



# 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

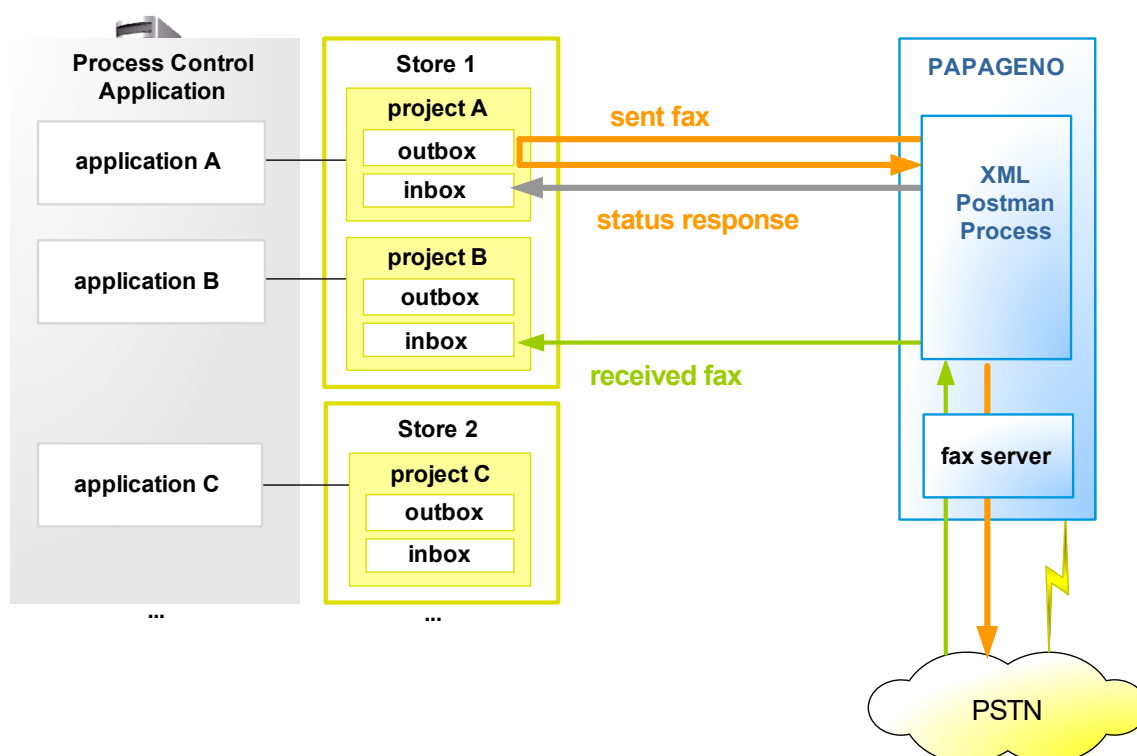
The use, reproduction or distribution of the program is subject to the restrictions described in your contract with VIPcom GmbH

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

The PAPAGENO XMLInterface ensures that faxes from process-driven application software are simply and securely sent and received.



**Outgoing faxes** are stored in the project store outbox folder by the application software. The PAPAGENO XML Postman Process fetches all these jobs and forwards them to the PAPAGENO Fax Server. The status feedbacks of the transmitted faxes are placed in the inbox of the respective project.

**Incoming faxes** are placed directly into the input system of the project in the store. For every received fax, an XML file is included which contains all data relevant to the fax. The process-controlled application can be programmed to read and evaluate data and fax contents.

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

- Either have or plan a **PAPAGENO installation in the house**

or

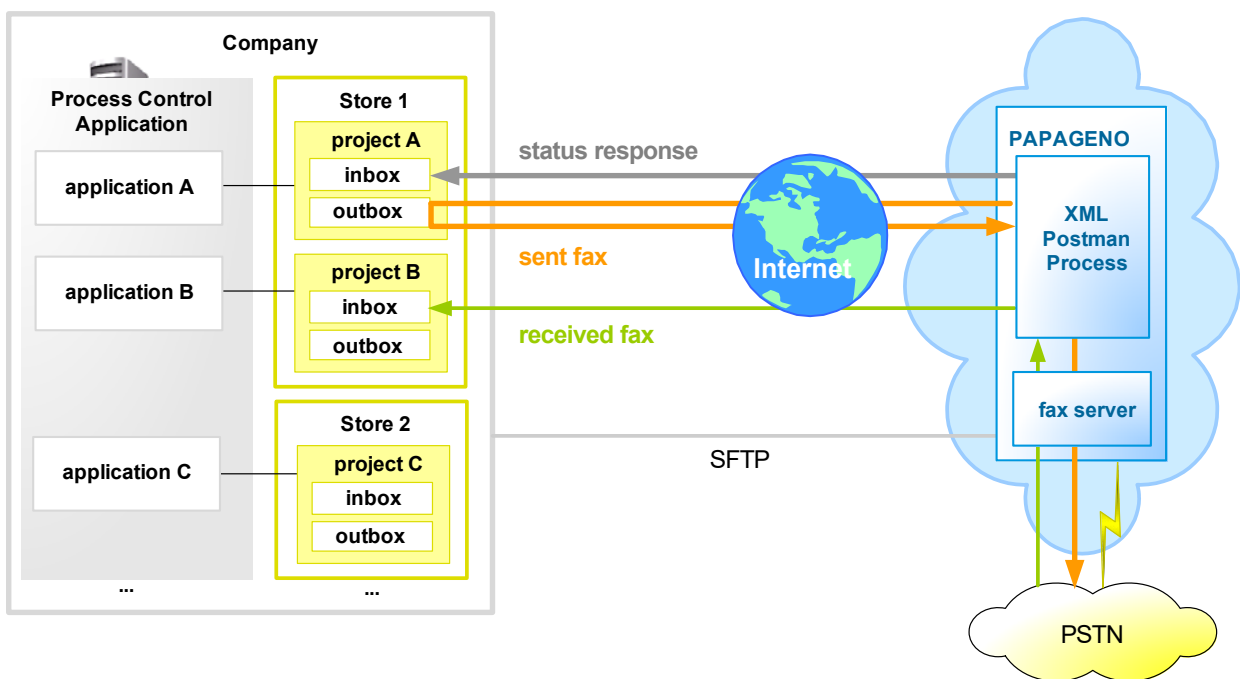
- use our **PAPAGENO installation in a data center.**

## In-house PAPAGENO Installation

- For incoming faxes you need an LDAP Directory Server and a PAPAGENO SMTP-Gateway. (see chapter 2. "Establishing Stores and Projects", page 9).
- You set up the store (s) and projects on the application page (see chapter 2. "Establishing Stores and Projects", page 9).
- You set up the XML postman for PAPAGENO (see chapter 3. "Establishing XML Postman", page 17).

## PAPAGENO Installation in a Data Center

- You only need to create and set up the store (s) with the respective 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
Send multiple fax documents in one job	12
Configure your 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 Fax Numbers in the LDAP Directory Server	18
3.3. Copying XML Postman Files	18
3.4. Publicize XML-Postman in PAPAGENO	19
Enter the XML Postman User	19
Setting an OMEGA Configuration Variable	19
3.5. Establishing the conversion to 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. Error Sttus .....26

# 1. PAPAGENO in-house Installation - Prerequisites

## LDAP Directory Server

When an incoming fax is received, the XML postman in the LDAP server will see which project the fax number of an incoming fax is assigned. This means that the fax numbers and names of the projects must be entered in the LDAP server.

### If no LDAP Directory Server is set up:

- ▶ Install it.

The project-relevant data to be entered is described in chapter 3.2. "Enter Fax Numbers in the LDAP Directory Server", page 18.

## SMTP Gateway

To access the LDAP server, the XML postman uses the variables set for the native user of the PAPAGENO SMTP gateway.

### If not done yet

- ▶ Install the PAPAGENO SMTP 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 are using a Linux operating system

- ▶ Make sure that the `expat` package is installed.





## 2. Setting up Stores and Projects

A process control application "project" consists of of so-called "projects" (eg "catering", „marketing“, „bookings“,... ).

### 2.1. Creating Stores

First, create one or more „stores“. A store is the location where, for example, all the projects of an application software are grouped together. It can be a separate computer or a folder in a file system. The XML postman needs the absolute path to each store.

- ▶ Create the store(s).

### 2.2. Creating Projects

In each store, you create the related projects with 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/  
      /project2 /in/  
        /  out/  
      /project3 /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 must be 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 is described below:

- the **name-giving syntax** of the documents to be sent 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 base name - a unique number sequence:

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

If a send job contains several identical documents (for example, two PDF files), *jobname* is extended by a number from the second document (in the example: name\_002).

#### Example:

```
jobcateringB201207 00017.xml belongs to 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 documents in the format .tif, .txt, .pcl, .ps and .pdf files as fax.

## Job control file (.xml)

The job control file with xml extension may only exist when it is complete and all document parts are present. It can first be endowed with the extension `.part` and later renamed to `.xml`.

### Content of a job control file:

The example entries are obliquely printed.

```
<?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 fax file name (`file-name`) may contain only letters (a-z), digits 1-9 and "\_" (underscore), NO special characters.

Only "-" (hyphen) or " " (blanks) are permitted as separators for the fax number (`fax-number`). These characters are automatically ignored when the number is read.

### 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.

In the JobControl file, you can add information tags as in any XML file. These are ignored during the sending process, but remain in the file. They are then added to the status feedback. This is in the input box and so the information in the tags can be evaluated.

Examples of information tags are "customer id" or "order number".

In information tags, only letters (a-z), digits 1-9 and "\_" (underscore) 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>
```

## Send multiple fax documents in one job

You can send multiple documents with 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 your Process Control Application

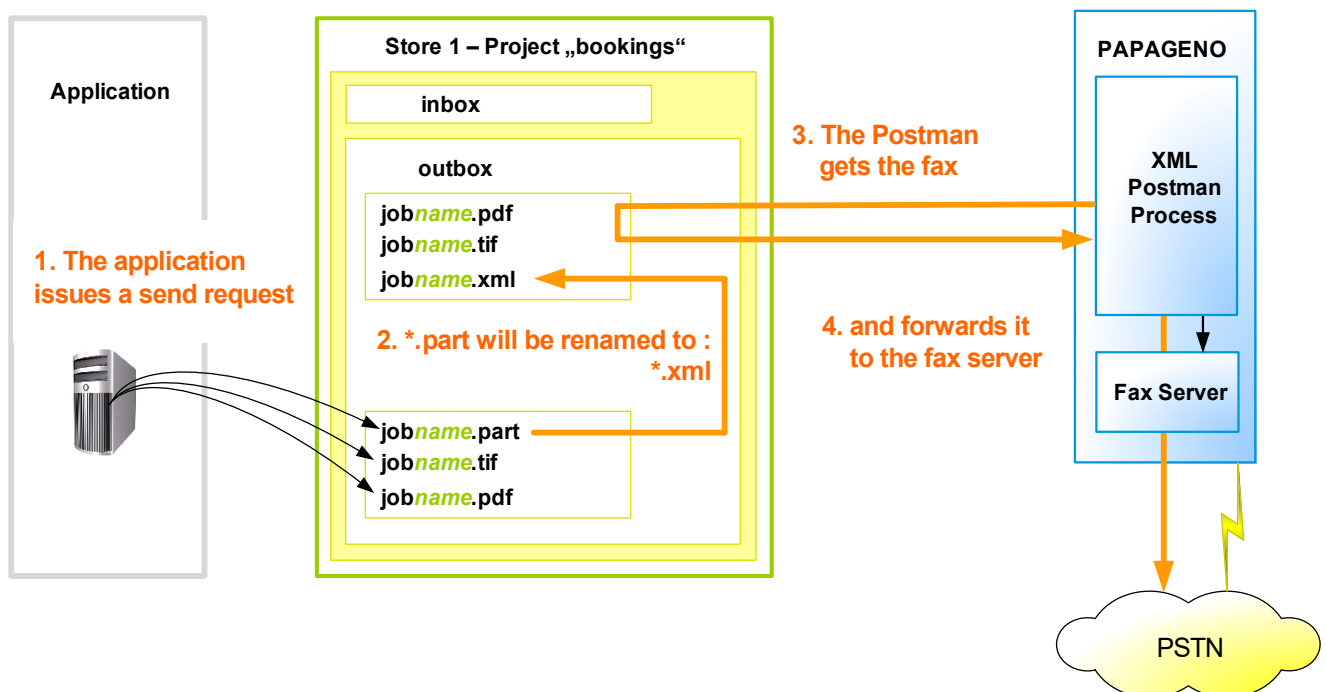
- Program the process control application software to move outgoing faxes to the out folder of the respective project and create a job control file. Note the above syntax rules (see „Name Giving Syntax”, page 10).

**Note!** Make sure all filenames are lowercase!

For example, name the control file `jobname.part` and rename it to `jobname.xml` after completion of the send job.

## Flow of the Send Process

With a single query, the XML Postman process captures all outgoing faxes in the outboxes and passes them to the PAPAGENO server.

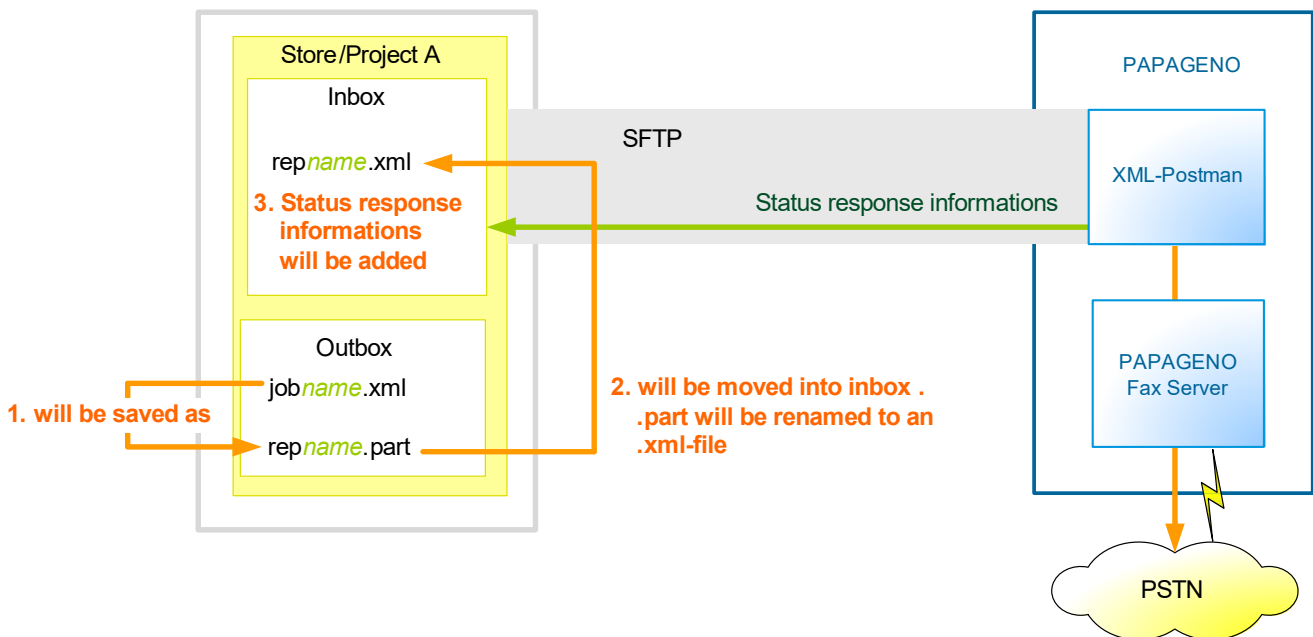


All files in an output box, which have the same basic name, are treated as a send job.

Already edited files in the output box will be deleted by the fax service.

## Status Response

After sending a fax the file „`jobname.xml`” is renamed to “`repname.part`” and moved to the „in”-folder. After the transfer is completed it gets the `.xml` suffix.



The status in which a fax is currently being sent is placed in the input file `repname.xml`. As soon as a new information arrives, the old status message is overwritten.

First the daemon adds a tag `<SendJobId>` for each fax address to the xml structure.

This is the internal ID for the processing of the transmission process by PAPAGENO. After the transmission job has been executed, the tags `<duration>`, `<status>`, `<timestamp>`, etc. are added.

## 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>
      </fax-address>
    </distribution-list>
  </fax>
</jobs>
```

```

        <status>OK</status>
        <remote-csid>+49/89/547500400-1702</remote-csid>
        <duration>46</duration>
        <timestamp>31. March 2017 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 2017 14:01:42</timestamp>
        <errorcode>47</errorcode>
        <reason>receiver not found</reason>
    </fax-address>
</distribution-list>
</fax>
</jobs>

```

**In a repname.xml file, the fax number can be extended by 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>: connection duration in seconds
<timestamp>: timestamp of sending time
<error code>: number of errorcode**

```

\* The status will be sent in any case

\*\* See page 27 for the list of error status messages.

## 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>
```

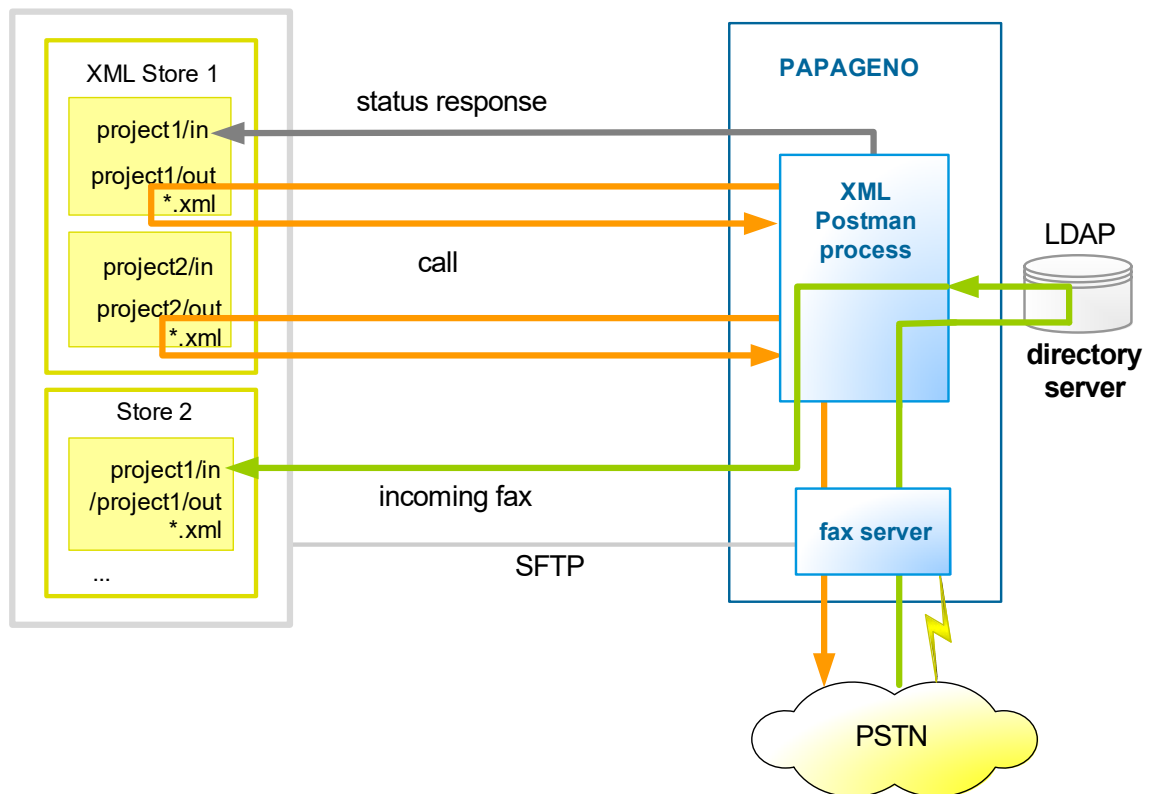
It is ensured that the control file (`inprojname.xml`) does not exist until it is complete and the corresponding document file (s) (`inprojname.tif`) is present.



## 3. Setup XML Postman

### 3.1. Overview

In the XML postman, the stores and the applications of the application software are entered as well as the paths to the `in` and `out` folders of the respective projects.



The XML Postman catches all **send jobs** from the project outboxes and transmits them to PAPAGENO. Status feedbacks of the transmitted faxes are placed in the input box of the respective project.

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

For incoming faxes, the XML postman uses the fax number in the LDAP database to determine the project to which the fax is to be delivered.

The Postman provides an XML file for each received fax containing all data relevant to the fax.

To set up the XML-Postman,

- enter fax numbers and the mail addresses of the projects in the LDAP directory server (see below, page 18)
- copy the files in 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 Fax Numbers in the LDAP Directory Server

In the LDAP database, the XML postman searches the `facsimileTelephoneNumber` attribute for the fax number of the incoming fax. If found, the XML postman uses the `mail` attribute to determine the project and store name.

### 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

In the case of call forwarding via a data center, for example, the recipient fax number changes. If a CSI is specified, the sender receives the recipient number to which he originally sent the fax.

### If you want to ensure that the correct recipient number is always sent to the sender:

- ▶ Enter a CSI in the LDAP directory server in a suitable attribute (field) for each project account

### To find the CSI in the database:

- ▶ For the XML postman user, set the `GD_LDAED` variable to the CSI field name.  
(XML Postman user see also below „XML Postman-Benutzer eintragen“, Seite 19).

For more GD variables, for example, for search filters, refer to the 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 computer.

- ▶ Create a directory for the postman in `$FAXROOT`, for example, with the name `postman`.

The XML postman files can be found in your PAPAGENO installation files, version 5.9, in the `tools` folder.

- ▶ Copy them to the `postman` directory.

### 3.4. Publicize XML-Postman in PAPAGENO

First of all you create an XML Postman user on a PAPAGENO ALPHA host. Then, via configuration variables, you set the host on which the Postman is running, the command via which it is started and the path to this command.

Using a configuration variable, enter the name and ALPHA host of the XML Postman user in the OMEGA database (see below).

If PAPAGENO is started on the Postman computer, the LAMBDA server finds in the OMEGA database the entry on which ALPHA server more detailed information on the Postman is stored. There, he uses the user configuration variable to determine where the postman is located and the command to start it.

#### Enter the XML Postman User

- ▶ Start the PAPAGENO administrator.
- ▶ Enter a new user on an ALPHA server (short description and description only), name, for example, `xmlpostman`.
- ▶ Go to 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 tree
DM_CMD	<i>Name_des Postmans</i>	Command to start the XML-Postman

- ▶ Save the entries.

#### Setting an OMEGA Configuration Variable

Since you can use several postmen, there are several configuration variables: `DAEMON0` set for the first, `DAEMON1` for the second, etc.

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

Example:

```
o_put_config DAEMON0 xmlpostman@faxserver3
```

### 3.5. Establishing the conversion to Fax Format

The Postman can convert these document types into fax format: `.txt`, `.pcl`, `.ps`.

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

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

### 3.6. Creating xmlpostman.cfg Configuration File

In the Postman directory (`xmlpostman`) you create a file `xmlpostman.cfg` into which you enter some configuration data as well as the stores and the current 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 computer with the PAPAGENO main installation
user= <i>xmlpostman_username</i>	Name of the XML-Postman user (see above „Enter the 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 are erased

### Loop Control

Keyword/Value	Meaning
Min_Sleep= <i>ziffer</i>	Minimum waiting time between 2 runs (A pass consists of: - pick up and send new faxes - receive faxes - update the sending status)
Max_Sleep= <i>ziffer</i>	Minimum waiting time between 2 runs
RunOnce	0: switched off 1: run should only be executed once. For debugging.

### Interface to the Local File System

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 the SFTP command
BEGIN:Stores	Marks the beginning of the store data
BEGIN: <i>storeid</i>	Marks the beginning of a 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 work directory
Pattern : <i>*/out/job*</i>	Here is the pattern of how the jobs are found in the specified store The search function <i>directorypath/WorkDir/*out/job*.xml</i> is formed from WorkDir and Pattern This will find all open send jobs 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 identification of the calling station.
TSI = <i>tsi</i>	Sender identification
priority : <i>ziffer</i>	Priority If high priority is given, these faxes are sent before those from other projects with lower priority.
tries : <i>ziffer</i>	Number of automatic redial attempts
HEADLINE = <i>headline</i>	Headline
FileType =	pdf or tif
FillPage =	0 or 1 0 means that the fax is sent as snippets, 1 means that the fax is sent as a whole page

Keyword/Value	Meaning
End : <i>projectid@storeid</i>	Marks the end of the project
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 will find a template.

- Copy this 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 is started via the LAMBDA server when PAPAGENO is started.

You can also start and stop the process separately using the following commands:

```
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 the 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. Error Status

Normally, the error stats are displayed in English. You can also set one of the 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



