Maiken Winterberg File Domain Jumper

By Martin Alexander Thomsen on 20 July 2024

The Maiken Winterberg File Domain Jumper is a tool for document exchange between domains. In order to route a document the jumper needs a domainname of the receiver of the document. It finds a receiver of the document by using the configured router of the folder of which the document is in. Each folder can have its own configuration. The jumper supports static, property, xpath, registry, hashtag and folder routing. If that is not enough you can implement your own routing by implementing the IDocumentRouter interface.

Requirements: In order for The Maiken Winterberg File Jumper to work properly you need to install Java 1.17+ onto your computer. And you need a static IP address along with a domainname that points to your address. If you have no static IP and a domainname I can recommend https://godaddy.com, https://cloud.google.com, https://aws.amazon.com and https://azure.microsoft.com. You wil need to install a MaikenWinterbergSocketRegistry along with your MaikenWintebergFileDomainJumper. The registry is open on port 4554 and allow the systems to find each other.

The jumper support multiple options of what to do with the file when it has been sent. It can put the file in another folder (error folder, sent foler, domain not found folder) or it can stay put where it is (with the synchronizedFileProcess) or delete it altogether.

The Maiken Winterberg File Jumper support a plugin solution for the security. If you want to change the default security implementation you can create your own implementation of the com.maikenwinterberg.socketregistry.security.IRegistrySecurity interface.

By default its using the H2 database (https://www.h2database.com/html/main.html) but you can use your your own DB by changing the JDBC configuration. There are 2 databases. One for the registry and one for the file domain jumper. The jumper uses the database in order for the synchronizedFileProcess to keep track of what files has been sent to what domains.

Why not emails? And email do not do routing based on the content of the file, and an email cannot send a whole tree of documents(unless you zip your tree into a single file) and route each document induvidually. Futhermore, emails do not allow you to iplement your own security (unless you encrypt the file manually) and emails do not support subscription. Also, the jumper allow you to turn your inbox into an outbox and you will become a node in a tree of nodes. The jumpler has a library function that allow you to subscribe to 1000+ of subscriptions at the same time.

Installation

Installation is simple. You go thrugh all configuration files and change localhost with your domainname (in the conf folder). You can copy the conf folder from the bin into the home directory if you like Then you go to the linux_service or the window_service folder and follow the documentation in the readme.txt file. If you want to run it on a Apple computer you need to do your own research. It shoud be possiple to make it work there also. In a future release I will also support Apple. The linux installation creates two services. A MaikenWinterberFileReceiver.service and a MaikenWinterbergFileSender.service. The FileReceiver puts document into the inbox (located in MaikeWinterbergFileDomainJumper/box/inbox) and the fileSender sends document from the many configured outboxes (located in MaikenWinterbergFileJumper/box/outbox). The sentbox, errorbox and the inbox is categorized by domainname. You can by the foldername see who sent the document to you and who you have sent document to. When you setup the MaikenWinterbergFileReceiver you have the option to select a limit of how big files received can be. If you are being flodded with documents by a bad source you can put the domainname on the inValidDomains.cfg list. Once installed you can test by sending a document to yourself. Make a directory with your domainname into the outbox and insert a file into it. Then, check if you have received a document in the inbox.

Maiken Winterberg File Sender

You configure the FileSender in the file conf/fileSender.properties. When configured you restart the service or wait 10 minutes. On linux you invoke the reInstallMaikenWinterbergFileSender.sh file in the service_linux folder in order to restart(the file will be created when you install the service).

Routing

In order to select a router for your documents you must configure the conf/fileSender.properties file. On folder level you must update the ${folderIndex}.routingclass attribute with the router selected for the folder. The parameters of the folder configuration startes with an index followed by a dot. The first folder configuration must start with 1 and the next 2 and so forth.

Example of conf/fileSender.properties

```
#please replace all localhost variables with the domainname on which your system is installed
defaultDomainNameOfClient=localhost
#if you cannot update the defaultDomainNameOfClient with a domainName you must leave
doDomainNameCheckOfClient true. Otherwise, set it to false.
useExternalID=true
defaultRegistries=localhost:4554
defaultServiceName=fileReceiver
defaultDomainNotFoundFolder=../box/domainNotFound
defaultErrorFolder=../box/errorbox
defaultSentFolder=../box/sentbox
tag2domainpath=../box/tag2domain.properties
xpathvalue2domainpropertyFileName=../box/xpathvalue2domain.properties
#database for the SyncrhonizedFileProcess. If you leave it out you are using in Memory
driver=org.h2.Driver
url=jdbc:h2:file:./fileregistrydb
username=sa
password=

1.outbox=../box/outbox
1.fileProcess=com.maikenwinterberg.filedomainjumper.file.StayPutUntilSentNoSentFileProcess
1.routingclass=com.maikenwinterberg.filedomainjumper.router.DocumentWrapperRouter
1.xpath.1=//AccountingSupplierParty/Party/Contact/ElectronicMail/text()
1.xpath.2=//Email/text()
1.keyxpath.1=//SupplierAssignedAccountID/text()
1.xpathvalue2domainpropertyFileName=../box/xpathvalue2domain.properties

#this a a subscription based message sender. make a network of messaging.
#You can turn your inbox into and outbox and become a node in a tree like this:
# 1) Change the outbox param to point to the inbox
# 2) Give your subscription a new serviceName (you can make it an url that point to site that describe your subscription
# 3) change registries to point to your own registry (you need to install a MaikenWinterbergSocketRegistry)
# you you want a closed group of people to your subscription you edit the validDomains.cfg of the router config.
2.outbox=../box/subsriptionOutbox
2.fileProcess=com.maikenwinterberg.filedomainjumper.file.SynchronizedFileProcess
2.routingclass=com.maikenwinterberg.filedomainjumper.router.SocketRegistryRouter
#the subscription registry inform the masses about your subscription (requires the SocketRegistryRouter)
2.subscriptionregistries=localhost:4554
#the registry is where the infor about your listeners live (requires the SocketRegistryRouter)
2.registries=localhost:4554
2.type=socket
2.serviceName=http://localhost/subscription1
```

Wrapper routing

This router uses the following routers in the described order:

Folder router (all documents)
Xpath router (XML)
ProppertyByXPath router (XML)
Tag router (pdf, txt)

Static routing

Static routing select the domainnames from a list of domains.

Eksample of static routing configuration

```
1.outbox=./box/outbox/static
1.routingclass=com.maikenwinterberg.filedomainjumper.router.StaticDocumentRouter
1.domainfile=./box/outbox/staticReceiverDomains.cfg
```

The domainfile contains a list of all domainnames that will receive the files that you but into the configured outbox folder.

Xpath routing

This router is for your XML documents that contain a domainname or an email. You can learn about Xpath on page: https://www.w3schools.com/xml/xpath_intro.asp

Eksample of XPath routing configuration

```
2.outbox=./box/outbox/xpath
2.routingclass=com.maikenwinterberg.filedomainjumper.router.XPathDocumentRouter
2.xpath.1=//AccountingSupplierParty/Party/Contact/ElectronicMail/text()
2.xpath.2=//ReceiverDomainName/text()
2.xpath.3=//receiverdomainname/text()
2.xpath.4=//domainName/text()
2.xpath.5=//domainname/text()
```

The Xpath router converte the files into an xml and looks for domainnames of the document by using path notation. If the node-value or the text-element that is found is an email it will converte the email into a domainname. The router will start with xpath index 1 and continue to the next index if nothing found. If there are no domainnames in your xml this router cannot be used (an email will do). The domainnames found are being validated by the router – only valid domainnames are accepted.

PropertyByXPath routing

Eksample of property by XPath routing configuration

This router is for your XML documents that does not contain a valid domainname or a valid email. Instead it finds a key that is used in the configured propertyfile.

```
3.outbox=./box/outbox/propertyByXPath
3.routingclass=com.maikenwinterberg.filedomainjumper.router.PropertyByXPathDocumentRouter
3.propertyFileName=box/outbox/propertyByXPath.cfg
3.xpath.1=//SupplierAssignedAccountID/text()
```

You specifify the xpath of the key and the location of the property file of wich the domainname is in. The propertyfile contains multiple key=domainname lines.

Folder routing

Eksample of domain folder routing configuration

If you put your files into a folder named after the domain of the receiver you can use this routing. It support a tree structure. Only requirement is that there is a domainfolder in the buttom of the tree.

```
4.outbox=./box/outbox/domainfolders
4.routingclass=com.maikenwinterberg.filedomainjumper.router.DomainFolderRouter
```

SocketRegistry routing

This routing option is for your subscriptions. By selecting this routing option you send your documents to everyone that have subscriped to a serviceName in the registry configurated. If you use this option you should install your own MaikenWinterbergSocketRegistry.

```
4.outbox=./box/outbox/registrylookup
4.routingclass=com.maikenwinterberg.filedomainjumper.router.SocketRegistryRouter
4.registries=maikenwinterberg.com:6666
4.type=socket
4.serviceName=www.subscriptiondomain.com/subscription1
```

File Process

You have 5 options of what to do with the files when processed. You can use the:

com.maikenwinterberg.filedomainjumper.file.SysnhronizedFileProcess
com.maikenwinterberg.filedomainjumper.file.DomainAndTimeFileProcess
com.maikenwinterberg.filedomainjumper.file.DomainFileProcess.
com.maikenwinterberg.filedomainjumper.file.StayPutUntilSentProcess
com.maikenwinterberg.filedomainjumper.file.StayPutUntilSentNoSentFileProcess

You also have the option of creating your own fileprocess by implementing the com.maikenwinterberg.filedomainjumper.file IFileProcess interfrace.

Configuration of the file process

```
4.fileProcesser=com.maikenwinterberg.filedomainjumper.file.SynchronizedFileProcess
```

The number 4 is replaced with the number of your folderIndex that you wish to configure. You select the classname of your choice. The synchronizedFileProcess do not delete files but keep trak of what has been sent. The DoaminAndTimeFileProcess adds a timestamp to the file and insert the file in a domainname folder. The DomainFileProcess does almost like the DomainAndTimeFileProcess but with a timestamp. The StayPutUntilSentFileProcess only moves the file away form the outbox when its sent or if it cannot find the domainname of the receiver.

Maiken Winterberg File Reciever

You configure the FileReceiver in the file conf/FileDomainJumper.properties. When configured you restart the service. On linux you invoke the reInstallMaikenWinterbergFileReceiver.sh file in the service_linux folder (the file will be created when you install the service).

Security

The Maiken Winterberg File Receiver decides what secury implementation to use. The class files used must be in the classpath of both the fileSender and the fileReceiver. I do not wish incompatibility to the network of documents. Therefore, if you create your own security implementation you must send me a copy so I can embed it into the Default Maiken Winterberg File Domain Jumper.

FileReceiver.properties

```
#please replace all localhost variables with the domainname on which your system is installed
bindaddr=0.0.0.0
port=4445
#change localhost to the domainname of the host of your installation
defaultDomainNameOfClient=localhost
acceptIpAsDomainName=true
defaultRegistries=localhost:4554
#if you cannot update the defaultDomainNameOfClient with a domainName you must leave useExternalID true. Otherwise, set it to false.
useExternalID=true
defaultInbox=../box/inbox

1.registration.id=1
1.registration.inboxFolder=../box/inbox
1.registration.type=socket
#must be replaced with your domainname.
1.registration.domainName=localhost
#you should have a least one registration with fileReceiver as a serviceName. Its the default service used for receiveing documents.
#Only if you are subscribing to documents via a registry for subscription you must change the name.
1.registration.serviceName=fileReceiver

2.registration.id=2
2.registration.inboxFolder=../box/inbox/subscription1
2.registration.registries=localhost:4554
2.registration.type=socket
#must be replaced with your domainname.
2.registration.domainName=localhost
```

```
2.registration.serviceName=http://localhost/subscription1

#multi subscription. Give me all subscription in the subscription registries
3.registration.id=3
3.registration.inboxFolder=../box/inbox/multisubscription
3.registration.subscriptionregistries=localhost:4554
3.registration.maxsubscriptions=1000'
```

Default setup

```
#You change the security plugin by adding the following line with your new implementation:
securityImpl=com.maikenwinterberg.socketregistry.security.StaticRegistrySecurity
#You change the max number byte with the following setting:
bindaddr=0.0.0.0
port=6667
#change localhost to the domainname of the host of your installation
defaultDomainNameOfClient=localhost
acceptIpAsDomainName=true
defaultRegistries=maikenwinterberg.com:6666;documentnetwork.com;localhost:6666
```

Setup the inbox

```
1.registration.id=1
1.registration.type=socket
1.registration.maxNumberOfBytes=10737418240
#you should have a least one registration with fileReceiver as a serviceName. Its the default service used for receiveing documents. Only if you are subscribing to documents via a registry for subscription you must change the name.
1.registration.serviceName=fileReceiver
1.registration.inboxFolder=./box/inbox
```

Subscribe

```
2.registration.id=2
#A subscription must include in which registry of which you subscribe.
2.registration.registries=maikenwinterberg.com:6666
2.registration.type=socket
#you should have a least one registration with fileReceiver as a serviceName. Its the default service used for receiveing documents. Only if you are subscribing to documents via a registry for subscription you must change the name.
2.registration.serviceName=www.subscriptiondomain.com/subscription1
2.registration.inboxFolder=./box/subscription1
```