400 address elements will be separated using the “/” character. If there is a space or a special character in one of the X.400 address elements the whole X.400 address will be set between quotation marks (“”). Be aware that some of the E-Mail clients will not add the quotation marks while replying to the original message, so the submission of the message (refused by its own SMTP server) or the delivery (refused by SMTP MTA) will fail.



In such a case please add the quotation marks manually (see also information about Telefax gateway in the following text).

Example for the X.400 address syntax: /G=ipm/S=tester/O=testag/A=viaT/C=de/@bmx400.de

Alternatively, you may use the rule "Mapping of all X.400 address elements" where the SMTP MTA will try to map the X.400 address into a "natural" RFC2822 address. If the X.400 address includes elements or characters where there is no counterpart in the "natural" RFC2822 address the SMTP MTA will create a "mixed" address. For example, there is no mapping for the CommonName or Generation address element and the "+" character that is allowed in an X.400 address is not allowed in a "natural" RFC2822 address. A space within X.400 address elements will be replaced using the tilde sign "~" in the RFC2822 address.

Example for a "natural" RFC2822 address: [ipm.test@testag.bmx400.de](mailto:ipm.test@testag.bmx400.de)

Example for a RFC2822 address with X.400 address elements: /CN=ipm~tester/@testag.bmx400.de

The third selection of this option is "Only mapping into natural address". In this case the SMTP MTA will refuse a X.400 message if the X.400 originator address or the X.400 address of another recipient contains address elements or characters that does not allow the mapping into a "natural" RFC2822 address and it will create a NDN with the error "conversion not possible" (reason code 2, diagnostic code 8).

When sending messages from SMTP to X.400 it is possible to use a variation of address types. In case of a "natural" or a "mixed" address the SMTP MTA will create an X.400 address based on the rules configured in the database.

A SMTP MTA user can also address other BusinessMail X.400 gateway solutions. When addressing the Telefax gateway it is necessary to set the X.400 address between quotation marks(") otherwise the ":" character in the X.400 address element "DDA" will cause problems:

Example: "/X121= 0391580217255/DDA:Service=FAX/A=viat/C=de/"@BMX400.DE

A SMTP MTA user may also add an alias to the RFC2822 address of an X.400 partner while sending a mail. The SMTP MTA will ignore this alias and also the alias of the originator address when transferring a SMTP message into the X.400 world. The addresses in a delivered SMTP message will not contain an alias.

If the X.400 partner receives a message MGPMDF will have mapped the RFC2822 address into an unambiguous X.400 originator address using the rules stored in the profile of the originator's domain entry. The original originator RFC2822 address (the first 64 characters) will be mapped into the X.400 address element "Freeform name" (Teletex character set), but the mailbox address will be set into square brackets

Aliasname [[givenname.surname@domain](mailto:givenname.surname@domain)]

to avoid compatibility problems with older X.400 clients. Microsoft Outlook also uses this kind of address structure within messages.

## X.400 Message Structure

RFC 2156/2157 (MIXER) describes the rules to map MIME content type into X.400 body parts. These conversion rules have been largely implemented in the SMTP MTA and MGPMDF and therefore also supports the X.400 body part of the type "message".

When using the default value of the mapping body parts option “variable/flat” the nested RFC2822 message contents (multipart/xxx nested in multipart/mixed) will be mapped without nesting (e.g., flat) into an X.400 message. The MIME contents will be mapped into X.400 body parts depending on their order in the RFC2822 message. If the RFC2822 message includes a MIME content type multipart/alternate (e.g., the RFC2822 message text is present as text as well as HTML content), the MIME contents will be mapped into appropriate X.400 body parts without any information regarding the identical content.

To map the information about this nested structure into an X.400 message the rule “variable/nested” must be selected. The nested MIME Content will be mapped into an X.400 message body part where the information about the structure is displayed in the subject field of the message, e.g., alternative message text body parts or HTML documents with imbedded graphics.

In contrast to the File Interface and AS2 the SMTP MTA will accept, in addition to MIME text content with character set ISO-Latin-x (x=1-9) and IA5, also other character sets (e.g., ISO-Latin-15, Windows-1252 or UTF-8). However, if there is no counterpart in X.400 the SMTP MTA will create a BP15 General Text ISO-Latin-1 (ISO 8859-1) body part and will add a MIME header with information about the original character set in front of the text.

When sending secured (signed and/or encrypted) data within a S/MIME content in the SMTP/ RFC2822 message the SMTP MTA will map this content in the X.400 message into a single BP15/ FTAM body part (see also Chapter 2.3.4 S/MIME secured content). The data in the S/MIME content will remain unchanged except transfer encoding will be converted from “Base64” to “Binary” for the encrypted (enveloped) content. Similarly, when receiving an X.400 message with secured data the SMTP MTA will also map the S/MIME content sent in the BP15/FTAM body part unchanged into the SMTP/ RFC2822 message. However, the transfer encoding is converted from “Binary” to “Base64”. If the E-Mail Clients are not able to process the Content-Transfer-Encoding “Binary” of the documents sent in the S/MIME content they must ask their X.400 partner to use an appropriate transfer encoding (quoted printable for text content and Base64 for binary content and for the signature) when creating the S/MIME content of the X.400 message.

## Forwarded Messages

In contrast to the File Interface, that does not support *forwarded messages*, the SMTP MTA will transfer unsecured messages in the following manner:

- a) An X.400 forwarded message (attachment of type message) to a SMTP MTA user will be stored as Content-Type: message/rfc822 (see RFC 2046). The included X.400 body parts will be imbedded for example in content type multipart/mixed. It is also possible to have nested X.400 messages imbedded as message/rfc822.
- b) There is no standard that defines the structure of a forwarded message within an RFC2822 message so MGPMDF will map the header of a forwarded message into a text body (normally the first text body) of an X.400 message and will not map it to a message body part. Only if the forwarded RFC2822 message is imbedded as a Content-Type: message/rfc822 and the option for mapping body part is “variable/flat” or “variable/nested” it will be mapped into an X.400 message body part. If the rule is set to “IA5-text”, “Bilateral” or “ISO-Latin-1” the MIME structure of a forwarded message will be mapped into an appropriate X.400 body part. The rule “IA5-text” or the option X.400 content type is “IPM84” may cause a loss of information.

## Message-ID

When mapping a received X.400 message into a RFC2822 message MGPMDF will use the X.400 Message-ID (P2 Message-ID, that the X.400 Client has defined) with a maximum length of 64 characters and will add, separated by a “#” character a 16 characters long extension to ensure an unambiguous RFC2822 Message-ID. Hence there should be no problem to update the transaction entry when receiving reports generated by SMTP mail services or SMTP mail clients and generate the appropriate X.400 reports.

Spaces within the X.400 Message-ID will be replaced by an underscore (“\_”) in the RFC2822 Message-ID. Some RFC2822 mail clients have problems with a Message-ID that includes a space character.

Example.: Message-ID: <341\_11/11/25#EA4A4CI404G0LBPN@tstmt\_pmdf.telebox400.de>

When mapping an RFC2822 message to an X.400 message the MGPMDF uses the first 64 characters of a RFC2822 Message-ID to build the X.400 P2 Message-ID. The SMTP mail service administrator should ensure that this Message-ID is unambiguous otherwise MGPMDF will not be able to update the transaction entry correctly and to send the appropriate RFC2822 notifications.

## Status report

To facilitate the tracking of transferred messages for the SMTP mail service administrators the message entry in the status report will include the originator and the recipient address. The entry will also show the RFC2822 Message-ID (instead of the order id), the X.400 Message-ID (the Message-ID of File Interface) and the MTS-id (generated by the X.400 MTA). The other values are equivalent to those of the MessageGate File Interface. So, for messages sent from X.400 to SMTP the status report will not show the submission time of the X.400 client, but the field “Received:” will show the time at which MGPMDF has received the message to forward it to the E-Mail recipient. For messages sent from SMTP to X.400 the field “Send:” will show the time at which MGPMDF had sent the X.400 message.

The status entry for a message sent from SMTP to X.400 will start with “To:”, “Cc:” or “Bcc:” and those sent from X.400 to SMTP with a “From:” address field. The additional recipients will be visible in the message header only. Only in the case where there are several recipients from the same domain, there will be an appropriate number of entries in the Trace Tab.

If the RFC2822 address includes an alias, the address part used for routing will be set in angle brackets (<>). The address will be not set in angle brackets if there is no alias.

Example for a message sent from SMTP to X.400:

```
To: erster tester </G=erster/S=tester/O=test-ag/A=viaT/C=de/@bmx400.DE>
From: test@pmdf-test.de
SMTP-Msg-ID: <4F1D739B.1070307@pmdf-test.de>
X400-Msg-ID: 4F1D739B.1070307
MTS-ID: FE19811D11E145D906005F96
Sent: 23-Jan-2012 15:50:37 +0100
Delivered: 23-Jan-2012 15:50:37 +0100
```

The RFC2822 address will be set in quotes if it includes spaces.

When downloading the Status report into a CSV structured file beside the fields/values already offered to MessageGate File Interface (e.g., From: and To:) it will provide at the end of the line the additional fields To: and From: and so the information about the originator and the recipient of this SMTP mail. This structure will use the same logic you will find in the better readable structure described on the last page. So, in case of the direction X.400 to SMTP there will be a value in the “first” From: and in the “second” To: field. For the direction SMTP to X.400 there will be a value in the “first” To: and the “second” From: field. In both cases the “Rcpt Type” field will define the type of recipient (To:, Cc: or Bcc:).

For editorial reasons this page is empty!

## 8 Implementation of MessageGate solution

### 8.1 Using Standard E-Mail Clients

#### 8.1.1 Test with Outlook Express for older Windows OS

Outlook Express can be used for a first test of the MessageGate functionality (create a message and send it via MessageGate or analyze a delivered message and its attachments). Outlook Express provides the capability of storing a message in a text file (Extension \*.eml) or to import such a message file.

First configure an E-Mail account in Outlook Express and enter the X.400 address in the "Display name" parameter (separate address fields using a semi colon ";" and do not use quotation marks for the address) and for E-Mail address enter the value [User-ID@viat.de](mailto:User-ID@viat.de) or [x@viat.de](mailto:x@viat.de) where User-ID is the local identifier of MessageGate account (e.g., [58111@viat.de](mailto:58111@viat.de)). In the fields "Incoming mail server" and "Outgoing mail server" you may enter "test" because these values will be not used.

Now add a partner address into contacts and enter the X.400 address in the last name field (separating the x.400 address elements by using semicolons ";" and not using quotation marks for the address) and define an E-Mail address using the values [User-ID@viat.de](mailto:User-ID@viat.de) (e.g., [58111@viat.de](mailto:58111@viat.de)) or [x@viat.de](mailto:x@viat.de). For the first test, use the address of your own MessageGate account so that you receive your sent test message directly in your own directory.

You are now able to create a message and to configure contacts. Please set the format of the message to plain text. Now add all types of attachments, a selection of characters in the text part of message and a subject. Store the message using "File → send later" into Outbox. Open Outbox, select mail, choose "File → Save as..." and define the name and the path where text file should be stored. Outlook Express will not add a message identifier in a draft message so you should use a file name/order identifier (e.g., M\_test00001.eml) that is also suitable as an X.400 message identifier. Rename the file to "\*.tmp" or upload it directly to the MessageGate directory using SFTP or HTTPS/WebDAV. When the file transfer is finished, rename the file to "\*.IN" so that MessageGate processes this file and sends the message. If you have addressed your own account a "M\_\*.OUT" file will be delivered into your MessageGate account directory.

You can now download this file, rename it to "\*.eml" and open it with Outlook Express. You will see that the message identifier of this mail is the original Order-ID. Outlook Express will be able to handle binary attachments where the content encoding is Base64 or Binary. You can configure the encoding used for a message in the base communication profile or in the partner profile.

#### 8.1.2 Test with Mozilla Thunderbird

Using Mozilla Thunderbird, it is also possible to create a message to test the MessageGate File Interface and to store this in a text file. Thunderbird will add a message identifier when storing a draft message but will not open attachments of imported messages if the content encoding is Binary.

You should first configure a user account and enter the value [User-ID@viat.de](mailto:User-ID@viat.de) (e.g., [58111@viat.de](mailto:58111@viat.de)) or [x@viat.de](mailto:x@viat.de) in the field E-Mail Address. For the field "Your name:" you should use the X.400 address of your MessageGate account (separate the X.400 address elements using a semicolon ";" and do not use quotation marks for the address). For the field "Incoming server" you may use the string "test" as this value is not used.

You can now add a partner address into the address book by entering the X.400 address (separate the X.400 address elements using a semicolon ";" and do not use quotation marks for the address) in the field "Display" and [User-ID@viat.de](mailto:User-ID@viat.de) (e.g., [58111@viat.de](mailto:58111@viat.de)) or [x@viat.de](mailto:x@viat.de) in the field "E-Mail:". For the first test use the address of your own MessageGate account so that you receive your sent test message directly in your own directory.

now able to create a message and adding a configured address. Please set the format of the message to plain text (Options → Format → Plain Text only). Now add all types of other attachments, a selection of characters in the text part of message and a subject. Store the message using "File → send later" into Unsent. Open Unsent, select mail, choose "File → Save as... → File" and define the name and the path where text file should be stored. Rename the file to "\*.tmp" or upload it directly to the MessageGate directory using FTP or HTTPS/WebDAV. When the file transfer is finished, rename the file to "\*.IN" so that MessageGate processes this file and sends the message. If you have addressed your own account, you will find a "M\_\*.OUT" file in your MessageGate account directory.

You can now download this file, rename it to "\*.eml" and open it with Thunderbird. You will see that the message identifier of this mail is the original Order-ID. Thunderbird is not able to handle binary attachments where content encoding is Binary. You must configure that encoding Base64 will be used when delivering messages in the base communication profile or in the partner profile.

### 8.1.3 Test with Microsoft Live Mail for newer Windows OS

On newer Windows OS, where Outlook Express is not longer available, it is possible to install the program Windows Live Mail 2012 as part of the optional package Microsoft Essentials (supported until February 2017). Now you can use Live Mail for a first test of the MessageGate functionality (create a message and send it via MessageGate or analyze a delivered message and its attachments). Like Outlook Express the Live Mail client also provides the capability of storing a message in a text file (Extension \*.eml) or to import such a message file.

First configure an E-Mail account while selecting the account tab and enter in the field "Email address" the values "User-ID@viat.de" or "x@viat.de" where User-ID is the local identifier of MessageGate account (e.g., 58111@viat.de). In the field "Display name" enter the X.400 address (separate address fields using a semi colon ";" and do not use quotation marks for the address). In the fields "Password", SMTP server" and "POP3 server" you may enter "Test" because they are not used for the tests.

Now add a partner address into contacts and enter the X.400 address in last name (separating the x.400 address elements by using semicolons ";" and not using quotation marks for the address) and define an Email address using the values User-ID@viat.de (e.g., 58111@viat.de) or x@viat.de. For the first test, use the address of your own MessageGate account so that you receive your sent test message directly in your own directory.

You are now able to create more messages and add configured contact. Please set the format of the message to plain text. Now add all types of attachments, a selection of characters in the text part of message and a subject. Store the message while choosing “File □ Save as...” and define the name and the path where the text file should be stored. Live Mail will not add a message identifier in a draft message, so you should use a file name/order identifier (e.g., M\_test00001.eml) that is also suitable as an X.400 message identifier. Rename the file to “\*.tmp” or upload it directly to the MessageGate directory using SFTP or HTTPS/WebDAV. When the file transfer is finished, rename the file to “\*.IN” so that MessageGate processes this file and sends the message. If you have addressed your own account a “M\_\*.OUT” file will be delivered into your MessageGate account directory.

You can now download this file, rename it to “\*.eml” and open it with Live Mail. You will see that the message identifier of this mail is the original Order-ID. Live Mail will be able to handle binary attachments where the content encoding is Base64 or Binary. You can configure the encoding used for a message in the base communication profile or in the partner profile.

## 8.2 Designing a MessageGate solution

There are many libraries and tools available to process SMTP/ MIME syntax for many Operating systems and programming languages (both commercial and open source). Hence, creating and processing message files should not pose a problem.

Please invest some time in the design of your solution about the handling of unsuccessful transactions. X.400 provides different kinds of reports/notifications for you to implement the tracking of the message transaction status. MessageGate will provide this information in Status Reports.

If you do not want to check the delivery of a message, you should at least check the Non-Delivery report or a possible send failure and handle this condition in your application (alarming or resend). If you use the central EDI function, we also recommend checking if the MessageGate process had refused incoming messages/ EDIFACT documents. All this information can be found in the Status Reports.

You have three different communication options for access your MessageGate directory, SFTP, HTTPS/WebDAV and HTTP/Web Service. If you order MessageGate with reduced functionality the access via HTTPS/WebDAV would be the best solution because here a normal browser would be able to download a delivered message. Depending on configured purge time the delivered messages (and if configured status reports) will be deleted automatically.



For editorial reasons this page is empty!

## Appendix A X.400 Address elements

This appendix shows the list of all X.400 Address elements you may use in alias of "To:" and "FROM:" and in host-based partner profiles:

C=	Country code (3 Characters Printable String)
A=	Administrative Domain Name (ADMD, 16 Characters Printable String)
P=	Private Domain Name (PRMD, 16 Characters Printable String)
O=	Organization (64 Characters Printable or Teletex String)
OU1=	Organization Unit 1 (32 Characters Printable or Teletex String)
OU2=	Organization Unit 2 (32 Characters Printable or Teletex String)
OU3=	Organization Unit 3 (32 Characters Printable or Teletex String)
OU4=	Organization Unit 4 (32 Characters Printable or Teletex String)
S=	Surname (40 Characters Printable or Teletex String)
G=	Given name (16 Characters Printable or Teletex String)
CN=	Common name (64 Characters Printable or Teletex String)
N-ID=	Box Identifier (UA-ID, 32 Characters Numerical)
X121=	Network Identifier (15 Characters Numerical)
T-ID=	Terminal Identifier (24 Characters Printable String)
I=	Initials (5 Characters Printable String)
Q=	Generation (Generation Qualifier, 3 Characters Printable String)
DDA:Type=Value	Domain Defined Attributes (Type 8 Characters = Value 128 Characters, both Printable or Teletex String , e.g., dda:service=fax)

*See Appendix D for more details about printable string characters.*

The following rules

Please remind the following rules for X.400 address:

1. The GDI (Global Domain Identifier), is made up of Country Name, ADMD name and PRMD Name, defines the mail system/mail service. It is necessary to add another address field to define the recipient. This might be Surname, Common name, or Unique UA ID.
2. The Unique UA ID might be used as an alternative to a mnemonic address that is made up of name and organizational elements.
3. When using a mnemonic address, it might be necessary to define several fields to ensure an unambiguous recipient address
4. An address element Personal Name (PN) that includes Surname, Givenname, Initials und Generation Qualifier might be used if the SMTP Gateway sends a message to internet.

## Specialties of *BusinessMail X.400 MailBox Service*

1. Unlike other X.400 services the *BusinessMail MailBox X.400* service does not require the use of **all** the address elements when addressing a user of this service. It is enough that the recipient address is unambiguous. However, the use of such a shortened address carries the inherent risk that it may become ambiguous if new *BusinessMail MailBox X.400* service users are provisioned. In addition, an existing trading relation may fail as a result. Hence, we recommend using the complete X.400 address with all the address elements or to address a user in the *BusinessMail MailBox X.400* service using the User-ID (MessageGate will use the X.400 address configured in the Database) or the Unique Agent ID (Box identifier).

2. When sending messages to a partner located on the Internet, who is using a SMTP client, the *BusinessMail X.400* SMTP gateway will be used. When addressing the SMTP gateway the GDI "C=de;A=viat-smtp" needs to be used. To conform to the X.400 standard the surname and if applicable the first name of the SMTP recipient needs to be mapped into the X.400 address element surname. In addition, the whole RFC2822 address will be mapped into a DDA address element. The type of this DDA element is "RFC-822" and the value is the Internet RFC2822 address of the recipient. Please also consider that the "@" is not a valid character in the printable string set so that this needs to be replaced with "(a)". <x@viat> needs to be added in the following address examples.

For example:

"c=de; a=viat-smtp; g=hans; s=meier; DDA:rfc-822=hans.meier(a)telekom.de"

3. If a fax is sent to your partner then the *BusinessMail X.400* Fax gateway is used. The GDI of this gateway is "C=de;A=viaT" and the fax number has to be written into the X.400 address element "X121". The DDA address element where type is "Service" and the value is "Fax" needs to be added.

For example:

"c=de;a=viaT;X121=061519992725;DDA:service=fax"

For additional information, please check <https://www.service-viat.de>.

## Appendix B: Error codes

### B1. Error codes of MessageGate Poller process:

Reason code	Error text	Description
0001	Invalid arguments	Internal error. Please contact Helpdesk for further information.
0002	Cannot separate sender ID	Internal error. Please contact Helpdesk for further information.
0003	Invalid file name	Internal error. Please contact Helpdesk or further information.
0004	File-OrderID too long	Order ID is longer than 26 Characters
0005	Cannot open file	File is locked by other process. Please upload file with extension “*.TMP” first and then rename it to “*.IN”.
0006	Cannot create file	Internal error. Please contact Helpdesk for further information.
0007	Invalid HDR in file	Internal error. Please contact Helpdesk for further information.
0008	Error writing body part file	Internal error. Please contact Helpdesk for further information.
0009	Error writing header file	Internal error. Please contact Helpdesk for further information.
0010	Cannot move	Internal error. Please contact Helpdesk for further information.
0011	Wrong parameter specified	File includes invalid values
0012	Empty file	File is empty. Please upload file with extension “*.TMP” first and then rename it to “*.IN”.
0013	Invalid content in status request file	Request for Status report includes invalid values.
0014	Invalid msg type	Invalid syntax in message structure
0015	Missing header element To:	Mandatory element recipient is missing
0016	Invalid SMTP address	“TO:” or “FROM:” address is invalid or incomplete (e.g., Alias or RFC2822 address part is missing)
0017	Missing header element Content-Type:	Mandatory element definition of data (content) is missing
0018	Missing header element Content-Transfer-Encoding:	Mandatory element definition of transfer encoding content is missing

9999	Status report request ignored	Status report request ignored because stored within xxxx time.
------	-------------------------------	--

## B2. MessageGate Error codes

Error code	Internal Text symbol	Description
134250499	SHM_EXISTS	shared memory already exists>
134250500	SHM_NOT_EXISTS	shared memory does not exist>
134250501	PRC_DULPNAM	process name %s already exists>
134250505	ATTRIB_INVALID	invalid or unsupported attribute>
134250506	BUFFER_EMPTY	buffer is empty>
134250507	BUFFER_OVERFLOW	buffer overflow>
134250508	BUFFER_TOO_SMALL	buffer too small for primitive>
134250509	NO_BUFFER	no buffer>
134250510	CHECKSUM_INVALID	invalid checksum: %s>
134250511	CLASS_EMPTY	pom_class holds no elements>
134250512	CLASS_END	end of class reached>
134250513	CLASSCTX_NULL	internal error: class context is null>
134250514	CLASSCTX_INVALID	internal error: invalid class context>
134250515	DESCR_NOT_FOUND	descriptor %s not found>
134250516	NO_DEVICE	no device available>
134250517	DIR_CREATE	cannot create directory %s>
134250518	DIR_NAME_INVALID	directory name invalid %s>
134250519	DIR_NOT_FOUND	directory not found %s>
134250520	DIR_NO_ACCESS	no access to directory %s>
134250521	DISK_FULL	disk is full %s>
134250522	DISK_NAME_INVALID	invalid disk name %s>
134250523	DISK_NOT_FOUND	disk not found %s>
134250524	DISK_NO_ACCESS	no access to disk %s>
134250525	DS_INIT	DS API function ds_init failed>
134250526	DS_SHUT	DS API function ds_shut failed>
134250527	DS_BIND	DS API function ds_bind failed>
134250528	DS_UNBIND	DS API function ds_unbind failed>
134250529	DS_ADD_ENTRY	DS API function ds_add_entry failed>
134250530	DS_MODIFY_ENTRY	DS API function ds_modify_entry failed>
134250531	DS_REMOVE_ENTRY	DS API function ds_remove_entry failed>
134250532	DS_SEARCH	DS API function ds_search failed>
134250533	ELEM_LENGTH_MISS	tried pom_write on an element created without length>
134250534	ELEM_NOT_FOUND	cannot find element of specified type %s>
134250535	ELEM_READONLY	tried to modify readonly element %s>
134250536	ELEM_NOT_PRESENT	element not present>
134250537	ELEM_MULTI_VALUED	element is multi-valued>
134250538	ENCOD_ANY	ANY syntax found in %s>
134250539	ENCOD_END	end of encoding; %s>
134250540	ENCOD_EXCEEDED	encoding exceeds 4 bytes length>

134250541	ENCOD_INVALID	invalid encoding; %s>
134250542	ENCOD_EOC_EXPECTED	expected EOC; %s>
134250543	ENCOD_INCOMPLETE	incompleted decode; %s>
134250544	ENCOD_LENGTH	element length exceeded; %s>
134250545	ENCOD_EMPTY	tried to encode an empty primitive; %s>
134250546	ENCOD_MANDATORY	missing mandatory element; %s>
134250547	ENCOD_LIMIT	limit exceeded; %s>
134250548	UNSUP_EXTID	ExtensionId %s is not supported>
134250549	ENTITY_ACCESS	invalid access method for entity>
134250550	ENTITY_ATTR	invalid type %s of entity attribute>
134250551	ENTITY_TYPE	invalid entity type>
134250552	ENTITY_SLOT_INV	invalid slot number %s for entity>
134250553	ENTITY_SLOT_NOFR	no slot free for entity>
134250554	ENTITY_TYPE_ATTR	expected attribute TYPE>
134250555	ENTITY_CMD_NOTSUPP	command not supported>
134250556	ENTITY_RESTART	can not restart entity %s>
134250557	ENTITY_ATTR_TAB	attribute description not found>
134250558	ENTITY_DUPLNAM	name for entity already exists>
134250559	ENTITY_MGMT	Master not active>
134250560	ENTITY_NOT_EXIST	entity not exist>
134250561	ENTITY_WILDCARD	wildcard not supported>
134250562	ENTITY_CREATE	can not create entity %s>
134250563	ENTITY_LIMIT	Entity %s exceeds restarting limit>
134250564	MGMT_SHUTDOWN	OMS system is down>
134250565	ENTITY_NORESTART	restarting not allowed>
134250566	ENTITY_ABNORMAL	Entity %s terminated abnormally>
134250567	ENTITY_ERROR	Entity %s terminated due to an error>
134250568	ENTRY_NOT_FOUND	found no or no more entry>
134250569	ENTRY_IGNORE	ignore this entry>
134250570	ENTRY_EXISTS	entry already exists>
134250571	ENTRY_ISCHILD	cannot delete child-entry without its parent>
134250572	ENTRY_SELECT	entry selected by MSK>
134250573	ENV_LOG	environment/logical %s not set>
134250574	EXPR_EMPTY	%s-expression is empty>
134250575	FEAT_NOT_SUPP_YET	feature not supported yet>
134250576	FILE_CONNECT	cannot connect to record access block of file %s>
134250577	FILE_CREATE	cannot create file %s>
134250578	FILE_DELETE	cannot delete file %s>
134250579	FILE_END	end of file detected %s>
134250580	FILE_BEGIN	beginning of file detected %s>
134250581	FILE_FREE	cannot release lock (possibly not set), file: %s>
134250582	FILE_LENGTH	attempt to read past end of file %s>
134250583	FILE_LOCK	cannot lock file %s>
134250584	FILE_NAME_INVALID	invalid filename %s>
134250585	FILE_NO_SUCH	no such file: %s>

134250586	FILE_OPEN	cannot open file %s>
134250587	FILE_READ	error reading on file %s>
134250588	FILE_SEEK	cannot seek to file position, file: %s>
134250589	FILE_TRUNCATE	cannot truncate file %s>
134250590	FILE_WRITE	error writing to file %s>
134250591	FILE_PARTIAL	cannot read as many bytes as asked for>
134250592	FUNC_NOT_IMPLM	function %s not implemented>
134250593	FUNC_SDS_NOT_EXIST	this function will never exist>
134250594	FUNC_SEQUENCE	invalid sequence of function-calls>
134250595	IPC_KEY	invalid key name %s>
134250596	IPC_LOCK_NOT_GRANT	lock not granted>
134250597	IPC_MBX_REMOVED	message queue is removed>
134250598	IPC_CREATION	process creation error (%s)>
134250599	IPC_MBX	message queue error (%s)>
134250600	IPC_LOCK	locking error (%s)>
134250601	IPC_SHM	shared memory error (%s)>
134250602	IPC_LNM	logical name error (%s)>
134250603	IPC_NO_LOGTAB	logical name table %s for mailbox does not exist>
134250604	IPC_NO_PRIV	insufficient privilege for IPC operation>
134250605	IPC_USRQUOTA	quota of user %s failed (%s)>
134250606	IPC_USER_UNKNOWN	user %s unknown>
134250607	IPC_LOGNAM	error on logical name passed through VMS function>
134250608	LOCSUBM_VIOLATED	non local submission>
134250609	MATCH_INAPPR	inappropriate matching>
134250610	MEMORY_INSUFF	no memory>
134250611	MODE_LOCK_UNKNOWN	unknown locking mode %s>
134250612	MODE_OPER_UNKNOWN	got unknown operation mode %s>
134250613	MSG_CONTENT_LONG	content too long>
134250614	MSG_CONTENT_MULTI	more than one content>
134250615	MSG_CONTENT_NONE	content missing>
134250616	MSG_ENV_MISS	envelope missing>
134250617	MSG_ENV_WHAT	unknown element in envelope>
134250618	MSG_IFC_NONE	child entry without IFC entry encountered>
134250619	MSG_MISSING	message missing>
134250620	MSG_NOT_REC	no message received>
134250621	MSG_NOT_SEND	no message sent>
134250622	MSG_ORIGIN_MULTI	more than one originator>
134250623	MSG_ORIGIN_NONE	originator missing>
134250624	MSG_ORR_MULTI	more than one originator report requested>
134250625	MSG_ORR_NONE	no originator report requested>
134250626	MSG_RECIP_NONE	recipient missing>
134250627	MSG_RECNAME_MULTI	more than one recipient name>
134250628	MSG_RECNAME_NONE	no recipient name>
134250629	MSG_REPORT_WHAT	unknown element in report>
134250630	MSG_ORIGIN_INVALID	invalid message originator>

134250631	MTA_CANCEL	MTA function ma_cancel failed,%s>
134250632	MTA_CLOSE	MTA function ma_close failed,%s>
134250633	MTA_FINISH_DEL	MTA function ma_finish_delivery failed,%s>
134250634	MTA_NOT_AVAIL	MTA not available>
134250635	MTA_NO_MPDU	MTA has not MPDU %s>
134250636	MTA_OPEN	MTA function ma_open failed,%s>
134250637	MTA_START_DEL	MTA function ma_start delivery failed,%s>
134250638	MTA_SUBMIT	MTA function ma_submit failed,%s>
134250639	MTA_WAIT	MTA function ma_wait failed,%s>
134250640	MTA_AGENTNAME	MTA agent name invalid>
134250641	OCOM_PORT_INVALID	Invalid port number>
134250642	OCOM_FREE	The osak has queued the request. There is free block>
134250643	OCOM_QUEUED	The osak has queued the request>
134250644	OCOM_DISRUPTED	A disruptive event has occurred>
134250645	OCOM_INVAEI	The application entity invocation is invalid>
134250646	OCOM_INVDEFCTXT	The default context response is invalid>
134250647	OCOM_INVFUNC	The call is invalid>
134250648	OCOM_INVFUS	The functional units are invalid>
134250649	OCOM_INVID	The activity identifier is too long>
134250650	OCOM_INVPCTXT	The presentation context list is invalid>
134250651	OCOM_INVSYNCPNT	The synchronization point serial number is invalid>
134250652	OCOM_NOPROCINFO	The is no process-id and no process-name>
134250653	OCOM_NOSYNCPNT	The synchronization point serial number is missing>
134250654	OCOM_TRANSERR	There is error in transport provider>
134250655	OCOM_NOEVENT	There is no event>
134250656	OCOM_INCPCI	The PCI is not complete>
134250657	OCOM_INSFWS	There is not enough workspace in the parameter block>
134250658	OCOM_NOBUFFERS	There are not enough user data buffers>
134250659	OCOM_OVERFLOW	Too much user data has been sent for session v-1>
134250660	OCOM_INVTOKEN	The token setting is invalid>
134250661	OCOM_INVEVENT	There is invalid event>
134250662	OM_CREATE	Object Management function om_create failed,%s>
134250663	OM_DELETE	Object Management function om_delete failed,%s>
134250664	OM_GET	Object Management function om_get failed,%s>
134250665	OM_INSTANCE	Object Management function om_instance failed,%s>



134250666	OM_PUT	Object Management function om_put failed,%s>
134250667	OM_READ	Object Management function om_read failed,%s>
134250668	OM_WRITE	Object Management function om_write failed,%s>
134250669	OPER_UNKNOWN	Operation %s is unknown>
134250670	PARAM_INVALID	invalid parameter %s>
134250671	PARAM_NULL	parameter %s was a NULL pointer>
134250672	LENGTH_INVALID	invalid length %s>
134250673	PORT_INVALID	invalid port %s>
134250674	PRIV_MISSES	process misses privilege>
134250675	PVERS_INVALID	protocol version invalid>
134250676	QUEUE_EMPTY	empty queue>
134250677	QUOTA_EXHAUSTED	process quota exhausted>
134250678	RANGE_REVERSED	range reversed>
134250679	RANGE_NOTVALID	range out of bounds>
134250680	RESTR_EXCEEDED	restrictions exceeded>
134250681	RULE_UNKNOWN	rule %s is unknown>
134250682	SERVER_BUSY	server is busy>
134250683	SERVER_DOWN	server is down>
134250684	SIGNAL_NOT_SUPP	Signal (interrupt) is not supported: %s>
134250685	SQL_ERROR	SQL error: %s>
134250686	STATE_INVALID	current facility state does not allow this operation>
134250687	STATUS_NEW_DEL	tried to delete a NEW-message>
134250688	STATUS_CHANGE	change from actual status to given is not supported>
134250689	STATUS_UNKNOWN	status %s is not known>
134250690	STRUCT_USER_ERROR	got wrong structures from user agent>
134250691	SYNTAX_DIFFERENT	different OM_syntax between pom_add and pom_write>
134250692	SYNTAX_UNKNOWN	given OM_syntax %s is unknown>
134250693	SYNTAX_ERROR	syntax error>
134250694	TABLE_FULL	table overflow>
134250695	TABLE_UNKNOWN	tried to lock an unknown MDB-table %s>
134250696	TAG_TOO_BIG	tag too big .gt. 4 bytes>
134250697	TRANSACTION_INACTIVE	Transaction %s inactive>
134250698	TRANSACTION_ACTIVE	Transaction %s active>
134250699	TRANSACTIONID_WRONG	Transaction Id %s wrong>
134250700	TYPE_DIFFERENT	different OM_type between pom_add and pom_write>
134250701	USER_AMBIGUOUS	user name is ambiguous>
134250702	USER_NEW_NOT_SPEC	existing user name has same elements plus some other>
134250703	USER_OLD_NOT_SPEC	existing user name has same elements but fewer>
134250704	USER_PWD_INVALID	invalid password given by user>

134250705	USER_UNKNOWN	user name is unknown>
134250706	USER_DOUBLE_LOGIN	user is already logged in %s>
134250707	USER_ACTIV_NOT_DEL	cannot delete user with status ACTIVE>
134250708	USER_NAME_NOT_MOD	orname elements modify only single user>
134250709	USER_PWD_EXPIRED	user password expired>
134250710	USER_SRVC_EXPIRED	user service expired>
134250711	USER EDI_DENIED	no agreement between EDI sender and receiver>
134250712	USER EDI_NO_SND	Sending Partner not found>
134250713	USER EDI_NO_REC	Receiving Partner not found>
134250714	USER EDI_NO_AGROP	Agreement for open receiver not found>
134250715	USER EDI_NO_AGRCL	Agreement for closed receiver not found>
134250716	USER_MAX_LOGIN_FAILS	maximum login fails reached>
134250717	DOMAIN_AMBIGUOUS	domain name is ambiguous>
134250718	ORNAME_INVALID	no valid addressing form specified>
134250719	USER EDI_NO_RUT	Routing Partner not found> !
134250720	USER_DISCONNECT_NOT_DEL	cannot delete user with status DISCONNECTED>
134250721	VERSION_INVALID	version invalid>
134250722	VALUE_TOO_BIG	value too long>
134250723	WRONG_VALUE	wrong values: %s>
134250724	WRONG_VALUE_TYPE	value type is unknown: %s>
134250725	WRONG_VALUE_LENGTH	value length is incorrect>
134250726	WRONG_VALUE_NUMBER	digits in value is not a number>
134250727	WRONG_VALUE_MAKEUP	make-up of value is wrong>
134250728	WRONG_VALUE_RANGE	value out of range>
134250729	WRONG_VALUE_SYNTAX	wrong value syntax>
134250730	WILDCARD_INVALID	wildcard not allowed>
134250731	DECODE_END	end of decoding>
134250732	NO_SUCH_SND	no such sender>
134250733	NO_SUCH_REC	no such recipient>
134250734	TP_AMBIGUOUS	trading partner is ambiguous>
134250735	NO_SUCH_RUT	no such router>
134250736	NO_DEFAULT_VALIDFOR	no default validfor-entry available>
134250737	HAVE_SPECIAL_VALIDFOR	special validfor-entries still exist>
134250738	LOGONNAME_AMBIGUOUS	logonname is ambiguous>
134250739	MANDATORY_ATTRIBUTE	mandatory attribute missing>
134250740	MANDATORY_SECTION	mandatory section missing>
134250741	MANDATORY_TABLE	mandatory table missing>
134250742	BCKP_PURG	Backup/Purger/Repair cannot run parallel>
134250743	TIME_RELATIVE	cannot convert relative time into UTC format>
134250744	CFG_TOKEN_UNKNOWN	found unknown token in config file>
134250745	CFG_TOKEN_AMBIGUOUS	found ambiguous token in config file>

134250746	CFG_SYNTAX	found token without '=' in config file>
134250747	CFG_VALUE_UNKNOWN	value not found in conversion table>
134250748	CFG_VALUE_AMBIGUOUS	value has ambiguous conversion>
134250749	CFG_VALUE_SYNTAX	syntax error in config file>
134250750	CFG_VALUE_NOTMULTI	config value is not multi valued>
134250751	CFG_TABLE_SYNTAX	error in conversion table>
134250752	EDPRS_INVIC	invalid interchange syntax>
134250753	EDPRS_INVTRAIL	invalid interchange trailer>
134250754	EDPRS_INVHEAD	invalid interchange header>
134250755	EDPRS_RUBBISH	too many useless characters>
134250756	EDPRS_CTRLREF	control reference mismatch>
134250757	EDPRS_TAGLONG	found too long EDI tag>
134250758	EDPRS_ELEMLONG	found too long EDI element>
134250759	EDPRS_TOOMANYIC	too many Interchanges>
134250760	PARSE_BREAK	break current parsing>
134250761	UTL_LOCK_CREATE	Lock create failed>
134250762	UTL_LOCK_DESTROY	Lock destroy failed>
134250763	UTL_LOCK	Locking failed>
134250764	UTL_UNLOCK	Unlocking failed>
134250765	POMSORT_IGNORED	pom type %s ignored (reflist: %s)>
134250766	STOP_RESOURCE	out of resources>
134250767	ADDINFO	Additional info: %s>
134250768	RSC_NOT_FOUND	Resource information not found>
Under certain circumstances you will see the error code of another host process instead of that of MessageGate (see the following example). Please send this error code to the Helpdesk to analyze this problem.		
159416462		MTA function ma_submit failed,%s

## B3. MTA Error codes (Non Delivery Notification)

### Error codes in NDN:

Reason code	X.400 Standard	Description
0	Transfer-failure	Indicates that, while the MTS was attempting to deliver or probe delivery of the subject-message, some communication failure prevented it from doing so.
1	Unable-to-transfer	Indicates that, due to some problem with the subject itself, the MTS could not deliver or probe delivery of the subject-message.
2	Conversion-not-performed	Indicates that a conversion necessary for the delivery of the subject-message could not (or cannot) be performed.
3	Physical-rendition-not-performed	Indicates that the PDAU was unable to physically render the subject-message.
4	Physical-delivery-not-performed	Indicates that the PDS was unable to physically deliver the subject-message.
5	Restricted-delivery	Indicates that the recipient subscribes to the restricted-delivery element-of-service (as defined in Recommendation X.400) which prevented (or would prevent) the delivery of the subject-message.
6	Directory-operation-unsuccessful	Indicates that the outcome of a required directory operation was unsuccessful.
7	deferred-delivery-not-performed	Indicates that a request for deferred delivery of the subject-message was unable to be performed;
8	transfer-failure-for-security-reason	Indicates that, while the MTS was attempting to deliver or probe delivery of the subject-message, a security failure prevented it from doing so
99	MessageGate Poller Error (non-standard)	No value of X.400 Standard: Error reported by MessageGate Poller while processing an “*.IN”-file. In Diagnostic Code you will find the Poller error code (see B1.).

**Reason codes in NDN**

Diagnostic Code	X.400 Standard	Description
0	Unrecognized-OR-name	The recipient-name argument of the subject does not contain an OR-name recognized by the MTS.
1	Ambiguous-OR-name	The recipient-name argument of the subject identifies more than one potential recipient (e.g., is ambiguous).
2	MTS-congestion	The subject could not be progressed, due to congestion in the MTS.
3	Loop-detected	The subject was detected looping within the MTS.
4	Recipient-unavailable	The recipient MTS-user was (or would be) unavailable to take delivery of the subject-message.
5	Maximum-time-expired	The maximum time for delivering the subject-message, or performing the subject-probe, expired.
6	Encoded-Information-Types-unsupported	The encoded-information-types of the subject-message are unsupported by the recipient MTS-user.
7	Content-too-long	The content-length of the subject-message is too long for the recipient MTS-user to take delivery (exceeds the deliverable-maximum-content-length).
8	Conversion-impractical	A conversion required for the subject-message to be delivered is impractical.
9	Implicit-conversion-prohibited	A conversion required for the subject-message to be delivered has been prohibited by the originator of the subject.
10	Implicit-conversion-not-subscribed	A conversion required for the subject-message to be delivered has not been subscribed to by the recipient.
11	Invalid-arguments	One or more arguments in the subject were detected as being invalid.

12	Content-Syntax-error	A syntax error was detected in the content of the subject-message (not applicable to subject-probes).
13	Size-constraint-violation	Indicates that the value of one or more parameter(s) of the subject violated the size constraints defined in the X.411 Recommendation, and that the MTS was not prepared to handle the specified value(s).
14	Protocol-violation	Indicates that one or more mandatory argument(s) were missing from the subject.
15	Content-type-not-supported	Indicates that processing of a content-type not supported by the MTS was required to deliver the subject-message.
16	Too-many-recipients	Indicates that the MTS was unable to deliver the subject-message due to the number of specified recipients of the subject-message.
17	No-bilateral-agreement	Indicates that delivery of the subject-message required a bilateral agreement where no such agreement exists.
18	Unsupported-critical-function	Indicates that a critical function required for the transfer or delivery of the subject-message was not supported by the originating-MTA of the report.
19	Conversion-with-loss-prohibited	A conversion required for the subject-message to be delivered would have resulted in loss of information; conversion with loss of information was prohibited by the originator of the subject.
20	Line-too-long	A conversion required for the subject message to be delivered would have resulted in loss of information because the original line length was too long.
21	Page-split	A conversion required for the subject-message to be delivered would have resulted in loss of information because an original page would be split.

22	Pictorial-symbol-loss	A conversion required for the subject-message to be delivered would have resulted in loss of information because of a loss of one or more pictorial symbols.
23	Punctuation-symbol-loss	A conversion required for the subject-message to be delivered would have resulted in loss of information because of a loss of one or more punctuation symbols.
24	Alphabetic-character-loss	A conversion required for the subject-message to be delivered would have resulted in loss of information because of a loss of one or more alphabetic characters.
25	Multiple-information-loss	A conversion required for the subject-message to be delivered would have resulted in multiple loss of information.
26	Recipient-reassignment-prohibited	Indicates that the MTS was unable to deliver the subject-message because the originator of the subject prohibited redirection to a recipient - assigned-alternate-recipient.
27	Redirection-loop-prohibited	The subject-message could not be redirected to an alternate-recipient because that recipient had previously redirected the message (redirection-loop).
28	DI-expansion-prohibited	Indicates that the MTS was unable to deliver the subject-message because the originator of the subject prohibited the expansion of DLs.
29	No-dl-submit-permission	The originator of the subject (or the DL of which this DL is a member, in the case of nested DLS) does not have permission to submit messages to this DL.
30	DI-expansion-failure	Indicates that the MTS was unable to complete the expansion of a DL.
31	Physical-rendition-attributes-not-supported	The PDAU does not support the physical-rendition-attributes requested.

32	Undeliverable-mail-physical-delivery-address-incorrect	The subject-message was undeliverable because the specified recipient postal-OR-address was incorrect.
33	Undeliverable-mail-physical-delivery-office-incorrect-or-invalid	The subject-message was undeliverable because the physical-delivery-office identified by the specified recipient postal-OR address was incorrect or invalid (does not exist).
34	Undeliverable-mail-physical-delivery-address-incomplete	The subject-message was undeliverable because the specified recipient postal-OR-address was incompletely specified.
35	Undeliverable-mail-recipient-unknown	The subject-message was undeliverable because the recipient specified in the recipient postal-OR-address was not known at that address.
36	Undeliverable-mail-recipient-deceased	The subject-message was undeliverable because the recipient specified in the recipient postal-OR-address is deceased.
37	Undeliverable-mail-organisation-expired	The subject-message was undeliverable because the recipient organization specified in the recipient postal-OR-address has expired.
38	Undeliverable-mail-recipient-refused-to-accept	The subject-message was undeliverable because the recipient specified in the recipient postal-OR-address refused to accept it.
39	Undeliverable-mail-recipient-did-not-claim	The subject-message was undeliverable because the recipient specified in the recipient postal-OR-address did not collect the mail.
40	Undeliverable-mail-recipient-changed-address-permanently	The subject-message was undeliverable because the recipient specified in the recipient postal-OR-address has changed address permanently (Tmoved'), and forwarding was not applicable.



41	Undeliverable-mail-recipient-changed-address-temporarily	The subject-message was undeliverable because the recipient specified in the recipient postal-OR-address has changed address temporarily (T on travel'), and forwarding was not applicable.
42	Undeliverable-mail-recipient-changed-temporary-address	The subject-message was undeliverable because the recipient specified in the recipient postal-OR-address had changed temporary address (Tdeparted'), and forwarding was not applicable.
43	Undeliverable-mail-new-address-unknown	The subject-message was undeliverable because the recipient has moved, and the recipient's new address is unknown.
44	Undeliverable-mail-recipient-did-not-want-forwarding	The subject-message was undeliverable because delivery would have required physical-forwarding which the recipient did not want.
45	Undeliverable-mail-originator-prohibited-forwarding	The physical forwarding required for the subject-message to be delivered has been prohibited by the originator of the subject-message.
46	Secure-messaging-error	The subject could not be progressed because it would violate the security-policy in force.
47	Unable-to-downgrade	The subject could not be transferred because it could not be downgraded (see Annex B to Recommendation X.419).
48	Unable-to-complete-transfer	Delivery failed (e.g., size of message exceeds limit)
49	Transfer-attempts-limit-reached	Maximum number of attempts to establish a connection for message transfer reached
50	Incorrect-notification-type	The report type defined in message is not corresponding to content of message
51	DI-expansion-prohibited-by-security-policy	The subject-message was addressed to a Distribution List, but the security policy prohibited expansion of that DL

52	Forbidden-alternate-recipient	The subject-message would have been redirected, but the new recipient is unacceptable for security reasons
53	Security-policy-violation	The security-policy is violated
54	Security-services-refusal	The security services requested cannot be supported
55	Unauthorised-dl-member	The DL-expansion was not performed because the MTA discovered that one of the members of the Distribution List was prohibited by the security policy from receiving this message
56	Unauthorised-dl-name	The MTA has detected that the recipient OR-name identifies a Distribution List, but the local security policy does not permit the onward transfer towards the DL-expansion point
57	Unauthorised-originally-intended-recipient-name	The OR-name of the originally intended recipient of the redirected or DL-expanded message is unauthorised for security reasons
58	Unauthorised-originator-name	The originator MTS-user OR-name is unauthorised for security reasons
59	Unauthorised-recipient-name	The recipient MTS-user OR-name is unauthorised for security reasons
60	Unreliable-system	Delivery of the subject-message would require that the subject-message be transferred to an insecure system, which is incompatible with the message security label
61	Authentication-failure-on-subject-message	Validation of the content-integrity-check, message-originauthentication-check, or message-token (e.g., signature, or any other token data) argument of the subject-message failed, and therefore the contents of the subject-message could not be authenticated or validated
62	Decryption-failed	The subject-message content could not be decrypted

63	Decryption-key-unobtainable	The required key could not be obtained to decrypt the message-token encrypted-data or for content confidentiality
64	Double-envelope-creation-failure	The security policy required the creation of an outer envelope to protect the subject-message. However, the MTA was unable to create the outer envelope
65	Double-enveloping-message-restoring-failure	The subject-message contained an inner envelope, but failure of security services on the outer envelope prevented the MTA from extracting the inner message for subsequent processing
66	Failure-of-proof-of-message	A fault was detected in the proof of security arguments in the subject-message;
67	Integrity-failure-on-subject-message	Validation of the content-integrity-check argument of the subject-message failed, and therefore the contents of the subject-message could not be validated
68	Invalid-security-label	The security policy identifier in the message security label identifies a policy which is known to the recipient UA or MTA, but which is not acceptable to that system
69	Key-failure	The required keys could not be obtained
70	Mandatory-parameter-absence	A mandated security element for compliance with the security-policy in force is absent
71	Operation-security-failure	The transfer or delivery operation failed for security reasons
72	Repudiation-failure-of-message	The security policy required use of a signature with non-repudiation properties, but the subject-message was not signed with a non-repudiable signature on origination
73	Security-context-failure	The message security label is incompatible with the security-context in force
74	Token-decryption-failed	The message token could not be decrypted

75	Token-error	An error has been detected with the message-token argument of the subject-message
76	Unknown-security-label	The security policy identifier in the message security label is not recognised by the recipient UA or MTA. Such a policy is not supported by that system
77	Unsupported-algorithm-identifier	The recipient does not support the algorithm identifiers used in the security argument of the subject-message
78	Unsupported-security-policy	The recipient does not support the required security-policy, as identified in the message-security-label argument of subject-message

## B4. X.400 User Agent Error codes (Non Receipt Notification)

### Error codes in NRN:

Code number	X.400 Standard	Description
0	IPM-discarded	Message was discarded
1	IPM-auto-forwarded	Message was auto forwarded and there is no guarantee for the processing of this message

### Reason to discard messages:

Code number	X.400 Standard	Description
0	IPM-expired	Message has expired
1	IPM-obsolete	Message is not valid
2	User-subscription-terminated	User agreement is not longer valid

## B5. Mapping rules NDN to DSN

X.400 Reason code	X.400 Diagnostic Code	X.400 Standard	SMTP Action
6	0	Unrecognized-OR-name	5.1.1
6	1	Ambiguous-OR-name	5.1.4
0	2	MTS-congestion	5.4.5
0	3	Loop-detected	5.4.6.
0	4	Recipient-unavailable	5.2.1
0	5	Maximum-time-expired	5.4.7
2	6	Encoded-Information-Types-unsupported	5.6.1
1	7	Content-too-long	5.3.4
2	8	Conversion-impractical	5.6.3
2	9	Implicit-conversion-prohibited	5.6.2
2	10	Implicit-conversion-not-subscribed	5.6.5
1	11	Invalid-arguments	5.5.4
1	12	Content-Syntax-error	5.5.2
1	13	Size-constraint-violation	5.5.0
1	14	Protocol-violation	5.5.0
1	15	Content-type-not-supported	5.6.1
1	16	Too-many-recipients	5.5.3
1	17	No-bilateral-agreement	5.7.1
1	18	Unsupported-critical-function	5.5.1
2	19	Conversion-with-loss-prohibited	5.6.5
2	20	Line-too-long	5.6.5
2	21	Page-split	5.6.5
2	22	Pictorial-symbol-loss	5.6.5
2	23	Punctuation-symbol-loss	5.6.5
2	24	Alphabetic-character-loss	5.6.5
2	25	Multiple-information-loss	5.6.5
1	26	Recipient-reassignment-prohibited	5.4.0
1	27	Redirection-loop-prohibited	5.4.0
1	28	DI-expansion-prohibited	5.7.2
1	29	No-di-submit-permission	5.7.2
1	30	DI-expansion-failure	5.2.4
3	31	Physical-rendition-attributes-not-supported	5.0.0

4	32	Undeliverable-mail-physical-delivery-address-incorrect	5.0.0
4	33	Undeliverable-mail-physical-delivery-office-incorrect-or-invalid	5.0.0
4	34	Undeliverable-mail-physical-delivery-address-incomplete	5.0.0
4	35	Undeliverable-mail-recipient-unknown	5.0.0
4	36	Undeliverable-mail-recipient-deceased	5.0.0
4	37	Undeliverable-mail-organisation-expired	5.0.0
4	38	Undeliverable-mail-recipient-refused-to-accept	5.0.0
4	39	Undeliverable-mail-recipient-did-not-claim	5.0.0
4	40	Undeliverable-mail-recipient-changed-address-permanently	5.0.0
4	41	Undeliverable-mail-recipient-changed-address-temporarily	5.0.0
4	42	Undeliverable-mail-recipient-changed-temporary-address	5.0.0
4	43	Undeliverable-mail-new-address-unknown	5.0.0
4	44	Undeliverable-mail-recipient-did-not-want-forwarding	5.0.0
4	45	Undeliverable-mail-originator-prohibited-forwarding	5.0.0
8	46	Secure-messaging-error	5.7.0
1	47	Unable-to-downgrade	5.6.5
0	48	Unable-to-complete-transfer	5.5.0
0	49	Transfer-attempts-limit-reached	5.4.0
1	50	Incorrect-notification-type	5.6.0
8	51	DI-expansion-prohibited-by-security-policy	5.7.2
8	52	Forbidden-alternate-recipient	5.7.0
8	53	Security-policy-violation	5.7.0
8	54	Security-services-refusal	5.7.0
8	55	Unauthorised-dl-member	5.7.0
8	56	Unauthorised-dl-name	5.7.2
8	57	Unauthorised-originally-intended-recipient-name	5.7.0
8	58	Unauthorised-originator-name	5.7.1
8	59	Unauthorised-recipient-name	5.7.0

8	60	Unreliable-system	5.7.3
8	61	Authentication-failure-on-subject-message	5.7.7
8	62	Decryption-failed	5.7.5
8	63	Decryption-key-unobtainable	5.7.5
8	64	Double-envelope-creation-failure	5.7.0
8	65	Double-enveloping-message-restoring-failure	5.7.0
8	66	Failure-of-proof-of-message	5.7.0
8	67	Integrity-failure-on-subject-message	5.7.0
8	68	Invalid-security-label	5.7.0
8	69	Key-failure	5.7.5
8	70	Mandatory-parameter-absence	5.7.0
8	71	Operation-security-failure	5.7.0
8	72	Repudiation-failure-of-message	5.7.0
8	73	Security-context-failure	5.7.0
8	74	Token-decryption-failed	5.7.5
8	75	Token-error	5.7.0
8	76	Unknown-security-label	5.7.0
8	77	Unsupported-algorithm-identifier	5.7.4
8	78	Unsupported-security-policy	5.7.4

## Appendix C: Examples for Messages and Reports

### C1. Delivered Message with text attachment

#### Delivered Message (M\_5K00AG0HBDM0F2F9.OUT)

To: "G=MG1;S=MGATE;CN=MG1 MGATE;O=TESTAG;A=VIAT;C=DE" <49603@viaT.de>  
 From: "G=ipm;S=tester;O=testag;A=viaT;C=de" <49637@viaT.de>  
 Message-ID: 615 10/11/13  
 X-MPDUID: 3D23437A11DCEC31170084BF  
 Date: 13 Nov 2010 13:10:22 +0100  
 Subject: Test mit Textbodypart  
 Disposition-Notification-To: "G=ipm;S=tester;O=testag;A=viaT;C=de"  
 MIME-Version: 1.0  
 Content-Type: text/plain  
 Content-Transfer-Encoding: 8bit

Test  
 äöüÄÖÜß1234567890123456789012345678901234567890123456789012345678901  
 234567890123456789012345678901234567890123456789012345678901234567890  
 123456789012345678901234567890123456789012345678901234567890123456789  
 01234567890123456789012345678901234567890

For this message (filename M\_5K00AG0HBDM0F2F9.OUT) a receipt notification is requested but not sent (see following examples for Reports)

### C2. Delivered Message with binary attachment

#### Delivered Message (M\_5K00AG0HBDM0F2FA.OUT)

To: "G=MG1;S=MGATE;CN=MG1 MGATE;O=TESTAG;P=MGATE;A=VIAT;C=DE" <49603@viaT.de>  
 From: "G=ipm;S=tester;O=testag;A=viaT;C=de" <49637@viaT.de>  
 Message-ID: 616 10/11/13  
 X-MPDUID: 575BCBFB11DCEB9F1700C184  
 Date: 13 Nov 2010 13:10:22 +0100  
 Subject: Test mit Binäranhang  
 Disposition-Notification-To: "G=ipm;S=tester;O=testag;A=viaT;C=de"  
 MIME-Version: 1.0  
 Content-Type: application/octet-stream  
 Content-Disposition: attachment; filename="4d654d1d.zip"  
 Content-Transfer-Encoding: binary

PK            tYr2ÄQa6÷    <  
 .  
 .  
 .  
 ¶ 4d654d1d.0PK            8

For this message (filename M\_5K00AG0HBDM0F2FA.OUT) a receipt notification is requested, and a Non-Receipt Notification was sent (see following examples for Reports)



## C3. Delivered Message with Multiple attachments

### Delivered Message (M\_5K00AG0HBDM0F2F8.OUT)

To: "G=MG1;S=MGATE;CN=MG1 MGATE;O=TESTAG;P=MGATE;A=VIAT;C=DE" <49603@viaT.de>  
 From: "G=ipm;S=tester;O=testag;A=viaT;C=de" <49637@viaT.de>  
 Message-ID: 614 10/11/13  
 X-MPDUID: 8B0663A011DCEC4417009682  
 Date: 13 Nov 2010 13:10:22 +0100  
 Subject: Test mit 3 Bodyparts  
 Disposition-Notification-To: "G=ipm;S=tester;O=testag;A=viaT;C=de"  
 MIME-Version: 1.0  
 Content-Type: multipart/mixed; boundary="MG\_=\_CA610D0211DC91E900007CAD\_=\_MG"

--MG\_=\_CA610D0211DC91E900007CAD\_=\_MG  
 Content-Type: text/plain  
 Content-Transfer-Encoding: 8bit

Test äöüÄÖÜß

--MG\_=\_CA610D0211DC91E900007CAD\_=\_MG  
 Content-Type: application/octet-stream  
 Content-Disposition: attachment; filename="4d654d1d.zip"  
 Content-Transfer-Encoding: binary

PK tYr2ÄQa6÷ < 4d654d1d.0µ"ls£: ...÷Tñ"ê¥Öi3xJU/\$!;DØb2xgd °1Ø

.

• 4d654d1d.0PK 8

--MG\_=\_CA610D0211DC91E900007CAD\_=\_MG  
 Content-Type: application/octet-stream  
 Content-Transfer-Encoding: binary

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from url=(0062)https://securep7.viat-test.de/~0000001045/result/fetch_all.RES -->
<HTML><HEAD>
<META http-equiv=Content-Type content="text/html; charset=windows-1252">
<META content="MSHTML 5.50.4930.1700" name=GENERATOR></HEAD>
<BODY><XMP>LOGIN::
.
.
.
</XMP></BODY></HTML>
```

--MG\_=\_CA610D0211DC91E900007CAD\_=\_MG—

For this message (filename M\_5K00AG0HBDM0F2F8.OUT) a receipt notification is requested, and a Receipt Notification was sent (see following examples for Reports).

## C4. Delivered Message with Multi-Recipients

To: "G=edi;S=tester;O=testag;A=viat;C=de" <49638@viaT.de>  
 To: "G=ipm;S=tester;CN=ipm tester;O=TESTAG;A=viat;C=DE" <49637@viaT.de>  
 Cc: "G=ipm;S=testmiv;O=testag;A=viat;C=de" <23998@viaT.de>  
 Cc: "G=EDI;S=TESTMIV;CN=EDI TESTMIV;O=TESTAG;A=VIAT;C=DE" <23999@viaT.de>  
 Cc: "S=murxer;O=murx;A=viat;C=de" <X@viaT.de>  
 Bcc: "G=MG1;S=MGATE;CN=MG1 MGATE;O=TESTAG;P=MGATE;A=VIAT;C=DE" <49603@viaT.de>  
 From: "G=MG1;S=MGATE;CN=MG1 MGATE;O=TESTAG;P=MGATE;A=VIAT;C=DE" <49603@viaT.de>  
 Message-ID: MGATE 0001 11/03

X-MPDUID: 5758CA1B11E0498E00005292  
 Date: 8 Mar 2011 14:14:12 +0100  
 Subject: test Multi Recipients  
 MIME-Version: 1.0  
 Content-Type: text/plain  
 Content-Transfer-Encoding: 8bit

test

The message has been sent to seven recipients and the own MessageGate account was addressed as a Blind carbon copy recipient.

## C5. Submitted Message and no Report Request

### Submitted Message (M\_Test\_3\_Body010.IN)

To: "" <49637@viat.de>  
 Subject: test 3 ohne Leerzeile Bodyparts  
 Message-Id: 260001 12/11/10 MGATE Test  
 Date: Tue, 12 Nov 2010 13:16:24 +0100  
 MIME-Version: 1.0  
 Content-Type: multipart/mixed;  
   boundary="-----=\_NextPart\_000\_0007\_01C7E331.7A0CA460"  
 Content-Transfer-Encoding: binary

-----=\_NextPart\_000\_0007\_01C7E331.7A0CA460  
 Content-Type: text/plain;  
   charset="iso-8859-1"  
 Content-Transfer-Encoding: 7bit

-----=\_NextPart\_000\_0007\_01C7E331.7A0CA460  
 Content-Type: application/x-pkcs12;  
   name="hpm-webdav.p12"  
 Content-Transfer-Encoding: base64  
 Content-Disposition: attachment;  
   filename="hpm-webdav.p12"

MIIKAQBAzCCCCcGCSqGSIb3DQEHAaCCCbGEGgm0MIIJsDCCBGcGCSqGSIb3DQEHBqCCBFgwgwRU  
 AgEAMIIETQYJKoZIhvcNAQcBMBwGCiqGSIb3DQEEMAQYwDgQlZnN9gMIQvocsCAggAgIIILQGaNZr  
 0IW6bN6jEdphtjnBzmCv8W9ipvE8wmpVxzUEwj5Mh226vHaBp2WtMBaHPSomsXFMPEJJj9JFnF2S  
 gPxZDVjUe5ImUB2EQDAopQEYLxJSX0YXh8uqnSD5Se4vuex+kunnb6o2nGXT8+Y9m3/uNCD9MEb6  
 CGIA0JExtmWQJXkDeHDZLjYiVCpcltNeMNC7EGH842jRGzS1umfOeSWb8+TcA2/uZtzaE9uIL7I  
 LfD7dfIJz/4uawC+LstCfO984pFKR8vOxKlAdbOn1CpuSIQFHHdgCZYVy1EHODllmQbml+bJ2Gwx  
 UPKDDuGdyK6G45JHZjuj4zDUSRXwfnrRmSUHAMZhUpRQwApPYyQo6zxhdd7NsdXpu7mDisNE/p6p  
 0DNPTf97j/AiPwVMEwz0nsfITqF+4LONXVKia7Mp8o7Zzm5XpwJ0/LP+47/+ZyCaClqB/qYtGlb  
 xlgI04DFbS6xaoUu7iNh7ZSqnXNMRJREtBx/WVoMChpYHuvVqitPWdsBpawNpUHS5uEXUopa0Uly  
 XOn9ALfLE0t9v5FP4NE3xSHMPGSAc5iisH7Fys8g5Z+SGp3n9ynM8Jw97JhZfjKoQMqrMFzNL5FI  
 ZUBVwNYOtUNXxKJ3L+1WtRXSEQgmfhptKZicCZKHoGZQ4Z8F4r9sA7wmS9CbLjINQlmWlrvaMWE3  
 fi6dzhrUOFIdu2LE7TI7+1Qmh/AcP3NVIUSUZIGJqqGc5I1BUmpMP3CJPo25xJ7zAek/YECJmQ5p9  
 l+c2Ja60suKAlt6VfBcd747nIEQXdxYvI8cXQeuzhVmvbBrX12Hg4ISioyGEg5XFsd4DutZAxTuW  
 gReDf8Hw/rMQfE6fhHilS7YirkqJt+q53uILMuN4sdV6u+nFsaoRYT84vTJZ30B5Wsh3Zs4T47r1  
 rTCn/BpQoQ8N62QF9zAPPL5AfctnDw/oZahJUqnQUNW7H86dLJ1ZkPJEICQ9quCSvjCMWZviliyr  
 lnyeW0JE53V5N/38me3xV89f6iUkNvWg3catzHTH5Bay1E1NGVi9cYfuNJ+qsHMxegcu5h9UGiVX  
 Z6AFQ5TOwPrObyOunVjUsGT8yllcpHEBwilPFP4GXq30gt3H7S2sDZSbrrDUYeWgJBgwmJaEjo/z  
 Pl67psBqnh4HKZoXAKSrfcF2JK2nt6q442tpIREVpkTXFGF6p7nqVvnP4RBD2LbFD/uzBxpchjR3  
 62I6LZ75qjSf4hZHnAVCD7BtfPx3jmg8fICp7ZyGRgSARpaLrYoMzbMXglPFYUOqf8rug/AoCqB  
 SD6OvMtvRfn5c3JceC9lZQ3/LGaqx7RGaUHYJaSXHPPrFCiozxtt2slw5nhWIFF1fgfJqVf2L41E2  
 8f7pRyHPejTBK1tozyHaWvsTm7kFm8FliDCCBUeGCSqGSIb3DQEHAaCCBTIEggUuMIIFFKjCCBSYG  
 CyqGSIb3DQEOMCGECollIE7jCCBOowHAYKKoZlHvcNAQwBAzAOBAiZFPkQkLYL2wICCAAEggTlcJfj  
 y/4rcNs13Bxzac5e9bbDPqW6l6Cng7jB6vXjSPBNMMLL+7BcVKeSIWupmsQeQkvZGhdcbY7Najsk  
 KE0EmaVVUPo1lgACKvZ2dc6nAVEeHbA14N1Zl2gCrvKZb1WHWj6NJ9e1xAKYzahVb5dkFNQIO8Y8  
 dQXgYhJF6davx+nFdhnoo8wnOA8ntwpJggGJAw5xM7GLIV2Xy0wahfoKG53Jxwgsz/OiLX/uh/Vk  
 c/kO+nKF4/au5igH9el8M6/I7A6kP854eXuMDWPXQHE35xAXrvt3gQd1D1n2wMGtRyCDA1h4mNr

Xwheql3nPScmwTRRsC6JSxZ1dx3kr4Zrw7yRR9HJT7oCn5VjonsdMfEATqY1GLDKOw4LE30Za3bW  
 kVSI3VzLxZx80WLcdiL9R1tn1FMbh/YJFs52OCF1MqnZdKq/fEP6yUK260PI6mZCS1FLTHI00vN3  
 +WPY8itoVb5qqPEHNCh1Li3mCHHv7hLS9t6p+JM4+/y2G6MD8pPp/dnUxSidpbglPV9/DQAMcx9P  
 PgHX01HnF8b8r61sKZX0KzizjdxTSdur/A5ZVDtZBzM2etFcEt17O08tko31UmAC/XWe9p2mjA1  
 ft2NAM3Yqyjf03zT7jw4uLskwcnipcd0snbviUY7prvn/7oBuTzklqmtvf9nfziyhNjByelytJyc  
 qqE8Q+MplrbWwUnQ2S1cg9zz38EVBIA6WGvYvgsKeDat1Ix1Zyol2InaCs4cXnZRR9HLZXL2hTW  
 aPZ7BVRtYohdme/18XtJgzySggdAMqxOG3I+JiqXXa4M3a38TrndEjNzY9pHLA6Pi1R3liBZ4ZiB  
 Ayo2Z42HiU83ZAsDxYTPbb2oYHgziJbvCQ7WomhefhTTV/q5S89FOEtIabYrjBdPliI+Q4GiPYin  
 Aw/BgDTHhKx5FXZz8L8WTTwnlvZ0OXq6tQ9ZfiNblZJYth/C/pjSf2kLUH/bj8X8RHeXv3DfaOkn  
 brqyx401gwIPx2JSqpcxX7kHroVxF+IYEhpfeEaTJ/650V3yUXmKAwL/CNxOhEqIj6QlpfPRAzb7  
 kgDFFX1Y/cVTKYQGs7tHmVFFiEvn93MVri0hIReDxTElhOQ6a/8D/laitiNO9nF4zATVl7nulxkg  
 l2NU3iJMqrXNlpgnh63Z8phS74Q6RW5O8DtVXFDBVNvIFiMUvBYxdt8bPLi/c9ZrXFjTr5ozSn2h  
 4WSKCiRpDfZGU+u9a4Tr+sU8GZyrDf8QnOB0sUo6aqF3Bjbb2jUHqRgVX1UgmeDGutZSW2qZY2DQ  
 HIJMC7E1BjmVsyPlodJLFRN8hBZCseJwuQ/6dtDoITsrbPtFTNN2kvtpgly6voQYolDWTyWxFCB8  
 NE3hyhkeTtVB92VQ3hxPGvgAp2ybolxKKnoBBrvDSpyawB/a629Op3a1NO82A6w6JwFVjOUvURSj  
 PovxBSACQtxr/dPAEuZIGNyftqHpWbO01CelSvKJ6VnoPYh6R2AJwWgDGPVqdBRuSVIWq4PDasTG  
 8yUCWqdfYFGsbbTMyDy0n5vzHmSlg1Z/3w7nU4ze+alRRB+xRjiBUBzi+An1qUCwHk9tMa9lqNwC  
 1wct6024js0/wocPpq7kVKBD2zf9Uy4KMSUwlvYJKoZlHvcNAQkVMRYEFA73TcSOycMboZppjFUR  
 siolKUFCMDEwITAJBgUrDgMCGGUABBTI7cq0AOvHFv4Aixdzm1d/1GaKNgQIUQPUR3fqiCICAggA

-----=\_NextPart\_000\_0007\_01C7E331.7A0CA460

Content-Type: application/octet-stream;  
 name="dtag-06.mod"  
 Content-Transfer-Encoding: 7bit  
 Content-Disposition: attachment;  
 filename="dtag-06.mod"

[Modul]

Name = "DTAG-06"

Bemerkung = "DTAG-Reservemodul"

Zielverzeichnis = "DTAG"

Ueberschreiben = 0

Delete = 1

Betreff\_auswerten = 1

CASE\_SENSITIVE = 1

Absender = "S=KUNDENBUCHHALTUNG;O=DTAG;A=viaT;C=DE"

EXAKT\_auswerten = 0

[Subject]

Start1 = 1

Text1 = "DTAG\_\_>>\$06"

[File]

[Message]

K1Type = Betreff

K1Start = 23

K1Laenge = 8

K2Type = Fest

K2Wert = ".Z"

K3Type = Betreff

K3Start = 32

K3Laenge = 2

[Text]

-----=\_NextPart\_000\_0007\_01C7E331.7A0CA460—

In this message no report was requested so the Status "Sent" will not change until the entry is purged in the database (see following examples for Reports).

## C6. Submitted Message with Report Request

### Submitted Message (M\_Test\_3\_Body011.IN)

To: "" <49637@viat.de>

Subject: test 3 Bodyparts

Message-Id: 260002 12/11/10 MGATE Test

Date: Tue, 12 Nov 2010 14:46:24 +0100

Disposition-Notification-To: ""

```

MIME-Version: 1.0
Content-Type: multipart/mixed;
  boundary="-----_NextPart_000_0007_01C7E331.7A0CA460"
Content-Transfer-Encoding: binary

-----_NextPart_000_0007_01C7E331.7A0CA460
Content-Type: text/plain;
  charset="iso-8859-1"
Content-Transfer-Encoding: 7bit

-----_NextPart_000_0007_01C7E331.7A0CA460
Content-Type: application/x-pkcs12;
  name="hpm-webdav.p12"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
  filename="hpm-webdav.p12"

MIIKAQIBAzCCCcGCSqGSIb3DQEHAaCCCBgEggm0MIIJsDCCBGcGCSqGSIb3DQEHBqCCBFgwgguRU
.
.
.
sioIKUFCMDEwITAjBgUrDgMCGgUABBTI7cq0AOvHFv4Aixdzm1d/1GaKNgQIUQPUR3fqiCICAggA

-----_NextPart_000_0007_01C7E331.7A0CA460
Content-Type: application/octet-stream;
.
.
.
[Text]

-----_NextPart_000_0007_01C7E331.7A0CA460—

```

In this message a report (it is not necessary to write an X.400 address between quotation marks) was requested and based on the parameter in the Profile (Value is 2) this request is mapped into an X.400 Receipt Notification request. The status of the message will change when the report arrives (see following examples for Reports).

## C7. Submitted Message with Multi-Recipients

```

To: " G=ipm;S=tester;O=testag;A=viaT;C=de " <x@viaT.de>
to: "" <41040@viat.de>
CC: "G=edi;S=tester;O=testag;A=viaT;C=de " <x@viaT.de>
cc: "" <31044@viat.de>
cc: " c=de; a=viat; o=unknown; S=dummy " <x@viat.de>
cc: "" <70000@viat.de>
BCC: "" <49603@viat.de>
Message-ID: MGATE 0001 11/03/07
Date: 07 March 2011 10:56:05 +0100
Subject: test Multi Recipients
Disposition-Notification-To: ""
MIME-Version: 1.0
Content-Type: text/plain
Content-Transfer-Encoding: 8bit

```

Test

The message has been sent to seven recipients and it is not relevant whether capital or small letters are used for the address type. The own MessageGate account was addressed as a Blind carbon copy recipient.

## C8. Delivered Signed Message

From: "G=ipm;S=testmiv;O=testag;A=viaT;C=de" <23998@viaT.de>  
 To: "G=MG1;S=MGATE;CN=MG1 MGATE;O=TESTAG;P=MGATE;A=VIAT;C=DE " <49603@viaT.de >  
 Subject: Send signed with SHA256 und Binary Encoding  
 Message-Id: 2222 15/11/09  
 X-MPDUID: E181D4D911E586D985D4F5A1  
 Content-Type: multipart/signed; protocol="application/pkcs7-signature"; micalg="sha256"; boundary="-----3D000600EB712D0BEB61770F44A3E1D3"

This is an S/MIME signed message

-----3D000600EB712D0BEB61770F44A3E1D3  
 Content-Type: multipart/mixed; boundary="MyThi80Jh0IKFIrPNglzJ0GUk60HwJ4I"  
 Content-Transfer-Encoding: binary

This a multi-part message in MIME format.  
 --MyThi80Jh0IKFIrPNglzJ0GUk60HwJ4I  
 Content-Type: text/plain; charset=ISO-8859-1  
 Content-Transfer-Encoding: binary

test

--MyThi80Jh0IKFIrPNglzJ0GUk60HwJ4I  
 Content-Type: application/octet-stream  
 Content-Transfer-Encoding: binary  
 Content-Disposition: attachment; filename="dnembbba.zip"; modification-date="Wed, 29 Apr 2015 13:19:05 +0100"

PK kxŠB£J ÷ \$ fld-0000\fld-00.fldc``d ,ÀÔd PK kxŠB£J ÷ \$ fld-0000\fld-01.fldc``d ,ÀÔd PK kxŠB£J ÷ \$ fld-0000\fld-02.fldc``d ,ÀÔd

.

.

.

lmsfld.txtPK - - P b

--MyThi80Jh0IKFIrPNglzJ0GUk60HwJ4I--

-----3D000600EB712D0BEB61770F44A3E1D3  
 Content-Type: application/pkcs7-signature; name="smime.p7s"  
 Content-Transfer-Encoding: binary  
 Content-Disposition: attachment; filename="smime.p7s"

0, q \*†H†÷  
 , b0, ^ 1

.

.

.

```

ñYdbÜ-. -~SH%6• µ¼°Ê• X°i$Ý·S ämn• -~"ë•oc¹u -{ U• • Ó• 1i£VxæF: ó• e¶¶"ŽJ~7"Öl's• Ž@µ-
ëÜŽ+Ç-'ÜvÄkÿÖ@i @Øú» : q%ˆÆ=fµµà°^â $k ÇwD] Gà,?• Ĩ~y°ŸH...<GÓYDh3~t°?Ü'PŽl@D}\$;%nžÓ~x
—¶¼[ĭ>°°g÷• Úòã½\Ú ĭ)v=ðfœt ç%• o WWu¶¶€'Æ;ÖÖ...İß9-=Ü~ X8Š Ń¿C:ê^L±_p Xî[¶—•XWâØÂB€9
-----3D000600EB712D0BEB61770F44A3E1D3--

```

The content of the Message has been signed by the Originator (P7 User Agent) and as the default MIME content settings Binary(8bit) has been chosen, the same Content-Transfer-Content for all the wrapped MIME body parts will be used. With signed messages the MessageGate Parameter „Encode Binary data" will be ignored, as a change of the Content-Transfer-Encoding e.g., from binary to Base64 would invalidate the Signature.

If your application requires 7bit encoding, then you will have to ask your partner to use this encoding for the user data before signing the content of the X.400 message. Depending on the needs of your partner you will have to decide whether to use 7 or 8bit Encoding when sending signed content. If your partner is using FileWork or UA FI you must add the certificate of your application to the signature, otherwise those clients will be not able to process the signed user data.

## C9. Delivered Encrypted Message

```

From: "G=ipm;S=tester;O=testag;A=viaT;C=de" <49637@viaT.de @viaT.de>
To: "G=MG1;S=MGATE;CN=MG1 MGATE;O=TESTAG;P=MGATE;A=VIAT;C=DE " <49603@viaT.de >
Subject: Send signed with SHA256 and encrypted with AES256 and Binary Encoding
Message-ID: 2223 15/11/09
X-MPDUID: E1D6366911E586D985D414A2
Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name="smime.p7m"
Content-Disposition: attachment; filename="smime.p7m"
Content-Transfer-Encoding: base64

MIAGCSqGSIb3DQEHA6CAMIACAQAxggGSMIIbjIBADB2MHAxCzAJBgNVBAYTAkRFMSYwJAYD
VQQKDB1EZ XV0c2NoZSBUZWxla29tIFRIY2huaWsgR21iSDEVMBMGA1UEAwwMQ0EgVFBNIggu
.
.
.
e541VL/izFyq4wbrx/5n4+Pjc+qG+zbrsk48Hsp88R0UmYm8j9X/PwtGnymy5VC4JZAEEEE4i
L0DmaHuHaSSbTEd0QP8AAAAAAAAAAAAADQo=

```

The originator had signed the user data using the SHA256 hash algorithm and subsequently encrypted the content using AES256, hence it is not possible to see that the content has been signed. As a change of the Content-Transfer-Encoding of the encrypted content has no influence on the signed data, MessageGate uses the parameter "Encode binary data" to determine whether to use "Binary" or "Base64" for the message stored in the directory. When sending encrypted content, the MessageGate process will always convert the Content-Transfer-Encoding "Base64" to "Binary".

## C10. Transmission Set with two Interchanges

### Submitted Transmission Set file T\_TestEDI\_018.IN

UNA:+.? '

UNB+UNOA:2+MGATE1:65+TESTER:65+020508:1413+0709210008'

UNH+EVA0000001+INVOIC:D:95A:UN:ETEIB++0'

BGM++D--01/333700001003'

DTM+003:20080729'

DTM+263:9512:609'

NAD+II+++NL 2 STUTTGART-NORD+10 02 00+STUTTGART-NORD++70191+IC'

COM+0711/555-5002:TE'

COM+0711/555-5555:FX'

NAD+IV++TBX::FGNR 10110::93606 TESTHEIM'

CUX+1:DEM'

LIN+1+++333700001003:ISN:DT6:DTC++0'

LIN+2+++1:1+1'

MOA+203:0.2086'

LOC+1+33XXX:::TESTUNION'

QTY+107:2'

DTM+163:20080619090423:204'

DTM+048:131:807'

LIN+3+++1:1+1'

MOA+203:0.3129'

LOC+1+31XXX:::TESTUNION'

QTY+107:3'

DTM+163:20080626091536:204'

DTM+048:192:807'

LIN+4+++1:1+1'

MOA+203:0.1043'

LOC+1+9193XXX:::TESTUNION'

QTY+107:1'

DTM+163:20080711080945:204'

DTM+048:51:807'

LIN+5+++1:1+1'

MOA+203:0.1043'

LOC+1+9193XXX:::TESTUNION'

QTY+107:1'

DTM+163:20080711095040:204'

DTM+048:27:807'

UNS+S'

MOA+079:0.7301'

UNT+37+EVA0000001'

UNZ+1+0709210008'

UNA:+.? '

UNB+UNOA:2+MGATE1:65+TESTER:65+020508:1413+0709210009'

UNH+EVA0000001+INVOIC:D:95A:UN:ETEIB++0'

BGM++D--01/333700001003'

DTM+003:20080729'

DTM+263:9512:609'

NAD+II+++NL 2 STUTTGART-NORD+10 02 00+STUTTGART-NORD++70193+IC'  
 COM+0711/555-5002:TE'  
 COM+0711/555-5555:FX'  
 NAD+IV++TBX::FGNR 10110::93606 TESTHEIM'  
 CUX+1:DEM'  
 LIN+1+++333700001003:ISN:DT6:DTC++0'  
 LIN+2+++1:1+1'  
 MOA+203:0.2086'  
 LOC+1+33XXX:::TESTUNION'  
 QTY+107:2'  
 DTM+163:20080619090423:204'  
 DTM+048:131:807'  
 LIN+3+++1:1+1'  
 MOA+203:0.3129'  
 LOC+1+31XXX:::TESTUNION'  
 QTY+107:3'  
 DTM+163:20080626091536:204'  
 DTM+048:192:807'  
 LIN+4+++1:1+1'  
 MOA+203:0.1043'  
 LOC+1+9193XXX:::TESTUNION'  
 QTY+107:1'  
 DTM+163:20080711080945:204'  
 DTM+048:51:807'  
 LIN+5+++1:1+1'  
 MOA+203:0.1043'  
 LOC+1+9193XXX:::TESTUNION'  
 QTY+107:1'  
 DTM+163:20080711095040:204'  
 DTM+048:27:807'  
 UNS+S'  
 MOA+079:0.7301'  
 UNT+37+EVA0000001'  
 UNZ+1+0709210009'

The file may contain an empty line to separate the Interchanges, but this is not necessary.

## C11. Status Report without History

### Request Status report (S\_\*.IN)

Since: 13-Nov-2010

Direction: both



**Status report (S\_\*.OUT)**

Status Report for UserID 49603; generated 13-NOV-2010 14:56:23  
 Filters: Disposition=All, Direction=Both, Format=Actual, Since=13-Nov-2010

From: " G=ipm;S=tester;O=testag;A=viaT;C=de " <49637@viaT.de>  
 Order-ID: 5K00AG0HBDM0F2F8  
 Message-ID: 614 10/11/13  
 MTS-ID: CA610D0211DC91E900007CAD  
 Status: Read  
 Date: 13-Nov-2010 14:01:18 +0100

From: " G=ipm;S=tester;O=testag;A=viaT;C=de " <49637@viaT.de>  
 Order-ID: 5K00AG0HBDM0F2F9  
 Message-ID: 615 10/11/13  
 MTS-ID: CA79C90011DC91E900007EAD  
 Status: Received  
 Date: 13-Nov-2010 13:10:23 +0100

From: "G=ipm;S=tester;O=testag;A=viaT;C=de " <49637@viaT.de>  
 Order-ID: 5K00AG0HBDM0F2FA  
 Message-ID: 616 10/11/13  
 MTS-ID: CA9FC7DB11DC91E900007EAD  
 Status: Denied: (Reason: 0, Diagnostic: 0))  
 Date: 13-Nov-2010 14:01:18 +0100

To: "" <49637@viaT.de>  
 Order-ID: Test\_3\_Body010  
 Message-ID: 260001 12/11/10 MGATE Test  
 MTS-ID: 71F6370611DC91EB0000DDAE  
 Status: Sent  
 Date: 13-Nov-2010 13:22:12 +0100

To: "c=de;a=viat;s=nicht\_vorhanden,O=testag" <x@viaT.de>  
 Order-ID: NDN001  
 Message-ID: MGATE 49603 00001 13112010  
 MTS-ID: MGate<5K00AG0HBDM208B4>  
 Status: Error: (Reason: 159416490, Diagnostic: 0)  
 Date: 13-Nov-2010 13:22:13 +0100

To: "c=de;a=viat;s=nicht-vorhanden,O=testag" <x@viaT.de>  
 Order-ID: NDN002  
 Message-ID: MGATE 49603 00002 13112010  
 MTS-ID: D1FC163311DC91F400007EBA  
 Status: Failed: (Reason: 6, Diagnostic: 0)  
 Date: 13-Nov-2010 14:29:25 +0100

To: "" >49637@viaT.de>  
 Order-ID: Test\_3\_Body011  
 Message-ID: 260002 12/11/10 MGATE Test  
 MTS-ID: 098FC66111DC91F80000A6BD  
 Status: Read  
 Date: 13-Nov-2010 14:54:00 +0100

This status report shows the status of different messages. To limit the number of entries in the report only those were selected which were sent or received since the 13<sup>th</sup> of November 2010.

The first message was received, a receipt report was requested and was sent → Status is Read

The second message was received but no receipt report was requested. The sender only receives a delivery report if he had requested it → Status is Received

The third message was received, a receipt report was requested, and a negative report was sent → Status is Denied

The fourth message was sent, and no report was requested. Hence the status remains "Sent" until the entry is purged. There will be no information if the delivery of the message had failed → Status is Sent

The fifth message was not sent because there is an invalid character in the address element (in "s=" the character "\_" is used) that cause a process error → Status is Error.

The sixth message was sent, and a report was requested. The message has an unknown "TO:" address and the MTA generated a Non-Delivery Notification → Status is Failed.

The seventh message was delivered to a partner and a receipt notification was sent back → Status is Read.

## C12. Status Report with History

### Request Status report (S\_\*.IN)

Format: History  
Direction: both

### Status report (S\_\*.OUT)

Status Report for UserID 49603; generated 13-NOV-2010 14:56:22  
Filters: Disposition=All, Direction=Both, Format=History

From: "G=ipm;S=tester;O=testag;A=viaT;C=de" <49637@viaT.de>  
Order-ID: 5K00AG0HBDM0F2F8  
Message-ID: 614 10/11/13  
MTS-ID: CA610D0211DC91E900007CAD  
Received: 13-Nov-2010 13:10:22 +0100  
Read: 13-Nov-2010 14:01:18 +0100

From: "G=ipm;S=tester;O=testag;A=viaT;C=de" <49637@viaT.de>  
Order-ID: 5K00AG0HBDM0F2F9  
Message-ID: 615 10/11/13  
MTS-ID: CA79C90011DC91E900007EAD  
Received: 13-Nov-2010 13:10:23 +0100

From: "G=ipm;S=tester;O=testag;A=viaT;C=de" <49637@viaT.de>  
Order-ID: 5K00AG0HBDM0F2FA  
Message-ID: 616 10/11/13  
MTS-ID: CA9FC7DB11DC91E900007EAD  
Received: 13-Nov-2010 13:10:23 +0100  
Denied: 13-Nov-2010 14:01:18 +0100 (Reason: 0, Diagnostic: 0)

To: "" <49637@viaT.de>  
Order-ID: Test\_3\_Body010  
Message-ID: 260001 12/11/10 MGATE Test  
MTS-ID: 71F6370611DC91EB0000DDAE  
Sent: 13-Nov-2010 13:22:12 +0100

To: "c=de;a=viat;s=nicht\_vorhanden,O=testag" <x@viaT.de>  
Order-ID: NDN001  
Message-ID: MGATE 49603 00001 13112010  
MTS-ID: MGate<5K00AG0HBDM208B4>  
Error: 13-Nov-2010 13:22:13 +0100 (Reason: 159416490, Diagnostic: 0)

To: "c=de;a=viat;s=nicht-vorhanden,O=testag" <x@viaT.de>  
Order-ID: NDN002  
Message-ID: MGATE 49603 00002 13112010  
MTS-ID: D1FC163311DC91F400007EBA  
Sent: 13-Nov-2010 14:29:19 +0100  
Failed: 13-Nov-2010 14:29:25 +0100 (Reason: 6, Diagnostic: 0)

To: "" <49637@viaT.de>  
 Order-ID: Test\_3\_Body011  
 Message-ID: 260002 12/11/10 MGATE Test  
 MTS-ID: 098FC66111DC91F80000A6BD  
 Sent: 13-Nov-2010 14:52:21 +0100  
 Delivered: 13-Nov-2010 14:52:27 +0100  
 Read: 13-Nov-2010 14:54:00 +0100

This status reports shows the status history of different messages:

The first message was received, and a receipt report was sent at 14:01

The second message was received at 13:10 but no receipt report was requested.  
 The sender only received a delivery report because he had requested it

The third message was received at 13:10, a receipt report was requested, and negative report was sent → Status is Denied

The fourth message was sent at 13:22 and no report was requested. Hence the status remains "Sent" until the entry is purged. There will be no information if the delivery of the message has failed.

The fifth message was not sent because there is an invalid character in the address element (in "s=" the character "\_" is used) that cause a process error → Status is Error.

The sixth message was sent at 14:29 and a report was requested. The message has an invalid "To:" address and the MTA generated a Non-Delivery Notification.

The seventh message was sent at 14:52, delivered to partner at 14:52 and a receipt notification was generated and sent back at 14:54.

## C13. Status Report for a selected Order-ID

### Request Status report (S\_\*.IN)

Order-ID: NDN002

### Status report (S\_\*.OUT)

Status Report for UserID 49603; generated 13-NOV-2010 14:57:22

Filters: Disposition=All, Direction=Sent, Format=Actual, Order-ID=NDN002

To: "c=de;a=viat;s=nicht-vorhanden,O=testag" <x@viaT.de>

Order-ID: NDN002

Message-ID: MGATE 49603 00002 13112010

MTS-ID: D1FC163311DC91F400007EBA

Status: Failed: (Reason: 6, Diagnostic: 0)

Date: 13-Nov-2010 14:29:25 +0100

This status reports shows the status of a message, which was selected using the Parameter Order-ID.

## C14. Status Report for a selected Message-ID

### Request Status report (S\_\*.IN)

Message-ID: 2600\*

### Status report (S\_\*.OUT)

Status Report for UserID 49603; generated 13-NOV-2010 14:58:21

Filters: Disposition=All, Direction=Sent, Format=Actual, Message-ID=2600\*

To: "" <49637@viaT.de>

Order-ID: Test\_3\_Body010

Message-ID: 260001 12/11/10 MGATE Test

MTS-ID: 71F6370611DC91EB0000DDAE

Status: Sent

Date: 13-Nov-2010 13:22:12 +0100

To: "" <49637@viaT.de>

Order-ID: Test\_3\_Body011

Message-ID: 260002 12/11/10 MGATE Test

MTS-ID: 098FC66111DC91F80000A6BD

Status: Read

Date: 13-Nov-2010 14:54:00 +0100

This status report shows the status of two messages, which were selected using the Parameter Message-ID and using a wildcard as part of this ID.

## C15. Status Report for denied Messages

### Request Status report (S\_\*.IN)

Format: History

Direction: both

### Status report (S\_\*.OUT)

Status Report for UserID 49603; generated 15-NOV-2010 16:21:04

Filters: Disposition=All, Direction=Both, Format=History

From: " G=ipm;S=tester;O=testag;A=viaT;C=de " <49637@viaT.de>

Order-ID: T730AG0HBFP1BBC0

Message-ID: 625 10/11/15

MTS-ID: 76CEBBE911DC93960000819A

Error: 15-Nov-2010 13:37:25 +0100 (Reason: 1, Diagnostic: 17)

From: " G=ipm;S=tester;O=testag;A=viaT;C=de " <49637@viaT.de>

Order-ID: T730AG0HBFP1BBC1

Message-ID: 626 07/11/15

MTS-ID: 7748EA6D11DC93960000889A

Error: 15-Nov-2010 15:11:57 +0100 (Reason: 1, Diagnostic: 11)

From: "G=ipm;S=tester;O=testag;A=viaT;C=de " <49637@viaT.de>  
 Order-ID: T730AG0HBFP1BBC3  
 Message-ID: 628 10/11/15  
 MTS-ID: 77AC72F911DC939600008A9A  
 Received: 15-Nov-2010 16:18:57 +0100

This status report shows the status of messages which are not delivered to a user's directory because of partnership errors (central EDI function).

In the first message a wrong recipient ID was found in the header of the EDIFACT interchange. The MTA has created a Non-Delivery Notification with a Diagnostic code "No-bilateral-agreement" (17) and has sent it to the sender of the message.

In the second message there is a wrong value in the UNZ of the EDIFACT interchange. The MTA has created a Non-Delivery Notification with a Diagnostic code "Invalid-arguments" (11) and has sent it to the sender of the message.

The third message was delivered to the directory at 16:18.

## C16. Report for submitted message (Multi-Recipients)

Status Report for UserID 49603; generated 8-Mar-2011 11:37:06 +0100  
 Filters: Disposition=All, Direction=Both, Format=History

To: "G=ipm;S=tester;O=testag;A=viaT;C=de" <x@viaT.de>  
 Order-ID: Test\_ISOTEXT\_M018  
 Message-ID: MGATE 0001 11/03/07  
 MTS-ID: A2CD418E11E048A90000D680  
 Sent: 7-Mar-2011 10:57:03 +0100  
 Delivered: 7-Mar-2011 11:39:44 +0100

To: "" <41040@viat.de>  
 Order-ID: Test\_ISOTEXT\_M018  
 Message-ID: MGATE 0001 11/03/07  
 MTS-ID: A2CD418E11E048A90000D680  
 Sent: 7-Mar-2011 10:57:03 +0100  
 Delivered: 7-Mar-2011 11:39:44 +0100

Cc: "G=edi;S=tester;O=testag;A=viaT;C=de" <x@viaT.de>  
 Order-ID: Test\_ISOTEXT\_M018  
 Message-ID: MGATE 0001 11/03/07  
 MTS-ID: A2CD418E11E048A90000D680  
 Sent: 7-Mar-2011 10:57:03 +0100  
 Delivered: 7-Mar-2011 11:39:44 +0100  
 Read: 7-Mar-2011 15:11:38 +0100

Cc: "" <31044@viat.de>  
Order-ID: Test\_ISOTEXT\_M018  
Message-ID: MGATE 0001 11/03/07  
MTS-ID: A2CD418E11E048A90000D680  
Sent: 7-Mar-2011 10:57:03 +0100  
Failed: 7-Mar-2011 10:57:03 +0100 (Reason: 1, Diagnostic: 11)

Cc: "c=de; a=viat; o=unknown; S=dummy" <x@viat.de>  
Order-ID: Test\_ISOTEXT\_M018  
Message-ID: MGATE 0001 11/03/07  
MTS-ID: A2CD418E11E048A90000D680  
Sent: 7-Mar-2011 10:57:03 +0100  
Failed: 7-Mar-2011 10:57:03 +0100 (Reason: 6, Diagnostic: 0)

Cc: "" <70000@viat.de>  
Order-ID: Test\_ISOTEXT\_M018  
Message-ID: MGATE 0001 11/03/07  
MTS-ID: A2CD418E11E048A90000D680  
Error: 7-Mar-2011 10:57:03 +0100 (Reason: 6, Diagnostic: 0)

Bcc: "" <49637@viat.de>  
Order-ID: Test\_ISOTEXT\_M018  
Message-ID: MGATE 0001 11/03/07  
MTS-ID: A2CD418E11E048A90000D680  
Sent: 7-Mar-2011 10:57:03 +0100  
Delivered: 7-Mar-2011 11:39:44 +0100

The Order-ID and the Message-ID are identical for all entries. Only the MTS-ID might be different because in case of an error the MessageGate process will create this ID and not the MTA. These entries are only unambiguous in conjunction with a specific recipient address. In this example all address types (To:, Cc: and Bcc.) were used and for some of these recipient addresses a Non Delivery Notification has been received.

The 4th recipient is an EDIBOX that supports only messages with one recipient and EDIFACT document → Error "Invalid arguments".

The 5th recipient has a wrong X.400 address → Error "Unknown User"

The 6th recipient has an invalid User-ID → Error "Unknown User"

For editorial reasons this page is empty!

## Appendix D: Character sets

### Printable String:

A, B...Z	Capital letter
a, b...z	Small letter
0, 1...9	Number
" "	Space
'	Apostrophe
(	Left Parenthesis
)	Right Parenthesis
+	Plus sign
-	Hyphen - Minus
,	Comma
.	Full stop
/	Solidus
:	Colon
=	Equal sign
?	Question mark

### ISO-Latin 1 (ISO 8859-1)

Decimal	Hexadecimal	Character
32	0x20	Space
33	0x21	!
34	0x22	"
35	0x23	#
36	0x24	\$
37	0x25	%
38	0x26	&
39	0x27	'
40	0x28	(
41	0x29	)
42	0x2A	*
43	0x2B	+
44	0x2C	,



45	0x2D	-
46	0x2E	.
47	0x2F	/
48	0x30	0
49	0x31	1
50	0x32	2
51	0x33	3
52	0x34	4
53	0x35	5
54	0x36	6
55	0x37	7
56	0x38	8
57	0x39	9
58	0x3A	:
59	0x3B	;
60	0x3C	<
61	0x3D	=
62	0x3E	>
63	0x3F	?
64	0x40	@
65	0x41	A
66	0x42	B
67	0x43	C
68	0x44	D
69	0x45	E

70	0x46	F
71	0x47	G
72	0x48	H
73	0x49	I
74	0x4A	J
75	0x4B	K
76	0x4C	L
77	0x4D	M
78	0x4E	N
79	0x4F	O
80	0x50	P
81	0x51	Q
82	0x52	R
83	0x53	S
84	0x54	T
85	0x55	U
86	0x56	V
87	0x57	W
88	0x58	X
89	0x59	Y
90	0x5A	Z
91	0x5B	[
92	0x5C	\
93	0x5D	]
94	0x5E	^

95	0x5F	—
96	0x60	`
97	0x61	a
98	0x62	b
99	0x63	c
100	0x64	d
101	0x65	e
102	0x66	f
103	0x67	g
104	0x68	h
105	0x69	i
106	0x6A	j
107	0x6B	k
108	0x6C	l
109	0x6D	m
110	0x6E	n
111	0x6F	o
112	0x70	p
113	0x71	q
114	0x72	r
115	0x73	s
116	0x74	t
117	0x75	u
118	0x76	v
119	0x77	w

120	0x78	x
121	0x79	y
122	0x7A	z
123	0x7B	{
124	0x7C	
125	0x7D	}
126	0x7E	~
161	0xA1	ı
162	0xA2	¢
163	0xA3	£
164	0xA4	¤
165	0xA5	¥
166	0xA6	¦
167	0xA7	§
168	0xA8	¨
169	0xA9	©
170	0xAA	ª
171	0xAB	«
172	0xAC	¬
173	0xAD	
174	0xAE	®
175	0xAF	¯
176	0xB0	°
177	0xB1	±
178	0xB2	²

179	0xB3	³
180	0xB4	´
181	0xB5	µ
182	0xB6	¶
183	0xB7	·
184	0xB8	¸
185	0xB9	¹
186	0xBA	º
187	0xBB	»
188	0xBC	¼
189	0xBD	½
190	0xBE	¾
191	0xBF	¿
192	0xC0	À
193	0xC1	Á
194	0xC2	Â
195	0xC3	Ã
196	0xC4	Ä
197	0xC5	Å
198	0xC6	Æ
199	0xC7	Ç
200	0xC8	È
201	0xC9	É
202	0xCA	Ê
203	0xCB	Ë

204	0xCC	ì
205	0xCD	í
206	0xCE	î
207	0xCF	ï
208	0xD0	Ð
209	0xD1	Ñ
210	0xD2	Ò
211	0xD3	Ó
212	0xD4	Ô
213	0xD5	Õ
214	0xD6	Ö
215	0xD7	×
216	0xD8	Ø
217	0xD9	Ù
218	0xDA	Ú
219	0xDB	Û
220	0xDC	Ü
221	0xDD	Ý
222	0xDE	Þ
223	0xDF	ß
224	0xE0	à
225	0xE1	á
226	0xE2	â
227	0xE3	ã
228	0xE4	ä

229	0xE5	å
230	0xE6	æ
231	0xE7	ç
232	0xE8	è
233	0xE9	é
234	0xEA	ê
235	0xEB	ë
236	0xEC	ì
237	0xED	í
238	0xEE	î
239	0xEF	ï
240	0xF0	ð
241	0xF1	ñ
242	0xF2	ò
243	0xF3	ó
244	0xF4	ô
245	0xF5	õ
246	0xF6	ö
247	0xF7	÷
248	0xF8	ø
249	0xF9	ù
250	0xFA	ú
251	0xFB	û
252	0xFC	ü
253	0xFD	ý

254	0xFE	þ
255	0xFF	ÿ