



# **B2B Gateway Synchrony EndPoint Activator Usage Requirements**

Date:	2012-11-06
Version:	0.9.0
Status:	
Reference:	Synchrony_EndPoint_Activator_Usage_Requirements_05.doc
Owner:	
Authors:	Ralf Mierbach

## **Table of Contents**

© Copyright by Telekom AG

Telekom Confidential and Proprietary

All rights reserved. No part of this document may be reproduced in any material form without the written permission of the copyright owner.

---

1	Introduction .....	3
2	Infrastructure Requirements.....	4
2.1	Communication with Partners.....	4
2.1.1	Using SMTP (not supported by Telekom ebXML gateway .....	4
2.1.2	Using HTTP(S) .....	5
3	Backend Integration .....	7
3.1	File I/O Interface .....	7
3.2	JMS-Interface.....	8

## Trademarks

All products and company names referred to in this documentation are the property of their respective trademark holders. A partial list follows.

Synchrony Gateway, Synchrony Gateway Interchange and Synchrony EndPoint Activator are registered trademarks of Axway Inc.

Microsoft Internet Explorer is a trademark of the Microsoft Corporation.

Windows 98, Windows Millennium Edition, Windows NT, Windows 2000 and Windows XP are trademarks of Microsoft Corporation.

# 1 Introduction

Telekom uses ebXML for communicating with there business partners.

ebXML (Electronic Business using eXtensible Markup Language), is a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet. Using ebXML, companies have a standard method to exchange business messages, conduct trading relationships and communicate data in common.

The ebXML suite of specifications is specified by OASIS, see <http://www.ebxml.org>).

The following ebXML standards provide the basis for Telekom's B2B communication:

- ebXML Collaboration-Protocol Profile and Agreement Specification 2.0
- ebXML Business Process Specification Schema 1.0
- ebXML Messaging Services Specification 2.0

ebXML envisages that both communication partners establish a transport connection to the other one when a message has to be transferred. In this sense ebXML does not follow the client / server paradigm, but is more oriented to a peer to peer model.

For implementing B2B communication Telekom builds on the Synchrony Products of Axway Inc, see <http://www.axway.com/>. While Telekom uses the *Synchrony Gateway Interchange* for this purpose, it recommends the use of Axway's *Synchrony EndPoint Activator* to its business partners. *Synchrony EndPoint Activator* provides a single connection that enables your company to easily and rapidly begin transacting over the Internet with a key trading partner. Any other ebXML conformant product implementing the ebXML standards listed above are also suited.

This document is based on the version of the *Axway Synchrony EndPoint Activator* 5.9 or higher.

This document discusses the preconditions and boundary conditions for using *Axway Synchrony EndPoint Activator* for this purpose.

## 2 Infrastructure Requirements

Synchrony EndPoint Activator is bundled with a java runtime environment and database software, thus the installation and operation of Synchrony EndPoint Activator does not require additional software components.

But the exchange of messages with the trading partner requires a certain communication infrastructure. Also, messages received must be sent to some backend application, where it is processed further, and messages which should be sent to the trading partner must be delivered to Synchrony EndPoint Activator first. This is called *backend integration*.

For all these 'channels' there are different possibilities, with different infrastructure requirements, which are discussed in the following.

### 2.1 Communication with Partners

In principal, there are two options for communicating with partners:

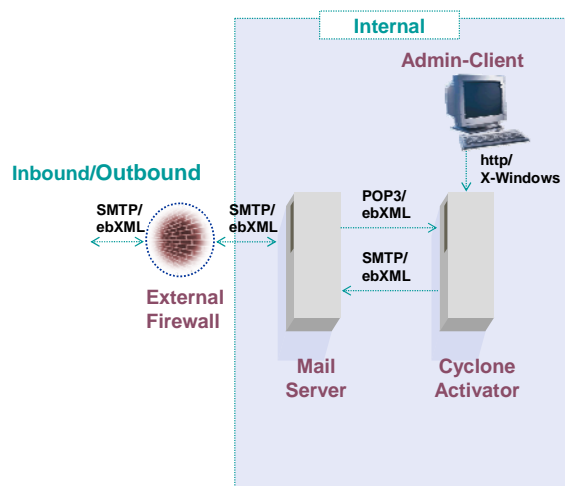
- Using SMTP (not supported by Telekom ebXML gateway)
- Using HTTP(S)

These infrastructure requirements depend on what communication protocol is selected for the partner communication. Both alternatives are considered in more detail in the following with respect to infrastructure requirements.

You have to implement the communication by HTTP(S).

#### 2.1.1 Using SMTP (not supported by Telekom ebXML gateway)

If you choose SMTP as messaging protocol, there must be SMTP server available which is capable of sending SMTP messages to your communication partner. The EndPoint Activator which you are going to install must be capable to access this SMTP server (usually on port 25) for sending outgoing mails.



You also need the infrastructure to access incoming messages. Again, there are two possibilities:

- You can use some POP server, which maintains an email post-box for you. Your partner addresses this post-box when sending messages to you. The EndPoint Activator which you are going to install must be capable to access this POP server (usually on port 110) for fetching incoming messages.
- Alternatively, you can configure your mail infrastructure in such a way, that e-mails to a given address are forwarded to an embedded SMTP Server which is itself part of EndPoint Activator.

The latter approach has the advantage that a message can be delivered just when it arrives. If you use a POP server, EndPoint Activator polls the mailbox at certain points in time. While you can configure the interval for polling, there is certain delay for delivering messages by default.

(Notice: This option is not supported by Telekom ebXML gateway)

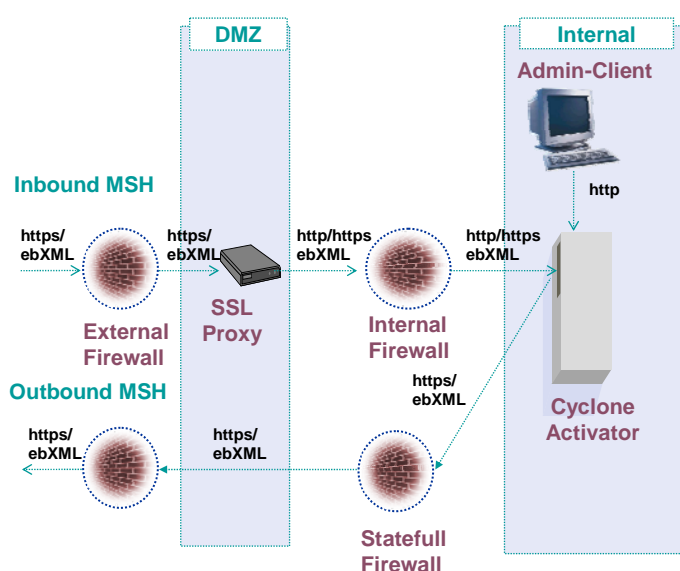
## 2.1.2 Using HTTP(S)

When choosing HTTP(S), you must be able to establish a network connection to your partner's HTTP(S) server and also your partner must be able to connect your HTTP(S) server. Your HTTP(S) server is part of your EndPoint Activator, thus it is an embedded HTTP server.

### 2.1.2.1 Outgoing connection

In general, for calling your partner's HTTP(S) server the system on which you install EndPoint Activator either needs direct access to the Internet or you use an HTTP proxy which provides this access. The latter is recommended for security reasons, and shown in picture xxx (Statefull Firewall = HTTP proxy). Use of a proxy server can be configured within EndPoint Activator. Of course, the EndPoint Activator which you are going to install must be capable to access this proxy.

### 2.1.2.2 Incoming connection



Beside establishing connectivity to your partner's system, enabling access to the embedded HTTP(S) server of your own EndPoint Activator usually is the more critical task. In general, this requires that your system is accessible from the Internet, more precisely, that port of your system on which the embedded HTTP(S) server of EndPoint Activator listens. This port is configurable; the default is 4080 if you use HTTP. If you choose to use HTTPS you may select the standard port 443 for that service, or any other, 4443 for example.

## 3 Backend Integration

### 3.1 File I/O Interface

Synchrony Endpoint Activator needs additional meta informations for sending the payload documents. If the backend integration is established as file I/O interface the backend system must support the Activator with that MMD-file

#### Activities:

- The backend system creates the payload document and additional the MMD file. The MMD MUST store in the defined "Intergration Pickup" directory. The location of the payload document is referenced inside the MMD-file
- Activator polls the "Intergration Pickup" directory and picks up the MMD-file
- Using the meta in formations the Activator accepts the payload, processes and sends it.
- The MMD file will delete in the "Intergration Pickup" directory automatically
- More information concerning backend integration see the official Admin Guide from AXWAY
- 

```
<?xml version="1.0" encoding="UTF-8"?>
<MessageMetadataDocument id="B2BTEO" protocol="ebXML" protocolVersion="2.0">
  <Metadata name="Service" type="string">KombiAuftrag</Metadata>
  <Metadata name="From" type="string">10050</Metadata>
  <Metadata name="Action">KombiAuftragAnfrage</Metadata>
  <Metadata name="FromRole">KombiAuftrag</Metadata>
  <Metadata name="ToRole">KombiAuftrag</Metadata>
  <Metadata name="ConversationId">10050_TF2_1300</Metadata>
  <Metadata name="To" type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns"> 341389328</Metadata>
  <MessagePayloads>
    <Payload id="Payload">
      <MimeTypeId>payload</MimeTypeId>
      <MimeTypeType>application/xml</MimeTypeType>
      <Location type="filePath">C:\B2B\Testfaelle\TF2\test.xml</Location>
    </Payload>
  </MessagePayloads>
</MessageMetadataDocument>
```

Figure 1: Content of a MMD-File

MMD Properties	Kardinalität	Bemerkung
Service	1	Name of service
Action	1	Name of action
CPAId	1	Id der CPA
From	1	ID of senders
FromRole	1	Role name of sender
To	1	ID of recipient
ToRole	1	Role name of recipient
ConversationId	1	Correlation of more than one message to one process
Location type	1 or more	Reference to the payload file(s)

Table 1: Properties of MMD file

## 3.2 JMS-Interface

The needed meta informations (see above) must provide inside the JMS properties if a JMS backend integration is established. The whole JMS message is composed of:

- JMS Header Properties(JMS HDR): Header Properties like the JMS Standard
- User defined properties(USER HDR): Usage for the AXWAY meta information (see also AXWAY Admin Manual).
- JMS Body: The payload is pasted into the JMS Body. Supported message types are
  - Byte Message
  - Text Message.

Das folgende Beispiel zeigt den Auszug einer JMS Payload Message:

```
-- JMS MESSAGE RECEIVED -----
JMS HDR timestamp:   Wed Jul 13 10:35:53 MEST 2005
JMS HDR message id:  ID:P<55078.1121243753114.0>
JMS HDR correlation id: 0
JMS HDR type:        null
JMS HDR delivery mode: 2
JMS HDR destination: ToCyclone
JMS HDR expiration:  0
JMS HDR priority:    4
JMS HDR redelivered: false
JMS HDR reply-to:    null
USER HDR Service:     1.0/KombiAuftrag.xml
USER HDR Action:      KombiAuftragAntwort
USER HDR ReceiverRoutingId: 10052
USER HDR FromRole:    TMDId-KombiAuftrag
USER HDR ToRole:      RVPartner-KombiAuftrag
USER HDR SenderRoutingIdType: urn:oasis:names:tc:ebxml-cppa:partyid-type:duns
USER HDR SenderRoutingId: 123456789
```



```

    USER HDR ServiceType:  string
    USER HDR ReceiverRoutingIdType: string
-- BEGIN MSG TEXT -----
<?xml version="1.0" encoding="UTF-8"?>
<test>
</test>
---END MSG TEXT -----

```

Figure 2: Extract of a JMS message

Folgende Tabelle listet die wichtigsten JMS Properties auf:

JMS Properties	Kardinalität	Bemerkung
Service	1	name of service
Action	1	name of action
SenderRoutingId	1	ID of sender
SenderRoutingIdType	1	Sender ID typ, e.g.: string, urn:oasis:names:tc:ebxml-cppa:partyid-type:duns
FromRole	1	Role name of sender
ReceiverRoutingId	1	ID of recipient
ReceiverRoutingIdType	1	Recipient ID typ, e.g.: string, urn:oasis:names:tc:ebxml-cppa:partyid-type:duns
ToRole	1	Role name of recipient
ConversationId	1	Correlation of more than one message to one process.

Table 2: Properties of the JMS properties

Known restriction: Using the JMS backend integration only ONE payload per message can send out!