



# **Funambol Email Connector & InboxListener Quick Start Guide**

# **Table of Contents**

1. Introduction.....	3
1.1. Release Notes.....	3
1.2. Known Issues.....	3
1.3. Related Documents.....	3
2. Funambol's Email Synchronization Environment.....	4
2.1. Environment Description.....	4
2.2. Installation Steps.....	5
3. Funambol's Email Synchronization Installation Procedure.....	6
3.1. Installing Funambol Email Connector.....	6
3.2. Officer Configuration.....	7
3.3. Configuring Email Connector.....	7
3.4. Configuring Email SyncSource.....	10
3.5. Installing Funambol InboxListener.....	12
3.6. IMAP Folder Subscriber.....	14
3.7. Encryption.....	15
3.8. Enabling Log.....	16
4. Notification System between Funambol DS-Server and Mobile Device.....	18
4.1. Mobile Device Settings.....	19
4.2. Admin Tool Settings.....	20
5. Appendicies.....	21
5.1. Microsoft Exchange 2000/2003 - Mail Server configuration.....	21
5.2. Domino 6.5/7.0 - Mail Server configuration.....	21
5.3. GMail - Mail Server configuration.....	22
5.4. Yahoo - Mail Server configuration.....	23
5.5. Courier Mail Server (Horde Web Access).....	23

# 1. Introduction

The purpose of this document is to describe how to manage and administer the Funambol Email Connector and the Inbox-Listener component. The admin user will use the FunambolAdmin console and a command line tool in order to configure the entire Funambol's email synchronization environment.

This document is intended to be read by the administrative users.

## 1.1. Release Notes

The current version of the email connector is compliant with the following clients:

- funambol-windowsmobile-plugin
- funambol-blackberry-plugin
- Nokia E60 (prototype)

The current version was tested with:

- Ms Exchange 2000/2003 Server
- Lotus Domino 7.0
- Zimbra Collaboration Suite
- Some unix/linux based Mail Server (Cyrus, Courier, ...)
- Some MS-Windows based Mail Server (Mercury, Winmail, ...)
- Some public Email Service Provider (Gmail, Yahoo, ...)

## 1.2. Known Issues

- ...

## 1.3. Related Documents

The following documents are related to this design document:

- [1] Sync4j Modules Development Guide
- [2] Sync4j SyncServer 5.0.x Administration Guide

## 2. Funambol's Email Synchronization Environment

### 2.1. Environment Description

Before starting with the installation procedure description we have to provide a quick explanation about the Funambol's email synchronization environment.

To implement the push email functionality and improve the performance, we start with the *Inbox-Listener Tool*. We configure this component to provide the following functionality:

- polling a Mail Server for new mail for a specific email account; when new mail has been received for that account the tool sends a notification to the DS server (inbox-listener feature)
- storing some basic email information (i.e. Message-ID, Date, ...) from the inbox folder into a local caching system in order to improve the synchronization performance. The user can define how many email the system should store in the caching system. (message processor feature)

To implement the email synchronization feature we have to install in the *Email Connector* in the DS Server. We configure this component to provide the following functionality:

- synchronize the email for a user between a Mail Server and a Mobile Device

The email synchronization process, as summarized below:

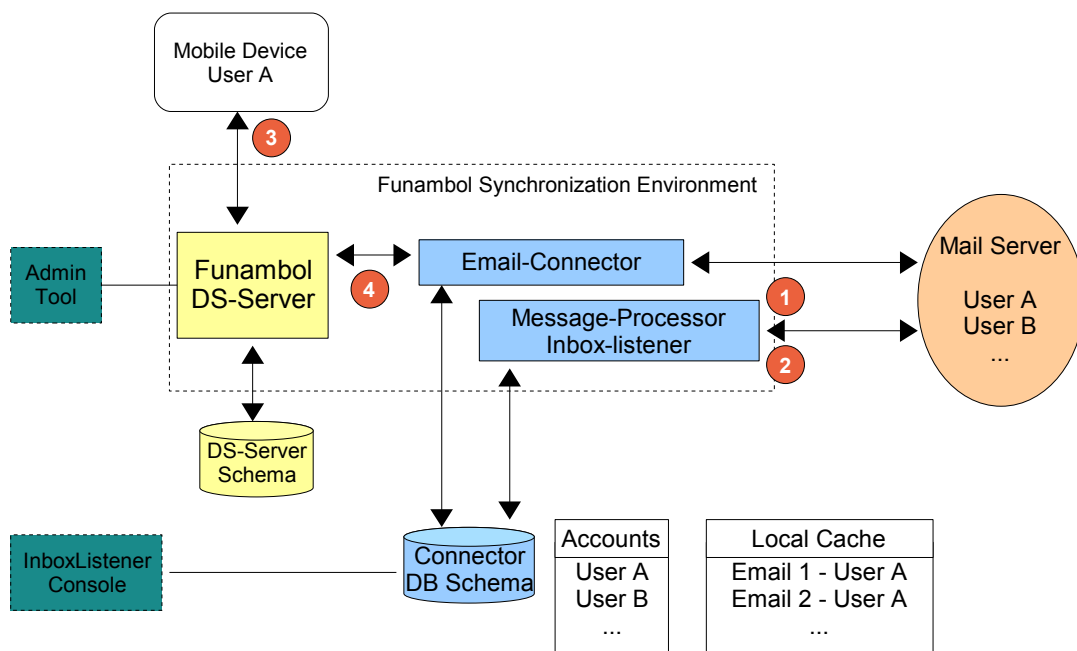


Figure 1: Funambol' email sync environment architecture

- 1) The *Messages-Processor* uses the “Connector DB Schema” and reads the users to sync from “Accounts”. It gets 'N' emails from the Server Inbox Folder and stores some basic info in the “Local Cache”. This behaviour allows the Email Connector to improve the performance during the sync process.
- 2) The *Inbox-Listener* uses the “Connector DB Schema” and reads the users to monitor from “Accounts”. It checks if in the Server Inbox Folder there is a new email, when new email has been received in the Server Inbox Folder it sends a notification to the DS-Server.
- 3) The *Mobile Device* performs a sync session. A sync session can be:
  1. driven by a DS server Notification. The DS Server sends a notification if a new email is in the inbox folder.
  2. driven by the user. The user starts the sync session pressing the 'sync button' on the syncML plug-in
- 4) The DS-Server uses the *Email Connector* in order to sync the email with the Mail Server. The *Email Connector* uses the Local Cache in order to improve the performance.

## 2.2. Installation Steps

In this paragraph we give a short list of the main steps in order to install the Funambol Email Connector & Inbox Listener:

- Install the Funambol Email Connector (see par. 2.3)
  - the installation automatically creates the Connector DB schema.
- Run the DS Server and the Admin Tool
- Configure the Officer (see par. 2.4)
  - set the officer in the Server Settings Panel in the Admin Tool
- Configure the Email Connector Properties (see par. 2.5)
  - set the parameters in the Email Connector Panel in the Admin Tool
- Create and Configure the Email SyncSources (see par. 2.6)
  - create and set the parameters in the Email SyncSource Panel in the Admin Tool
- [optional] Set the Encryption
- [optional] Set the log Level
- Install the Inbox-Listener Component (see par. 2.7)
  - configure the InboxListener.xml
- Run the Inbox-Listener *Console Tool* and insert Users credential in the Connector DB Schema.
- [optional, just for imap-based SyncSource] Run the *Subscriber Tool* in order to get the IMAP Folder Information
- set the “push environment” information in the Device Settings Panel (see ch. 4)
- Run the *Inbox-Listener Tool*
  - the Message-Processor loads the inbox folder emails in the cache table (just basic email info; NOT the entire email)
  - the Inbox-Listener starts polling the inbox folder.
- Get the Mobile Device and run a sync session from the Mobile Device or wait for a notification from the DS-Server

## 3. Funambol's Email Synchronization Installation Procedure

### 3.1. Installing Funambol Email Connector

The Funambol Email Connector is distributed as a standard Funambol module [1]. The distribution contains the following files:

- funambol-email-connector-<major>.<minor>.<buildnumber>.s4j (the module)
- the *Inbox-Listener Tool*
- this guide

To install the module you have to follow this steps:

1) put the s4j file in the directory

`<installation dir>\ds-server\modules`

2) modify “install.properties” file adding “funambol-email-connector-x.x.x” to the modules list:

`modules-to-install=foundation-3.0.0,pdi-1.5,pimweb-1.1,funambol-enterprise-activation-3.x.x  
,funambol-email-x.x.x`

3) start installation modules command.

For more details about the Funambol module installation see [2].

During the installation the following steps are performed automatically:

1. the database is initialized creating the connector specific tables and registering the connector into the server.
2. the EmailOfficer.xml file is copied in the directory:  
`<installation dir>\ds-server\config\funambol\server\security`

Note: before running the Funambol DS-Server you have to put the JDBC in the Application Server library folder; for instance in a Tomcat + Mysql installation you have to copy the mysql-connector-java-x.x.x-bin.jar library in the “<tomcat\_home>\common\lib” folder (just for brand new installation)

Note: the JavaMail library is already included in the distribution so that you don't have copy the mail.jar and the activation.jar in the Application Server library folder.

### 3.2. Officer Configuration

In order to set the Officer for the Funambol Email Connector, you have to set the parameters in the file; at the moment you have to set no parameters :

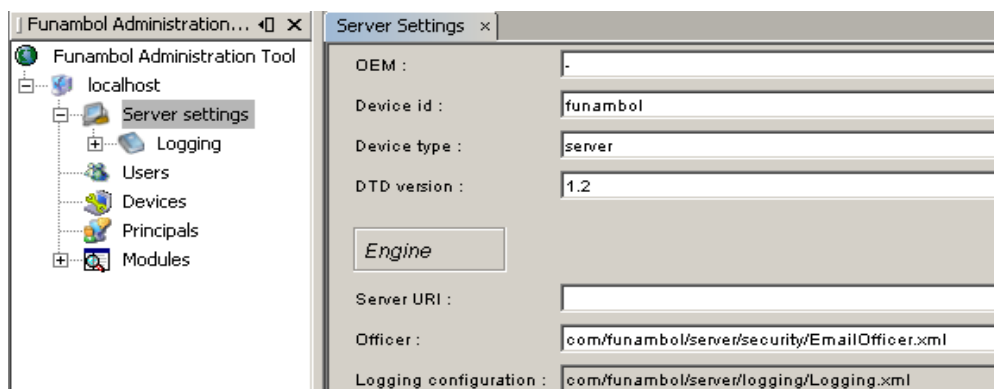
```
<installation dir>\ds-server\config\funambol\server\security\EmailOfficer.xml
```

This is an example of the EmailOfficer.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.4.0" class="java.beans.XMLDecoder">
  <object class="com.funambol.email.security.EmailOfficer">
    </object>
  </java>
```

In the Admin tool you have to specify the correct Officer:

Officer : com/funambol/server/security/EmailOfficer.xml

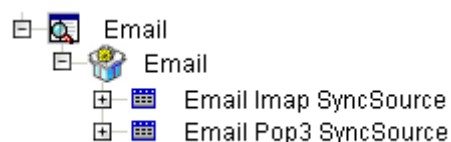


**Figure 2: Officer Settings**

The main information about the EmailOfficer are in the Email Connector Configuration Panel. See next paragraph for details.

### 3.3. Configuring Email Connector

In order to configure the Funambol Email Connector you can open the Funambol Administrator Console and browse in the following tree structure (Figure 3)



**Figure 3: Connector Tree**

When you point on the Email Connector item you can see the following panel (Figure 4).

**Funambol Email Connector**

**Officer Configuration**

Mail Server: localhost
Protocol: pop3
Port: 110
Enable Account Registration: ☒
Listener ID: 1
SSL: ☐
Keystore Path:
Keystore Passphrase:

**Funambol Configuration**

Funambol DataSource: jdbc/fnblds

**Properties Configuration**

Filter Activation: ☒
Save Only Header in DB: ☐
Max Sent Emails: 20

Save

**Figure 4: Email Connector Configuration Panel**

In this panel you can set the parameters like in the table below:

<i>Property</i>	<i>Description</i>
Mail Server	The Email Officer uses this Mail Server in order to authenticate the account
Protocol	The Email Officer uses this protocol in order to authenticate the account
Port	The Email Officer uses this port in order to authenticate the account
Enable Account Registration	The Email Officer saves the account in the “Accounts” table of the Connector DB Schema (see Account Automatically Registered paragraph)
Listener ID	(see Account Automatically Registered paragraph)
SSL	SSL Activation
Keystore path	Path of the java keystore



<i>Property</i>	<i>Description</i>
Keystore password	Password of the java keystore
Funambol DataSource	Funambol Datasource. (Used by the Caching system)
Apply Filter	You can choose to use or not the filters
Save only Header	Save just the header in the Local Sent Table (Used for pop-based SyncSource)
Max Sent Emails	(only for IMAP-based SyncSource) define the max number of emails that will synchronized in the Sent Folder

### **Account Automatically Registered**

This feature can be used in the following scenario:

- single Mail Server
- single SyncSource; the admin can create just one SyncSource (i.e. pop-based SyncSource)
- Email Officer settings in according on the SyncSource (i.e. pop-based SyncSource)

In this scenario the admin can set the following values

- Enable Account Registration = true
- listenerID = 1 (check the InboxListener.xml file)

In this scenario the admin can avoid to insert every single account using the Inbox-Listener Console Tool.

The Email Officer inserts a new account in the “Account” table. In fact in the first sync the credentials are not in the Local DB, but the Email Officer inserts the account in the Local DB.

During the first sync only this “activation” happens; after the first sync the user can sync the emails between the Mail Server and the Syncml Smail Client.

Using the current Email Connector architecture there is a difference between the Mail Server situation and the Message-Processor situation. Every 'x' minutes a synchronization starts between the two systems, so that the “ds-server user” must wait until the two system are in sync before starting a sync session from the syncml email client.

### 3.4. Configuring Email SyncSource

The email connector provides the following SyncSource type:

- “POP-based” SyncSource; using this SyncSource the User can download the emails [INBOX folder synchronization] and send the email from the client [OUTBOX/SENT folders synchronization])
- “IMAP-based” SyncSource; using this SyncSource the User has a complete mailbox synchronization feature.

When you point on the Email Imap SyncSource or Email Pop3 SyncSource item you can see the Figure 5

**Edit Email Connector SyncSource**

Source URI:

Name:

Server Type:

Outgoing Server:  Port:  Auth: ☐

Incoming Server:  Port:

SSL: ☐

Keystore Path:

Keystore Passphrase:

Complete Mailbox Activation: ☐

Inbox Folder Name:  ☒

Outbox Folder Name:  ☒

Sent Folder Name:  ☒

Drafts Folder Name:  ☐

Trash Folder Name:  ☐

Type:

Supported Types:

Supported Versions:

Encryption/Encoding: ☐

Soft Delete: ☐

**Figure 5: Email SyncSource Configuration Panel**

Note: for the pop-based syncsource the following text fields are fixed

- complete mailbox activation
- inbox, outbox, sent, drafts, trash folder
- all checkbox activation for the default folder

In fact the pop-based sync source can synchronize only the Inbox folder and send the email from the device.

In the SyncSource Panel you can set the following properties:

<i>Property</i>	<i>Description</i>
Source URI	The sync source URI [i.e. "mail" ].
Name	The SyncSource name. [i.e. "mail" ]
Outgoing Server / Port	Default SMTP port : 25
Incoming Server / Port	Default Imap port : 143, pop3 port : 110
Complete Mailbox Activation	You can choose to sync only the main default folders (Inbox, Outbox, Sent, Drafts, Trash) or the entire mailbox (this feature is not tested at the moment and it's disabled)
Folder Name / activation	For each main default folder you can define the Server Label and the activation  (see the Inbox-Lister IMAP Folder Tool – par. 3.6)
Type	application/vnd.omads-email+xml,application/vnd.omads-folder+xml
Supported Type	application/vnd.omads-email+xml,application/vnd.omads-folder+xml
version	1.2,1.2
encryption	the communication between syncml client and ds-server will be encrypted (DES and BASE64)
Soft Delete	The Email Connector doesn't remove the item on the mail server if there is a DELETE command from the client

### 3.5. Installing Funambol InboxListener

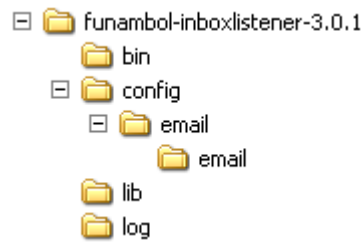
The Funambol InboxListener component is included in the Funambol Email Connector archive file:

– funambol-email-<major>.<minor>.<buildnumber>.zip

NOTE: Before Installing the Inbox-Listener you have to install the Funambol Email Connector. (the email connector installation procedure creates the inboxlistener schema in the database)

To install the module you have to follow this steps:

1) unzip the ".zip" and check the directory structure:



**Figure 6: inbox listener home**

2) modify the run script:

- startlistener.\*
- console.\*;
- subscribe.\*

change JAVA\_HOME and JDBC\_JAR according to your environment (in the bin directory)

Note: in the current version the “sh” script should be handled with the commands: *dos2unix* and *chmod* in order to assign execution permission

3) modify “email/ema/InboxListener.xml” according with your environment

Property	Description
listnerid	Inbox-Listerner module info; listener id linked to the mailboxes
startupNotification	If true when the module starts, it sends a notification for all the registered user to the sync server
interval	Mailbox polling interval. Every “x” minutes the Inbox-Listener checks the mailbox
reloadingTime	Every “x” minutes the Inbox-Listener checks the list of accounts in the DB
jdbcDriver	i.e. com.mysql.jdbc.Driver
jdbcUrl	i.e. jdbc:mysql://localhost/funambol_500_ent
jdbcUser	Database username
jdbcPassword	Database password
wsEndPoint	i.e. http://localhost:8080/funambol/services/admin
wsUsername	Funambol Administrator user
wsPassword	Funambol Administrator password
wsSyncSourceType	Type of the SyncSource; i.e. “imap” or “pop3”
wsSyncSource	Name of the Email SyncSource

**Note: An single installation of the Inbox-Listener serves a single SyncSource. If there is a pop-based SyncSource and a imap-based SyncSource the admin MUST install two Inbox-Listener instances with different “listenerid” and “wsSyncSource” values**

- 4) Configure the Users in the Connector DB Schema
- Open a Shell
  - go in the <installation dir>\bin directory
  - run the console.\* script (the main menu in the Figure 7 appears)



```
INBOX LISTENER ACCOUNT CONSOLE
0  quit
1  list accounts
2  get account
3  insert account
4  delete account
5  update max email number
6  update password
7  update activation
```

**Figure 7: Inbox Listener Main Menu**

- command 1: list all the mailbox accounts that should be monitored
- command 2: show the account detail
- command 3: create a new accounts
  - you have to insert:
    - username
    - password
    - inboxlistener id (in according to the xml file)
    - max number of cached emails
- command 4: delete an account
- command 5: update the max number of cached emails for a specific account
- command 6: update the password for a specific account
- command 7: update the activation status for a specific account

In order to configure the DB information at the moment there is only a command line tool. In the next release we provide a friendly user interface.

The Email Officer inserts a new account in the “Account” table, so that the admin can avoid to insert every single account. In fact in the first sync the credentials are not in the Local DB, but the Email Officer inserts the account in the Local DB.

During the first sync only this “activation” happens; after the first sync the user can sync the emails between the Mail Server and the Syncml Smail Client.

Using the current Email Connector architecture there is a difference between the Mail Server situation and the Message-Processor situation. Every 'x' minutes a synchronization starts between the two systems, so that the “ds-server user” must wait until the two system are in sync before starting a sync session from the syncml email client.

### **3.6. IMAP Folder Subscriber**

The Funambol InboxListener component includes a tool that allows the administrator to see the folder name for a specific account. This tool can be used as helper in order to complete the creation of a “IMAP-based” SyncSource.

In order to get the information about a specific account follow this steps:

- Open a Shell
- go in the <installation dir>\bin directory
- run the subscribe.\* script (the main menu in the Figure 8 appears)

```
IMAP FOLDER MANAGER TOOL
0  quit
1  select account
```

**Figure 8: Subscriber main menu**

- command 1: list all the folders for a specific account

the result should be like this:

```
IMAP Folders in the Account gilberto.migliavacca
folder name: Deleted Items - full name: Deleted Items
folder name: Drafts - full name: Drafts
folder name: INBOX - full name: INBOX
folder name: Outbox - full name: Outbox
folder name: Sent Items - full name: Sent Items
```

**Figure 9: Subscriber sample results**

Note: this tool works only for IMAP-based SyncSource. If in the InboxListener.xml there is a reference to a pop3-based SyncSource the tool gets an error.

### 3.7. Encryption

In order to enable the Encryption Communication between syncML client (i.e. The Funambol WM plugin) and the Funambol DS-Server/Email Connector you have to check the “encryption/encoding checkbox in the SyncSource configuration Panel

Type: application/vnd.omads-email+xml,application/vnd.omads-folder+xml

Supported Types: application/vnd.omads-email+xml,application/vnd.omads-folder+xml

Supported Versions: 1.2,1.2

encryption/encoding: ☐

**Figure 10: encryption settings**

if the encryption is enabled the synchronization will be provided with DES and BASE64 encoding

You can also check the configuration in the Data transformation panel in the Server Settings section

- Open the Funambol Admin Tool
- Settings – Data Transformation – Configure
- In the DataTrasformer Configuration Manager; add a raw in the Data Transformations table

Source URI	Transformation
scard	b64
briefcase	b64
scal	b64
stask	b64
note	b64

**Figure 11: Data Transformation Table**

- press “+”
  - SourceUri : set the SyncSource Name [i.e. “mail”]
  - Transformation: set “**des;b64**”
- press “save”

In this way the server is enabled to handle the encryption data. Meanwhile you have to set on the client the “encryption” property.

### 3.8. Enabling Log

#### Enabling log in the Email Connector

In order to configure the logging level for the Funambol Email Connector, you have to modify the file:

`<installation dir>\ds-server\config\sync4j\server\logging\Logging.xml`

adding this rows:

```
<void method="add">
  <object class="com.funambol.framework.config.LoggerConfiguration">
    <void property="append">
      <boolean>true</boolean>
    </void>
    <void property="count">
      <int>1</int>
    </void>
  </object>
</void>
```

```

</void>
<void property="inherit">
  <boolean>true</boolean>
</void>
<void property="level">
  <string>INFO</string>
</void>
<void property="limit">
  <int>100</int>
</void>
<void property="name">
  <string>funambol.email</string>
</void>
<void property="pattern">
  <string>logs/syncserver.email.log</string>
</void>
</object>
</void>

```

This file is re-written during server installation (or module installation) so, if you don't want to lose the changes, you may insert the same rows in the file:

```
<installation dir>\ds-server\default\config\common\beans\funambol\server\logging\Logging.xml
```

### Enabling log in the InboxListener

In order to configure the logging level for the Funambol InboxListener, you have to modify the file:

```
<installation dir>\bin\inboxlistenerlog.properties
```

```

# Specify the handlers to create in the root logger
# (all loggers are children of the root logger)

handlers=java.util.logging.ConsoleHandler

# To also add the FileHandler, use the following line instead.
# handlers= java.util.logging.FileHandler, java.util.logging.ConsoleHandler

# Set the default logging level for the root logger
.level= INFO

java.util.logging.FileHandler.level = ALL
java.util.logging.FileHandler.pattern = ../log/il.log
java.util.logging.FileHandler.limit = 50000
java.util.logging.FileHandler.count = 1
java.util.logging.FileHandler.formatter = java.util.logging.SimpleFormatter

java.util.logging.ConsoleHandler.level = ALL
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter

#####
# Facility specific properties.
# Provides extra control for each logger.
#####

funambol.email.level = FINEST

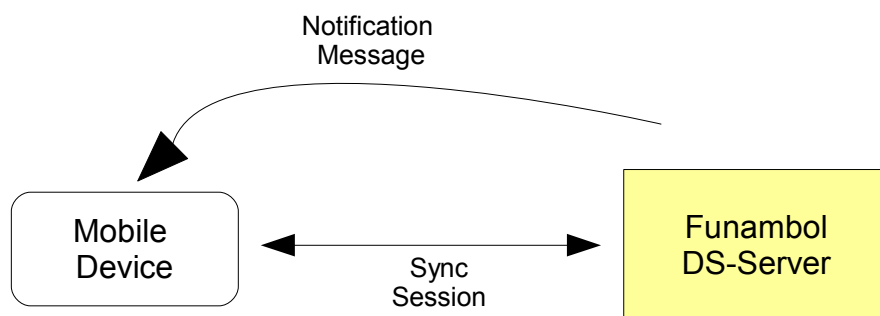
```

You can redirect the log in a log file set the paramter  
 handlers=java.util.logging.FileHandler  
 All the properties of the log file are already set.



## 4. Notification System between Funambol DS-Server and Mobile Device

In this chapter we show briefly the main information about the communication between the Funambol DS-Server and the Mobile Device (Figure 12). A detailed description is included in the specific Funambol Server and Funambol Windows Mobile Plug-in documentation.



**Figure 12: Communication between Server and Device**

The Funambol DS-Server sends notification to the device using two protocol.

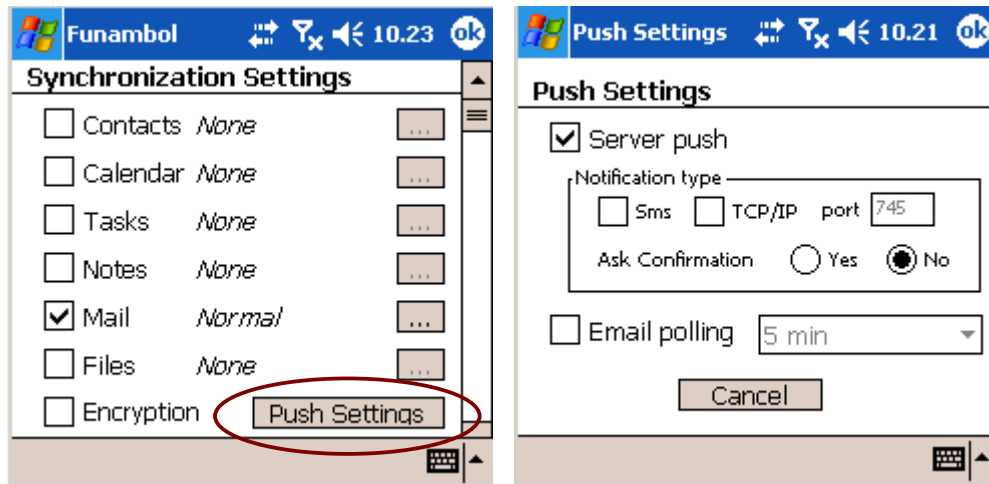
- **WAP protocol:** the engine sends the notification sending an sms. In this case we need
  - activate an SMS service (we have to ask a service to an SMS Provider)
  - provide a modules to install in the funambol engine (an s4j file .. )This module should implement the specification of the SMS provider (i.e. Send a http request with the sms to the provider server) In order to develop this module you can see the class `com/funambol/server/notification/sender/SimpleWAPSenderImpl.java` in the Objectweb cvs
- **TCP-IP protocol:** the engine send to the device a message using the tcp-ip protocol. In the case we need a Mobile Device with a public IP when the “push” is turned on.

When the client plug-in on the device receives the notification automatically performs an email synchronization via the Email Connector module.

Remember to set the push properties on the plug-in settings and to set the properties in the Admin tool

## 4.1. Mobile Device Settings

On the Mobile Device you have to set the following properties



**Figure 13: Mobile Device Settings**

In the Push settings you can choose if the notification is via Sms or TCP/IP

## 4.2. Admin Tool Settings

In order to set the properties for the DS-Server we have to use the Admin Tool.

- open the admin tool
- choose <server-name> ---> Devices ---> Edit



The screenshot shows a 'Device Details' dialog box with the following fields and values:

- ID :** fwm-0ED6FF542B91977E0
- Type :** (empty)
- Timezone :** Not specified (dropdown menu)
- ☐ Convert dates to this timezone
- Charset :** UTF-8 (dropdown menu)
- Address :** (empty)
- Msisdn :** (empty)
- Notification Builder :** (empty)
- Notification Sender :** (empty)
- Description :** (empty text area)

Buttons at the bottom: Save, Cancel

**Figure 14: DS - Server Settings**

We have to handle the following properties

- **Address:** IP address of the device (if applicable).
- **Msisdn:** Msisdn of the device (i.e., the phone number)
- **Notification Builder:** The builder (server component) used to create notification messages for this device. Example:  
<DS\_SERVER\_HOME\config\com\funambol\server\notification\DSNotificationBuilder.xml
- **Notification Sender:** The sender (server component) used to send notification messages to this device. Example:  
<DS\_SERVER\_HOME\config\com\funambol\server\notification\WAPSender.xml

## 5. Appendicies

### 5.1. Microsoft Exchange 2000/2003 - Mail Server configuration

We can use:

- imap-based syncsource
- pop3-based syncsource

We have to setup the MS Exchange Server

- creation of the user in the MS Exchange
- login using “outlook web access” or an account imap on outlook express / outlook

SyncSource Folder Name Configuration

Folder name

- inbox: inbox
- outbox: outbox
- sent: Sent Items
- drafts: Drafts
- trash: Deleted Items

### 5.2. Domino 6.5/7.0 - Mail Server configuration

About imap-based syncsource we have to consider that some aspects of a mail file are structured in template items that are visible only to a Notes client, and as such are not available to IMAP clients.

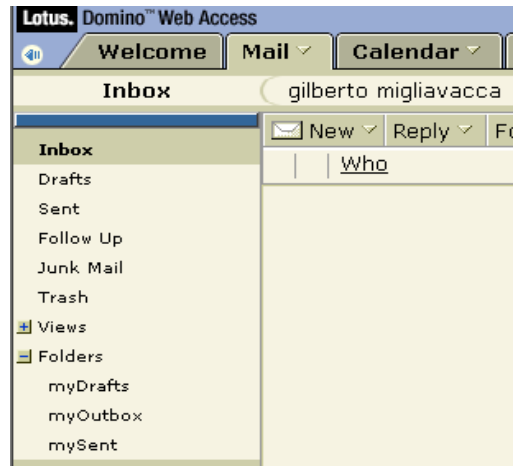
As a result, IMAP clients display certain folders and views in a mail file differently from Notes clients. For instance, from an IMAP client, the Inbox and Trash folders, and any public folders, appear as IMAP mailboxes.

Also, hidden and private folders are not visible to IMAP clients. And finally, IMAP clients do not display views that are part of the Notes mail file template, such as the Draft and Sent view

We have to setup the “domino” mail server

- creation of the user
- login using “Domino Web Access”
  - create the public folder in “folder”
    - pubOutbox
    - pubSent
    - pubDrafts

Note: the “sent” and “draft” folders are “private folder”. The Email connector will use the “public folder”. As a result, the connector displays/uses some folder/email in a mail file differently from Notes clients.



### SyncSource Folder Name Configuration

Folder name

- inbox: inbox
- outbox: myOutbox
- sent: mySent
- drafts: myDrafts
- trash: Trash

### 5.3. GMail - Mail Server configuration

Gmail doesn't currently support IMAP access.

As part of Gmail ongoing commitment to give Gmail Users easy access to their email, Gmail has introduced Pop access. In order to config the SyncSource you have to set the following parameters

field	value
Incoming server	pop.gmail.com
Incoming server Port	995
Outgoing server	Smtp.gmail.com
Outgoing server Port	465
SSL	checked
Keystore path	See next info
Keystore password	See next info

The Gmail pop access need SSL configuration.

1) Download the GMAIL certificate from link:

[https://www.geotrust.com/resources/root\\_certificates/certificates/Equifax\\_Secure\\_Certificate\\_Authority.cer](https://www.geotrust.com/resources/root_certificates/certificates/Equifax_Secure_Certificate_Authority.cer)

or

using cygwin type the command

wget

[https://www.geotrust.com/resources/root\\_certificates/certificates/Equifax\\_Secure\\_Certificate\\_Authority.cer](https://www.geotrust.com/resources/root_certificates/certificates/Equifax_Secure_Certificate_Authority.cer)

2) Use the java tool in order to create the keystore (using the certificate just downloaded). open a shell (i.e. A dos shell) and type the command

```
> keytool -genkey -alias gmailcert -keyalg RSA -keystore gmailcertkeys
```

Note: you have to install the jdk on the machine

3) Import the certificate in the keystore:

```
> keytool -import -file Equifax_Secure_Certificate_Authority.cer -keystore gmailcertkeys
```

4) Set the syncsource and the connector to use the SSL parameter

## 5.4. Yahoo - Mail Server configuration

We can use:

- pop3-based syncsource

We have to setup the “[yahoo.com](http://yahoo.com)” mail server.

- creation of the user
- login using “web access”

SyncSource Configuration

Source URI:	<input type="text" value="pyahoo"/>				
Name:	<input type="text" value="pyahoo"/>				
Outgoing Server :	<input type="text" value="smtp.mail.yahoo.it"/>	Port:	<input type="text" value="587"/>	Auth:	<input checked="" type="checkbox"/>
Incoming Server :	<input type="text" value="pop.mail.yahoo.it"/>	Port:	<input type="text" value="110"/>		

Note: yahoo server needs the authentication in order to send the email

Note: currently the yahoo server needs the “Yahoo Mail Plus” option (it's not a free option) in order to allow the POP configuration.

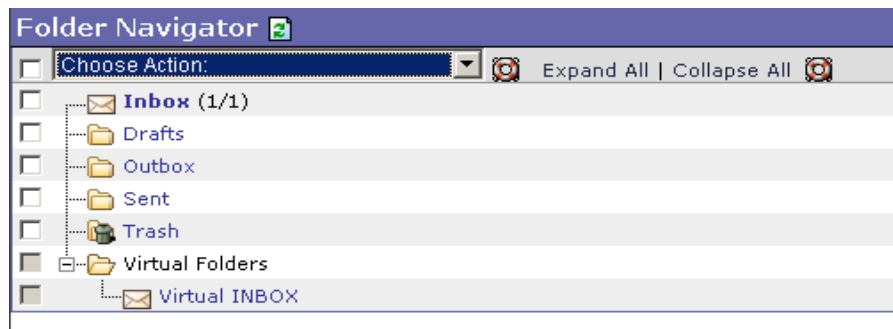
## 5.5. Courier Mail Server (Horde Web Access)

We can use:

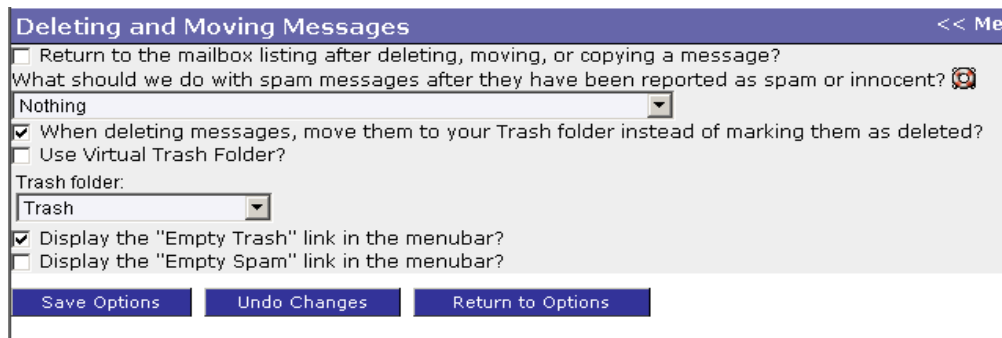
- imap-based syncsource
- pop3-based syncsource

We have to setup the “[mail.\\*\\*\\*\\*\\*.com](http://mail.*****.com)” mail server (Courier Imap Server).

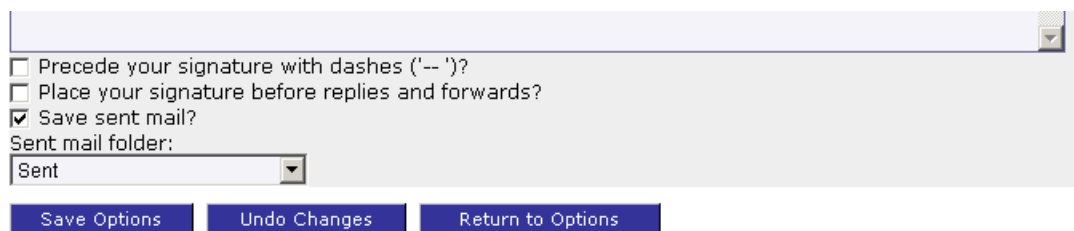
- creation of the user
- login using “Horde Web Access”
  - select INBOX on the left panel
  - select “folder” in the top bar
  - choose action: create folder
  - create: Sent, Outbox, Drafts



- select “options”
- choose “deleting and moving messages” box
- select the following option



- select “options”
- choose “Personal Information” box
- select the following option (at the end of the panel)



- select “options”
- choose “Server and Folder Information” box
- select the following option



## SyncSource Folder Name Configuration

### Folder name

- inbox: Inbox
- outbox: INBOX.Outbox

- sent: INBOX.Sent
- drafts: INBOX.Drafts
- trash: INBOX.Trash