



# PAPAGENO

XML Interface  
for  
Process Control  
Application

Version 5.9

Microsoft® and Windows® are registered trademarks of the Microsoft Corporation.  
The remaining hardware and software designations mentioned in this book are in most cases also registered trademarks and so are subjected to the statutory provisions.

VIPcom GmbH  
Ruedesheimer Str. 7  
80686 Munich  
Germany  
Phone: +49 89 54750-0  
Fax: +49 89 54750-200  
info@vipcomag.de  
www.vipcomag.de

The use, reproduction or distribution of the program is subject to the restrictions described in your contract with VIPcom GmbH

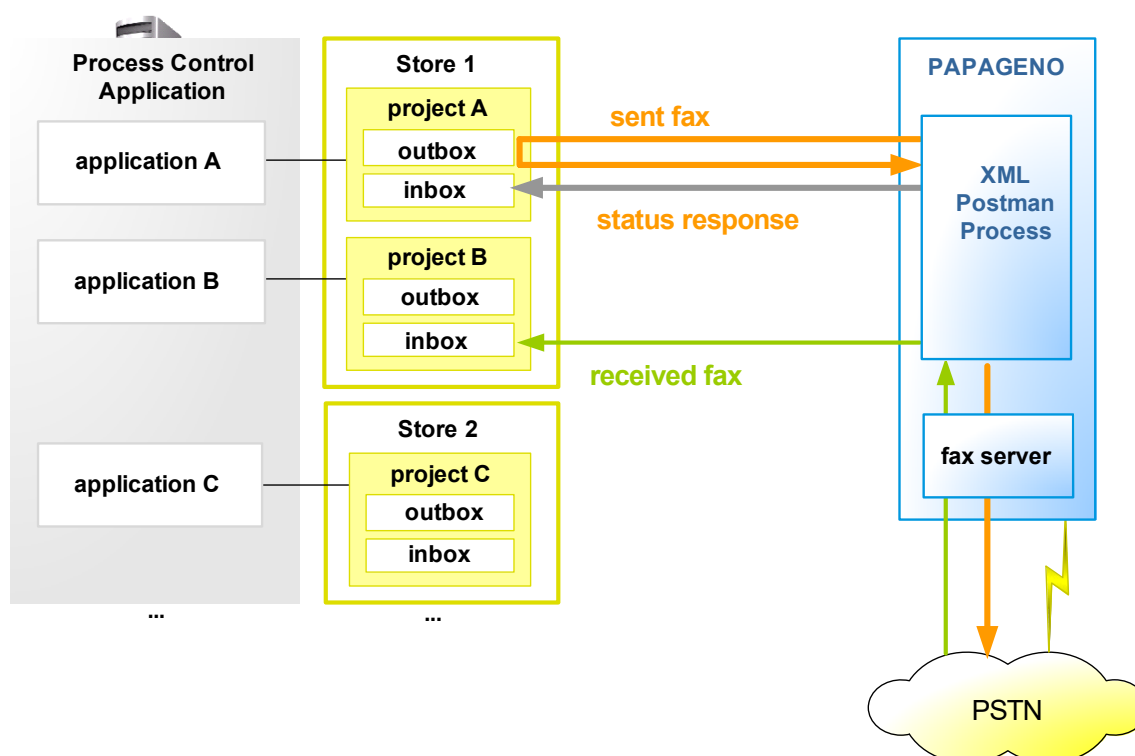
The specifications contained in this book are subject to correction and may be altered without any further notification.

VIPcom GmbH does not thereby commit itself to any further obligations.

# PAPAGENO XML Interface for Process Control Application



If a process control application software is used, the **PAPAGENO XML Interface** ensures a safe and simple connection to the PAPAGENO fax system.



The process control application caches **outbound fax requests** in the appropriate project store. The PAPAGENO XML Postman Process collects all these jobs and forwards them to the PAPAGENO Fax Server. It is also responsible to save the sent fax status (delivery status of a fax) in the appropriate inbox.

**Incoming faxes** are forwarded directly (via SFTP) to the inbox of the designated project (within the store). Every incoming fax consists of the document itself and a job control file with fax informations. These informations can be read out easily and then be evaluated by the application software.

**There are two ways to use the XML Interface:**

- You have already set up an **in-house** PAPAGENO installation or you are planning to do so

or

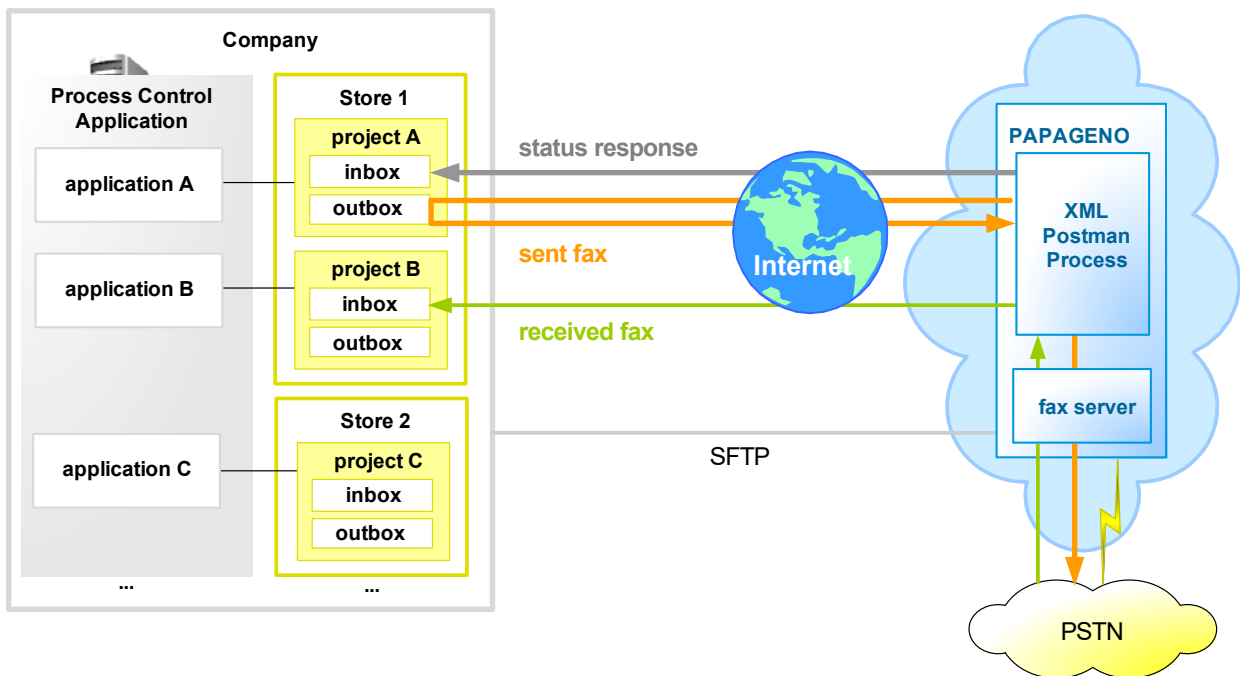
- you have access to a PAPAGENO installation located **in a data center**.

## In-house PAPAGENO Installation

- For incoming faxes you will need an LDAP Directory Server and a PAPAGENO SMTP-Gateway. (see chapter 2. "Establishing Stores and Projects", page 9).
- Establish the stores with your company projects (see chapter 2. "Establishing Stores and Projects", page 9).
- Configure the XMLPostman (see chapter 3. "Establishing XML Postman", page 17).

## PAPAGENO Installation in a Data Center

- You only have to establish the stores including your company projects (see chapter 2. "Establishing Stores and Projects", page 9).



The PAPAGENO XML Postman Process interacts with these stores via a safe SFTP connection.

# List of Contents

<b>1. Company PAPAGENO Installation - Prerequisites</b>	<b>5</b>
<b>2. Establishing Stores and Projects</b>	<b>5</b>
2.1. Creating Stores	6
2.2. Creating Projects	6
2.3. Outbound Fax Requests	7
Name Giving Syntax	10
Job control file (.xml)	11
To send more than one fax in one job file	12
Configure the Process Control Application	13
Flow of the Send Process	13
Status Response	14
2.4. Inbound Faxes	16
<b>3. Establishing XML Postman</b>	<b>17</b>
3.1. Overview	17
3.2. Enter the Fax Numbers into the LDAP Directory Server	18
3.3. Copying XML Postman Files	18
3.4. Publicize XML-Postman in PAPAGENO	19
Enter XML Postman User	19
Setting an OMEGA Configuration Variable	19
3.5. Establishing the conversion into the Fax Format	19
3.6. Creating xmlpostman.cfg Configuration File	20
3.7. Starting and Stopping the Postman via Commands	23
<b>4. Appendix</b>	<b>24</b>

4.1. xmlpostman.cfg Template .....24

4.2. Errorcode .....26

# 1. In-house PAPAGENO Installation - Prerequisites

## LDAP Directory Server

When receiving a fax the XML Postman checks the fax number and the appropriate project in the LDAP database. This means that fax number and project name must be entered in the LDAP database.

### If there is no LDAP Directory Server :

- ▶ Install it.

Details on all project relevant data you have to enter you will find in chapter 3.2. "Enter the Fax Numbers into the LDAP Directory Server", page 18.

## SMTP Gateway

The XML Postman uses the PAPAGENO SMTP Gateway native user variables.

### If not done yet

- ▶ Install the PAPAGENO S
- ▶
- ▶ MTP Gateway.
- ▶ Enter a gateway user in PAPAGENO.
- ▶ Set the gateway and the LDAP configuration variables.

See manual „PAPAGENO SMTP Gateway“, part B, chapter 1. "Note the Installation Prerequisites", page 21, to chapter 7. "Using LDAP", page 33.

### If you use Linux

- ▶ Ensure that the `expat` package is installed.





## 2. Establishing Stores and Projects

A process control application “project” consists of fe. g. “catering”, „marketing“, „bookings“,... .

### 2.1. Creating Stores

First create one or several stores. „Store“ means a safe location where all projects of a process control application are summarized. That can be a computer or a file system. The XML-Postman needs an absolute path to each store.

- ▶ Create the store(s).

### 2.2. Creating Projects

Each project gets a central folder and contains the subfolders “in” and “out”.

#### **Subfolder „in“**

for all incoming faxes and status responses.

#### **Subfolder „out“**

for all sent jobs.

#### **File structure of a store:**

```
.../ store/project1 /in/  
                        /out/  
    /projec2 /in/  
        /   out/  
    /projec3 /in/  
        /out/
```

#### **Example:**

```
xy/data/store/papageno/apps/bookings/in/  
                                bookings/out/  
xy/data/store/papageno/apps/catering/in/  
                                catering/out/  
xy/data/store/papageno/apps/marketing/in/  
                                marketing/out/
```

- ▶ Create the project(s) in a store - each with the subfolders „in“ and „out“.

## 2.3. Outbound Fax Requests

Outbound fax requests are stored in the project subfolder “out” by the process control application.

A fax request consists of two parts: the **document(s)** you want to send as a fax (e. g. .pdf) and the **corresponding job control file** (.xml)

The following will be described below:

- the **name-giving syntax** of the documents and the job control file
- the **structure of the job control file**
- the procedure of the **send process**
- the **status response**

### Name Giving Syntax

Fax files and job control file must have the same basic name - a unique numerical sequence:

```
jobname.xml  
jobname.pdf  
jobname002.pdf
```

If multiple documents are sent in one fax an additional number must be added to the filename (\_002 in the example above).

#### Example:

```
jobcateringB201207 00017.xml complies with all of the files  
jobcateringB201207-00017 *.* :  
jobcateringB201207-00017.pdf  
jobcateringB201207-00017_001.pdf  
jobcateringB201207-00017_002.pdf  
jobcateringB201207-00017.xml
```

**Attention!** You have to lowercase all file names!

### Supported Formats

The process control application can send .tif, .txt, .pcl, .ps and .pdf files as a fax.

## Job control file (.xml)

The job control file will not be completed until all information and document parts are in place. Only then it may get the suffix `.xml`.

Therefore the file can be saved with the suffix `.part` first and later on can be renamed into `.xml`.

### Content of a job control file:

Sample entries are cursiv.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE jobs>
<jobs>
  <fax>
    <fax-file-list>
      <fax-file>
        <file-name>jobuniquestring.pdf</file-name>
      </fax-file>
      <fax-file>
        <file-name>jobuniquestring _002.pdf</file-name>
      </fax-file>
    </fax-file-list>
    <distribution-list>
      <fax-address>
        <fax-number>00498954750200</fax-number>
      </fax-address >
      <fax-address>
        <fax-number>00498954750210</fax-number>
      </fax-address >
      <fax-address >
        <fax-number>+498954750220</fax-number>
      </fax- address >
    </distribution-list>
  </fax>
</jobs>
```

The **filename** may only contain the characters a-z, numeric characters 0-9 and “\_” (underscore).

Special characters are NOT allowed.

The fax number may contain “-” or “ ” (blanks). These characters will be skipped automatically.

### Additional Information Tags

It is possible to add information tags to the `.xml`-file. During the sending process the information in these tags will be ignored. But they remain in the file and will be added to the status response in the “in” file. Then they can be analyzed.

An example of an information tag is “customer id” or a special “order number”.

To name these tags only characters from a-z, numeric characters from 0-9 and “\_” are allowed.

## Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE jobs>
<jobs>
  <SpecIDNr>12345683902</SpecIDNr>
  <fax>
    <order number>08-13-F23</order number>
    <fax-file-list>
      <fax-file>
        <Typ>fax </Typ>
        <file-name>job7411.pdf</file-name>
      </fax-file>
      <fax-file>
        <file-name> job7411_002.pdf</file-name>
      </fax-file>
    </fax-file-list>
    <distribution-list>
      <fax-address>
        <customer id>Michael Schmitt</customer id>
        <customer id>S4711 0815 </customer id>
        <fax-number>00498954750200</fax-number>
      </fax-address>
      <fax-address>
        <fax-number>0049 64321 24567</fax-number>
      </fax-address>
      <fax-address>
        <fax-number>+49-89382-70 20690</fax-number>
      </fax-address>
    </distribution-list>
  </fax>
</jobs>
```

## To send more than one fax in one job file

It is possible to send out multiple faxes within one fax-job. The tag `<fax></fax>` marks these faxes.

### Example:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE jobs>
<jobs>
  <fax>
    first fax
  </fax>
  <fax>
    second fax
  </fax>
  <fax>
    etc...
  </fax>
</jobs>
```

## Configure the Process Control Application

- Program the process control application software so that outbound faxes will be moved to the current project `out`-folder and a job control file .

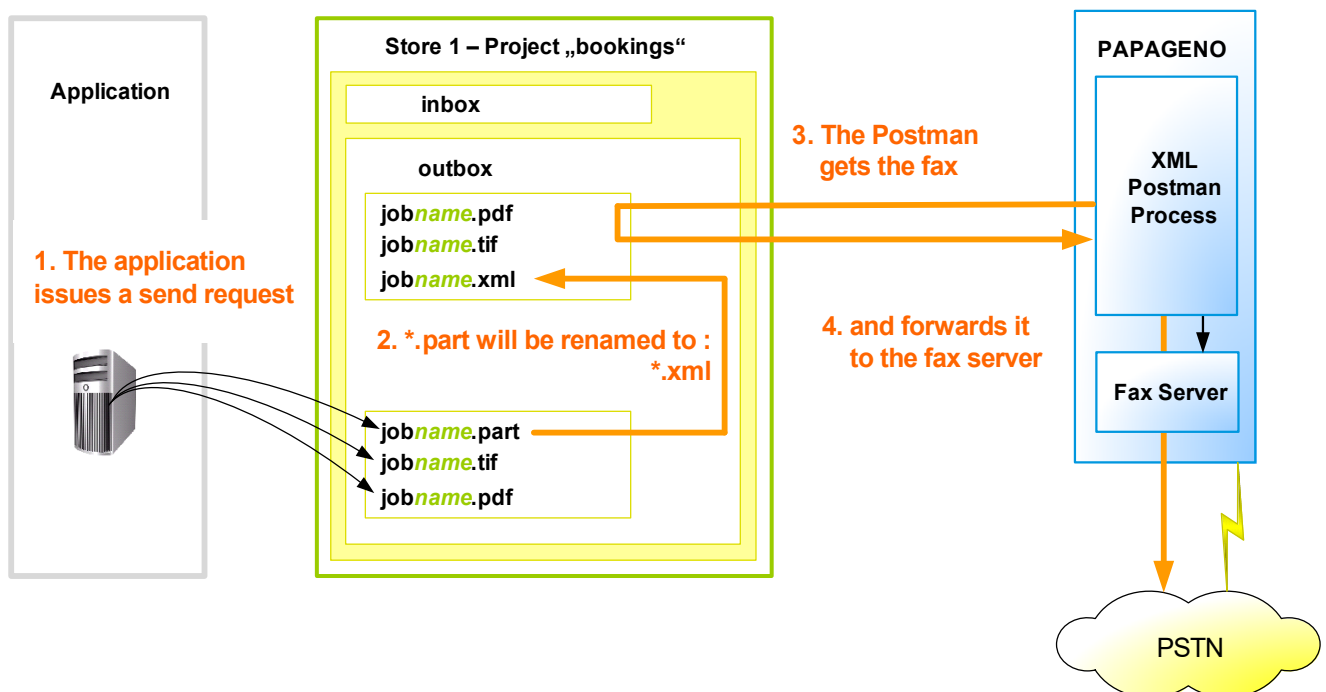
Pay attention to the syntax rules described above (see „Name Giving Syntax“, page 10).

**Note!** Lowercase all file names!

**Ensure** that the job control file first has the suffix `.part` and will be renamed in `...xml` after she is complete.

## Flow of the Send Process

With a single call the XML Postman process checks all outboxes in the store, gets all send jobs and transmits them to the PAPAGENO server.

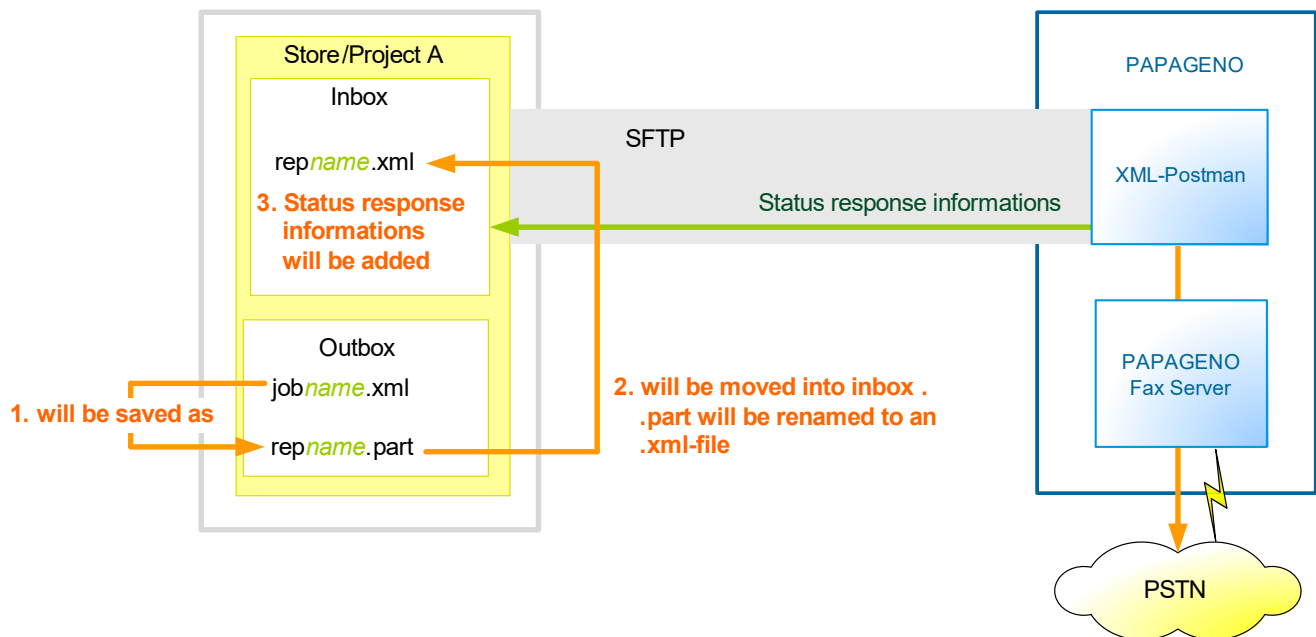


All files with the same basic name will be treated as one order.

Completed job orders will be deleted by the fax service.

## Status Response

After sending a fax the filename will be changed from “*jobuniquestring.xml*” to “*repuniquestring.part*” and moved into the „in“-folder. After the transfer is completed it will be renamed into an *.xml*-file.



The response information will be added to the file data. Not every status is shown at once, therefore the responses are written one after another. As soon as new information is available the corresponding file will be overwritten.

Initially the daemon adds a tag `<SendJobId>` for each fax address to the xml structure. This is the internal ID for the processing by PAPAGENO. When the send request was finished, it adds additionally the tags `<duration>`, `<status>`, `<timestamp>`, etc.

## Example of a *repname.xml* file

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE reports>
<jobs>
  <fax>
    <fax-file-list>
      <fax-file>
        <file-name>jobproj794311.pdf</file-name>
      </fax-file>
      <fax-file>
        <file-name>jobproj794311_001.pdf</file-name>
      </fax-file>
    </fax-file-list>

    <distribution-list>
      <SendJobId>59274547</SendJobId>
      <fax-address>
        <fax-number>+498954750200</fax-number>
        <status>OK</status>
      </fax-address>
    </distribution-list>
  </fax>
</jobs>
```

```

    <remote-csid>+49/89/547500400-1702</remote-csid>
    <duration>46</duration>
    <timestamp>31. March 2016 14:01:42</timestamp>
    <reason>this is a text</reason>
  </fax-address>
  <fax-address>
    <fax-number>+49 64321 24567</fax-number>
    <status>OK</status>
    <remote-csid>+49/89/1250400-1702</remote-csid>
    <duration>46</duration>
    <timestamp>31. March 2014 14:01:42</timestamp>
    <reason>This is the text</reason>
  </fax-address>
  <fax-address>
    <fax-number>089382-70 20690</fax-number>
    <status>no answer</status>
    <timestamp>31. March 2016 14:01:42</timestamp>
    <errorcode>47</errorcode>
    <reason>receiver not found</reason>
  </fax-address>
</distribution-list>
</fax>
</jobs>

```

**In fax reports the fax number will be extended to the following information:**

```

<status>*: oK or Errorcode
<reason>: if error-code in <status>, the description is added here**
    e. g. busy line, being transmitted, no_line
<remote-csid>: receivers number (CSI)
<duration>: duration of connection in seconds
<timestamp>: timestamp of sending time
<error code>: number of errorcode**

```

\* in any case the status will be sent

\*\* see list for all error messages on page 26.

## 2.4. Inbound Faxes

Incoming faxes are saved in the `in-` folder of a project as `inprojname.tif` or `inprojname.pdf` together with a control file.

### Example of an `inprojname.xml` file:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE inbound>
<reports>
  <fax>
    <fax-file>
      <file-name>inprojconfirmation1234.tif</file-name>
    </fax-file>
    <fax-address>
      <document-id>6543-21</document-id>
      <fax-number>0898954750200</fax-number>
      <status>OK</status>
      <remote-tsi>+49 89 54750200</remote-tsi>
      <duration>46</duration>
      <page-count>2</page-count>
      <timestamp>31. March 2014 14:01:42</timestamp>
      <error-message></error-message>
    </fax-address>
  </fax>
</reports>
```

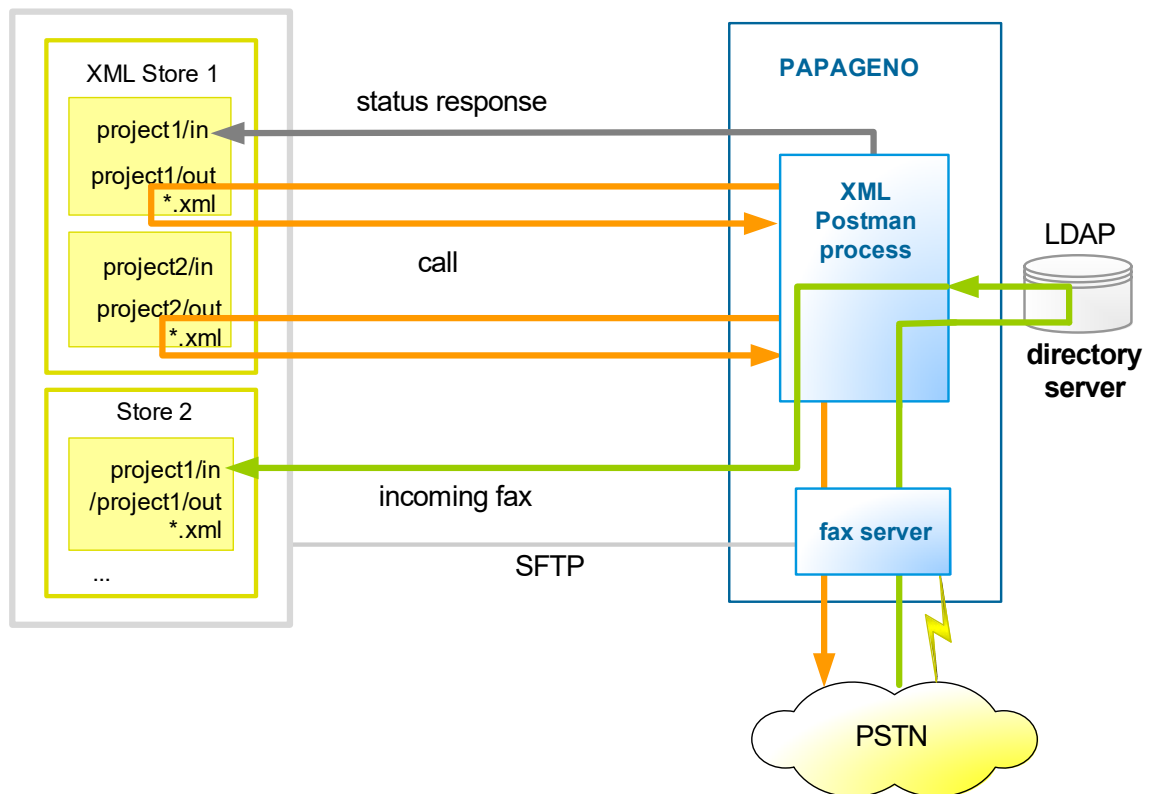
It is assured, that the control file (`inprojname.xml`) only exists when the incoming fax is complete.



## 3. Establishing XML Postman

### 3.1. Overview

Every store and every project is entered in XML Postman. The XML Postman contains the path to the stores and projects of the application software as well as the paths to the `in` and `out` folders.



The XML Postman catches all **send jobs** from the project outboxes and transmit them to PAPAGENO. Afterwards the XML-Postman saves the send fax status response in the project inbox.

The XML Postman converts documents with the formats `.txt`, `.pcl`, `.ps` or `.pdf` into the fax format `.tif`.

The fax number of an **incoming fax** the XML-Postman determines in the LDAP directory server. With the incoming fax the XML-Postman supplies an XML file with all appropriate fax data.

To establish the XML-Postman,

- enter the fax numbers and the mail addresses of the projects into the LDAP directory server (see below, page 18)
- copy the files into a directory (see page 18)
- publicize the XML-Postman in PAPAGENO (see page 19)
- Finally create the configuration file `xmlpostman.cfg` (see page 20)

## 3.2. Enter the Fax Numbers into the LDAP Directory Server

The XML Postman searches the `facsimileTelephoneNumber` attribute for the fax number of the incoming fax. Once found the XML Postman determines `project` and store via the `mail` attribute.

### To enter a project

- ▶ In the LDAP Directory Server switch to the `facsimileTelephoneNumber` or `proxyaddress:fax` attribute.
- ▶ Enter the fax number of the project.
- ▶ Switch to `mail` attribute.
- ▶ Enter name and store name of the project following the syntax `projectname.faxnumber@storeid`.  
`storeid` is the name of the project store (see above, chapter 2.2. "Creating Projects", page 9).  
Example: `reports.0898954750200@vipcomstore`
- ▶ Enter all projects names and fax numbers that way.

### If a project has more than one fax number:

- ▶ For each fax number enter an account with the attributes `facsimileTelephoneNumber` and `mail`.

## Entering a CSI

If a fax is redirected via a data center the recipient fax number changes. With a CSI entered the sender gets reconfirmed the original recipient number.

### If you want to ensure that the original recipient number will always be reconfirmed:

- ▶ For each project account enter a CSI in an appropriate directory server attribute.

### To find the CSI in the database:

- ▶ Set the `GD_LDAED` variable for the XML Postman user to the value of the CSI field name.  
(XML Postman user see also below „XML Postman-Benutzer eintragen“, Seite 19).

For further variables (e. g. search filter) see manual „PAPAGENO SMTP-Gateway“, Part B, chapter 9. "Configuring Telephone Access to Messages", page 44.

## 3.3. Copying XML Postman Files

### Prerequisite

The Postman must run on a PAPAGENO server.

- ▶ In `$FAXROOT` create a directory, e. g. with the name `postman`.
- ▶ Find the XML Postman files in your PAPAGENO installation files (version 5.9) in the `tools` directory
- ▶ Copy them in the `postman` directory.

### 3.4. Publicize XML-Postman in PAPAGENO

First of all you have to enter an XML Postman user in the ALPHA host of PAPAGENO. Then by using configuration variables, you have to set the host where Postman is running, the command, by which Postman will be started and the path to this command. Equally, you have to enter name and ALPHA host of this Postman user into the OMEGA database. (See below for more information)

If PAPAGENO is started on the XML Postman computer, the LAMBDA server is informed by an entry within the OMEGA database, that details for the Postman user are stored in the ALPHA database. Via the configuration variables the LAMBDA server gets the start command of the Postman.

#### Enter XML Postman User

- ▶ Start the PAPAGENO administrator.
- ▶ On a ALPHA Host enter a new user (only short name and description) e. g. `xmlpostman`.
- ▶ Select the `Extras` tab.
- ▶ Set the following variables

Variable	Value	Description
DM_HOST	<i>hostname</i>	Host name of the XML-Postman computer
DM_DIR	<i>verzeichnis</i>	Path to the XML-Postman in the PAPAGENO directory
DM_CMD	<i>Name_des Postmans</i>	Command to start the XML-Postman

- ▶ Save your changes.

#### Setting an OMEGA Configuration Variable

Because more than one XML Postman can be used the configuration variables are named `DAEMON0`, `DAEMON1` `DAEMON2`, etc.

- ▶ In a prompt enter  
`o_put_config DAEMON0 xmlpostmanuser@alphahost`

Example:

```
o_put_config DAEMON0 xmlpostman@faxserver3
```

### 3.5. Establishing the conversion into the Fax Format

#### Following formats are supported:

The XML-Postman can send `.txt`, `.pcl`, `.ps` files as a fax.

If the process control application software sends **PDF files**:

- ▶ Install the PAPAGENO tool `gs_pdftif`.  
See manual „PAPAGENO-Tools“ (german), page 14

### 3.6. Creating xmlpostman.cfg Configuration File

In the Postman directory (`xmlpostman`) create a file named `xmlpostman.cfg`. There you will enter several configuration data and the projects.

#### Example

<pre>#interface to PAPAGENO ALPHAHOST : comfax2 OMEGAHOST : comfax2 user = ngt PW = ***** defaultPrio = 320  # Logging control loglevel = 7 logage=5  # Loop control Min_Sleep = 5 Max_Sleep = 30 RunOnce = 1  # Interface to local file system tmpdir = tmpdir savedir = savedir  # Interface to the external storage sftpcmd = sftp BEGIN:Stores   BEGIN:ls02     UserHost : comfax@vpcls02     WorkDir : /daten/build/tmp/woso/testroot     Pattern : */out/job*   END:ls02   BEGIN:lpfax02test     UserHost : comfax@lpfax02.muc     WorkDir : /global/fax/data/applications- test     Pattern : */out/job*   END:lpfax02test END:Stores</pre>	<pre># List of project accounts BEGIN:Projects   BEGIN:woso@lpfax02test     CPI = 00493827000013     TSI = +49 382 70 00013     priority = 10000     HEADLINE = NG Account woso \$T\$ \$D%d %b %Y %X\$, Page \$p\$/     FileType = pdf     FillPage = 0     Comment: Testaccount     tries: 3   END:woso@lpfax02test   BEGIN:testproj@lpfax02test     CPI = 00498938241656     TSI = +49 89 382 41656     HEADLINE = NG Account testproj J. Miller \$T\$ \$D%d %b %Y %X\$, Page \$p\$/\$P\$     Comment: test account Miller     tries: 3   END:testproj@lpfax02test   BEGIN:test@ls02     CPI = 00498954750213     TSI = +49 89 54750 213     HEADLINE = NG Account test \$T\$ \$D%d %b %Y %X\$, Page \$p\$/\$P\$     Comment: test account     priority = 1000     tries: 3   END:test@ls02   # projects within other stores dont matter   BEGIN:test@lpfax02     CPI = 00498938241656     TSI = +49 89 382 41656     HEADLINE = NG Account test 1 \$T\$ \$D%d %b %Y %X\$, Page \$p\$/ \$P\$     Comment: test account Taylor     tries: 3   END:test@lpfax02 END:Projects</pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Description

### Interface to PAPAGENO

Keyword/Value	Meaning
OMEGAHOST= <i>hostname</i>	Name of the PAPAGENO computer with the main installation
user= <i>xmlpostman_username</i>	Name of the XML-Postman user (see above „Enter XML Postman User“, page 19“)
ALPHAHOST= <i>hostname</i>	ALPHA host of the XML-Postman user
PW= <i>password</i>	Password of the XML-Postman user
defaultPrio= <i>ziffer</i>	Default priority of faxes

### Logging Control

Keyword/Value	Meaning
loglevel= <i>ziffer</i>	1 – 7 Level of log messages
logage= <i>ziffer</i>	Number of days after which old logbooks will be erased

### Loop Control

Keyword/Value	Meaning
Min_Sleep= <i>ziffer</i>	Minimal waiting time between two passes (One pass consists: - picking up new faxes and dispatching them - getting new faxes - updating send status)
Max_Sleep= <i>ziffer</i>	Maximal waiting time between two passes
RunOnce	0: switched off 1: execute only one pass. For debugging.

### Interface to the External Storage

Keyword/Value	Meaning
tmpdir= <i>name</i>	Temporary directory
savendir= <i>name</i>	Backup directory

## Stores

Keyword/Value	Meaning
sftpcmd= <i>name</i>	Name of SFTP commands
BEGIN:Stores	Marks the beginning of the store data
BEGIN: <i>storeid</i>	Marks the beginning of the store <i>storeid</i> : Name of the store on the remote system
UserHost : <i>user@hostname</i>	Store user
WorkDir : <i>verzeichnispfad/workdir</i>	Absolute path to the working directory
Pattern : <i>*/out/job*</i>	Pattern how to find the jobs in the store WorkDir and Pattern reveal the search function <i>directorypath/WorkDir/*out/job*.xml</i> Therewith all open send jobs will be found in the store
END: <i>storeid</i>	Marks the end of a store <i>storeid</i> : Name of the store on the remote system
END:Stores	Marks the end of the store data

## Projects

Keyword/Value	Meaning
BEGIN:Projects	Marks the beginning of the project data
BEGIN: <i>projectid@storeid</i>	Marks the beginning of the project
Comment : <i>text</i>	Description of the project. Multiple lines are possible.
CPI = <i>cpi</i>	Connection identifier of the calling station.
TSI = <i>tsi</i>	Sender identification
priority : <i>ziffer</i>	Priority Faxes with a high priority will be sent before faxes with a lower priority even from other projects.
tries : <i>ziffer</i>	Number of automatic redials
HEADLINE = <i>headline</i>	Headline
FileType =	pdf or tif
FillPage =	0 or 1 0 means that the fax will be sent as fragments, 1 means that the fax will be sent as a page
End: <i>projectid@storeid</i>	Marks the end of the project

Keyword/Value	Meaning
END:Projects	Marks the end of the project data

For „CPI“ and „TSI“ see Manual: "PAPAGENO Configuration, Installation und Administration", part C, „How to Enter User's Fax Send Properties", page 98.

- ▶ In the XML Postman directory (`xmlpostman`) create a file named `xmlpostman.cfg`.

In the appendix on page 24 you'll find a template.

- ▶ Copy the template in an editor.
- ▶ Enter the configuration and project data.
- ▶ Save `xmlpostman.cfg`.

### 3.7. Starting and Stopping the Postman via Commands

Normally an XML Postman will be started via the LAMBDA server when PAPAGENO is started.

Manually you can start or stop the Postman as follows:

```
l_startdaemon xmlpostman_name
l_stopdaemon xmlpostman_name
```

`xmlpostman_name` is the short name of the XML Postman user in PAPAGENO.

## 4. Appendix

### 4.1. xmlpostman.cfg Template

#### **#Interface to PAPAGENO**

ALPHAHOST :  
OMEGAHOST :  
user =  
PW =  
defaultPrio =

#### **# Logging control**

loglevel =  
logage =

#### **# Loop control**

Min\_Sleep =  
Max\_Sleep =  
RunOnce =

#### **# Interface to local file system**

tmpdir = tmpdir  
savedir = savedir

#### **# Interface to the external storage**

sftpcmd = sftp  
BEGIN:Stores  
BEGIN:  
UserHost :  
WorkDir :  
Pattern : \*/out/job\*  
END:ls02  
END:Stores

#### **# List of project accounts**

BEGIN:Projects  
BEGIN:  
CPI =  
TSI =  
priority =  
HEADLINE =  
FileType =  
FillPage =



Comment:

tries:

END:

END:Projects

## 4.2. Errorcode

In general errorcodes are displayed in english. You can also use other languages.

### **german\_germany:**

Errstat 0 : unbearbeitet  
Errstat 1 : besetzt  
Errstat 2 : wird übertragen  
Errstat 3 : zuviele Versuche  
Errstat 4 : storniert  
Errstat 5 : erledigt  
Errstat 6 : Verbindung fehlt  
Errstat 7 : Übertragungsfehler  
Errstat 8 : Keine Antwort  
Errstat 9 : Timeout (Treiber)  
Errstat 10 : Gerätestörung  
Errstat 11 : Bitmapformat falsch  
Errstat 12 : Externe Übertragung  
Errstat 13 : Timeout (Route 1)  
Errstat 14 : Route 1 nicht erreichbar  
Errstat 15 : Externe Übertragung  
Errstat 16 : Timeout (Route 2)  
Errstat 17 : Route 2 nicht erreichbar

### **english\_us:**

Errstat 0 : idle  
Errstat 1 : busy line  
Errstat 2 : being transmitted  
Errstat 3 : too many retries  
Errstat 4 : cancelled  
Errstat 5 : done  
Errstat 6 : no line  
Errstat 7 : transmission error  
Errstat 8 : no answer  
Errstat 9 : driver timeout  
Errstat 10 : device error  
Errstat 11 : bad bitmap format  
Errstat 12 : routed  
Errstat 13 : route 1 timeout  
Errstat 14 : route 1 failed  
Errstat 15 : routed  
Errstat 16 : route 2 timeout  
Errstat 17 : route 2 failed

### **french\_france:**

Errstat 0 : Non traité  
Errstat 1 : Occupé  
Errstat 2 : En cours de transmission  
Errstat 3 : Trop d'essais  
Errstat 4 : Annulé  
Errstat 5 : Réalisé  
Errstat 6 : Pas de liaison  
Errstat 7 : Erreur de transmission  
Errstat 8 : Pas de réponse  
Errstat 9 : Timeout (Driver)  
Errstat 10 : Incident équipement  
Errstat 11 : Mauvais format de bitmap  
Errstat 12 : Transmission externe  
Errstat 13 : Timeout (Route 1)  
Errstat 14 : Route 1 inaccessible  
Errstat 15 : Transmission externe  
Errstat 16 : Timeout (Route 2)  
Errstat 17 : Route 2 inaccessible



