

Parapher depuis Efallia Process

Introduction

Dans l'interaction entre Process et MG9, le fichier lui-même ne transite pas d'un module à l'autre. Il reste dans MG9 et Process récupère les métadonnées permettant de pointer dessus, soit pour l'afficher dans un formulaire, soit pour le modifier en passant par les outils offerts par MG9.

Pour signer depuis Process un document se trouvant dans MG9, la mécanique nécessite pour Process de demander à MG9 d'envoyer le document à signer vers l'un des parapheurs disponibles.

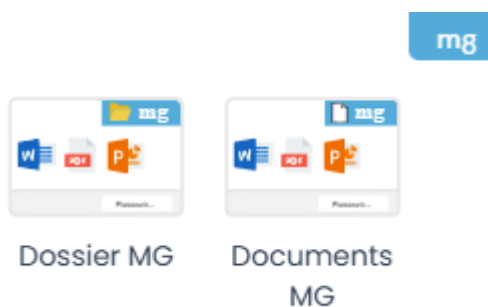
Ils sont actuellement au nombre de 6 :

- Contrat-Privé
- [Fast-Parapheur \(Docaposte\)](#) – conf : [MG9 – Process](#)
- [iParapheur \(Libriciel\)](#) – conf : [MG9 – Process](#)
- [iXparapheur \(iXbus\)](#) – conf : [MG9 – Process](#)
- Sunny stamp (Lex Persona)
- [Xparaph](#) – conf : [MG9 – Process](#)

(NB : depuis Process, n'importe quel parapheur d'MG9 peut être utilisé via le [connecteur générique](#))

Le principe de communication est le suivant :

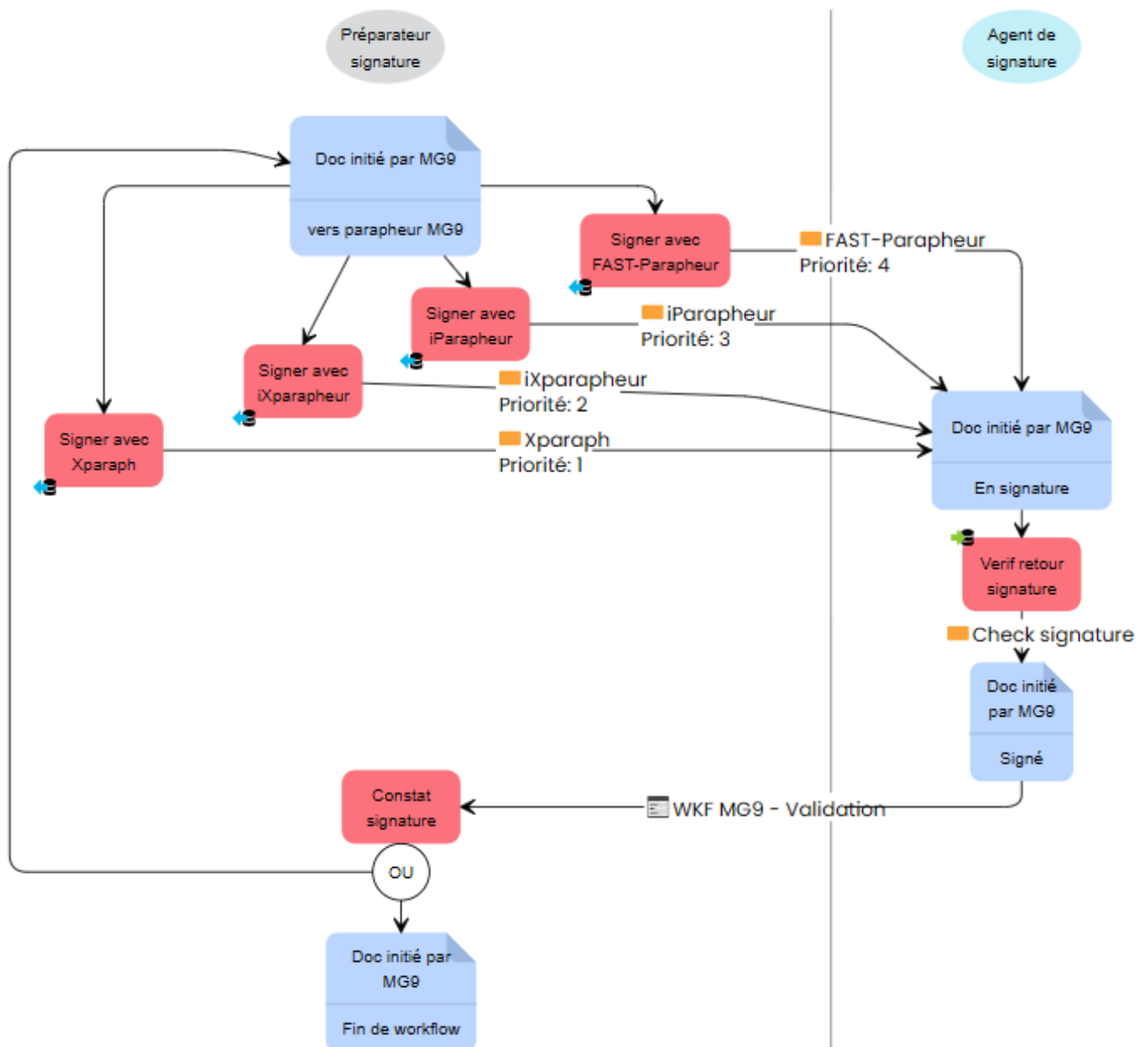
Dans votre formulaire côté Process, vous disposez du document MG9 via un composant MG documents ou MG dossier :



Quel que soit le parapheur que vous voulez utiliser, vous devez utiliser 2 connecteurs et un agent.

Le premier connecteur donne l'ordre à MG9 d'envoyer le fichier à signer et le second, utilisé par un agent, vérifie à intervalles réguliers si la signature a été effectuée.

Exemple de workflow (l'instance de workflow "Doc initié par MG9" contient un champ Dossier MG ou un champ Documents MG recensant les fichiers stockés sur MG9 qu'il faut envoyer vers l'un des parapheurs) :



Vous avez le choix entre différents connecteurs d'envoi en signature (certains spécifiques à un parapheur donné pour faciliter la modélisation) mais il n'y a qu'un seul connecteur qui vérifie l'état d'exécution de la signature.

Dans le détail, côté Process, vous disposez de différents connecteurs dont un générique qui se calque sur un contrat d'interface normalisé avec MG9.

Le contrat d'interface demande

- Un compte de parapheur à utiliser (il est configuré côté MG9)
- Un "custom data" json qui permet de spécifier les détails concernant chaque parapheur

Les connecteurs spécifiques permettent de faciliter la modélisation et sont détaillés plus bas.

Une fois l'opération portant le connecteur atteinte, Process envoie à MG9 l'ordre d'envoyer son fichier au parapheur désigné.

MG9 s'adresse alors au parapheur en question en lui envoyant le fichier à

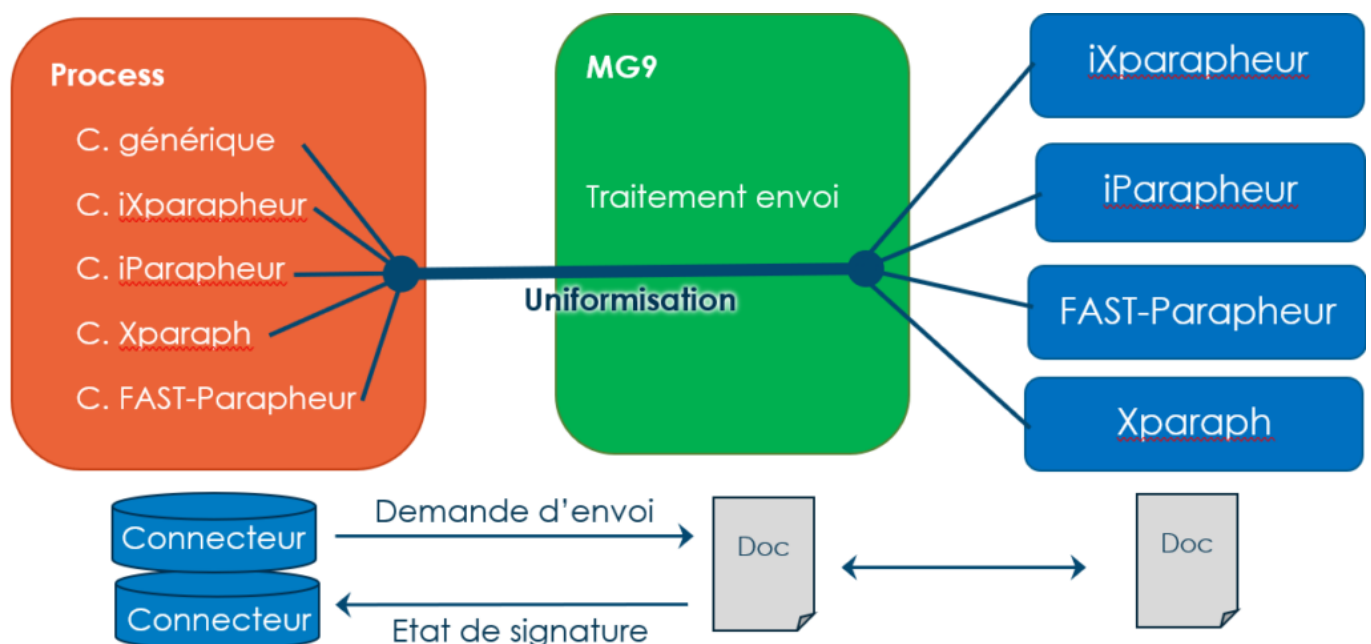
signer, les éventuelles annexes voulues ainsi que les métadonnées nécessaires.

Il attend ensuite que le fichier soit signé en interrogeant le parapheur à intervalles réguliers.

Pendant ce temps, Process appelle MG9 de la même façon pour savoir si la signature est effectuée.

Une fois le document signé dans le parapheur, MG9 en est informé et le récupère pour le stocker en version signée.

A l'itération suivante de l'agent côté Process, ce dernier récupère une information concernant l'action effectuée ; un document ayant terminé son processus de signature peut être signé ou s'être vu opposé un refus de signature, on récupère ces deux informations côté Process dans deux variables différentes en sortie du connecteur de vérification de signature. Cela permet de faire évoluer sa modélisation en fonction de si le signataire a validé le document ou s'il a refusé de le signer.



Connecteur de retour de signature universel

L'utilisation de parapheur externe étant asynchrone, côté Process il faudra utiliser un connecteur universel pour vérifier l'état d'avancement de la signature (le plus simple étant via un agent de signature).

Classe : **com.clog.workey.connectors.v2.MGParapherCheckDocumentStatus**

Ce connecteur nécessite 2 champs en entrée et 2 en sortie :

En entrée :

1. Un champ texte monovalué indiquant le compte parapheur utilisé lors de l'envoi et dont on cherche à connaître l'avancement
2. Composant MGFolder ou MGDdoc contenant le document envoyé à signer (celui

utilisé pour l'envoi)

En sortie :

1. Liste multivaluée des états des différentes signatures ("0": en cours, "1": terminé)
2. Liste multivaluée de la validation ou non des documents ("0": refusé, "1": validé)

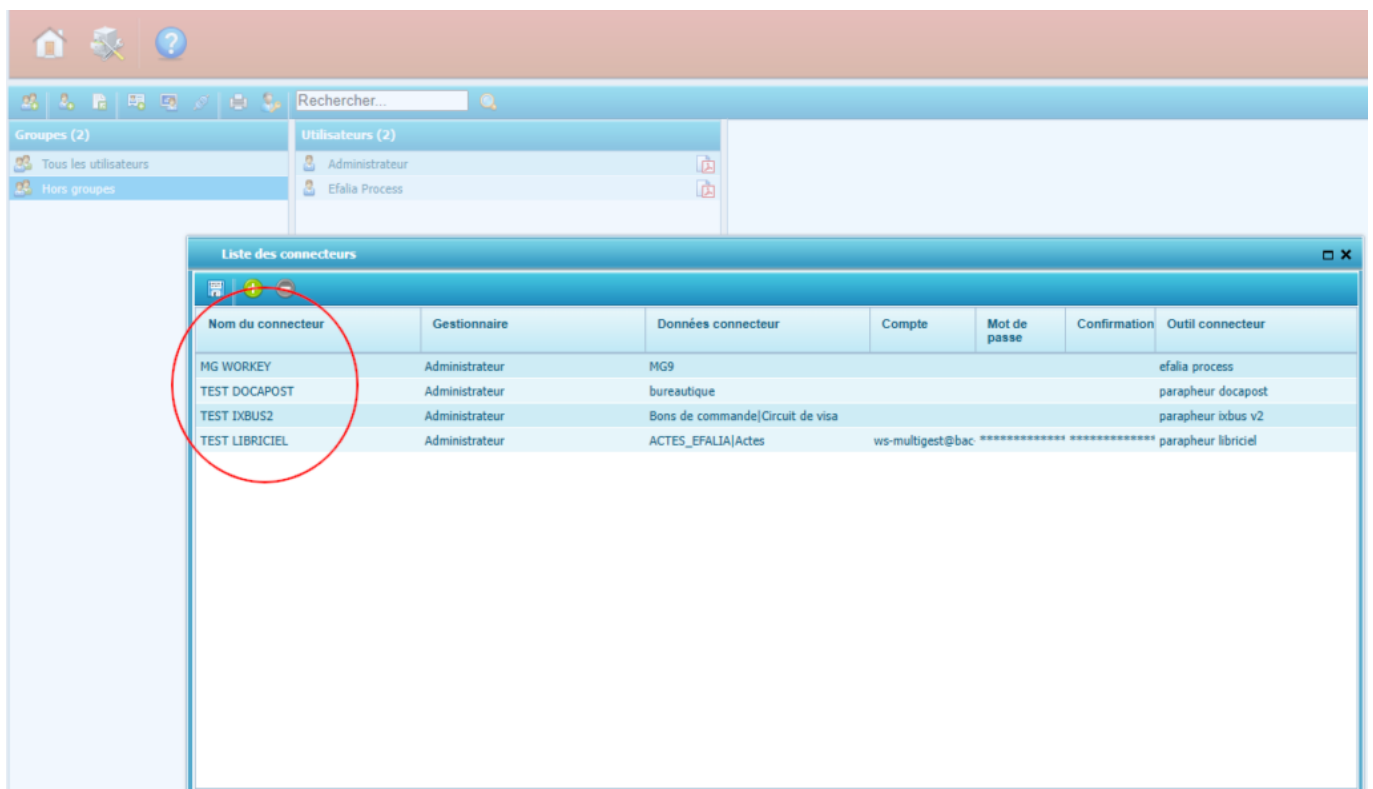
i.e. : Si une signature est terminée (premier champ) vous pouvez vérifier si le traitement a conduit à une validation du document ou un refus de signature (deuxième champ) et utiliser cette information dans votre workflow.

Connecteur d'envoi générique

MG9 – connecteur générique d'envoi vers un parapheur quelconque

Dépend du parapheur voulu. Voir dans les autres catégories pour les parapheurs déjà connus.

Pour utiliser un parapheur quelconque depuis Process, on a besoin de son nom :



Nom du connecteur	Gestionnaire	Données connecteur	Compte	Mot de passe	Confirmation	Outil connecteur
MG WORKEY	Administrateur	MG9				efalia process
TEST DOCAPOST	Administrateur	bureautique				parapheur docapost
TEST IxBUS2	Administrateur	Bons de commande Circuit de visa				parapheur ibbus v2
TEST LIBRICIEL	Administrateur	ACTES_EFALIA Actes	ws-multigest@bac	*****	*****	parapheur libriciel

Process – connecteur générique d’envoi vers un parapheur quelconque

Classe : `com.clog.workey.connectors.v2.MGParapherGenericSendDocument`

Le connecteur nécessite 3 composants au minimum en entrée (si on n’a pas d’annexes à joindre au document à signer) et un composant en sortie.

En entrée :

1. Le compte parapheur à utiliser (par exemple, “MonParapheurInconnu”) (champ texte monovalué). C’est le nom du parapheur [défini côté MG9](#).
2. Configuration json du custom data attendu par MG9 qui dépend du parapheur. (champ texte monovalué)
 - Exemple :

```
{  
  "Circuit_de_signature": "Circuit_2024",  
  "Metadonnees": {  
    "Cle": "valeur de la clé",  
    "Compte_systeme": "information quelconque"  
  }  
}
```
3. Composant MGFolder ou MGDdoc contenant le document à signer. (champ monovalué)
4. Composant MGFolder ou MGDdoc contenant des annexes à inclure dans le parapheur (optionnel, champ multivalué)

En sortie :

1. Id du dossier parapheur créé depuis MG9 (a priori, inutile dans Process, mais peut avoir un intérêt pour le debug) (champ texte monovalué)

Connecteur d’envoi – Fast-Parapheur (de Docapost)

NB : Les connecteurs de signature s’utilisent par paire. Un connecteur d’envoi en signature et un connecteur de retour de signature. Ce dernier est toujours le [connecteur générique de retour](#).

Fast-Parapheur

Configurer les circuits à accéder.

MG9 – Fast-Parapheur

Pour utiliser un parapheur depuis Process, il faut configurer un connecteur dans MG9.

La configuration du connecteur est réalisée dans le fichier
[REP_INSTALL_MULTIGEST]\bin\docapost\config.ini

Exemple de fichier de configuration :

```
[SERVICE]
ENDPOINT=https://demo-parapheur.dfast.fr/parapheur-soap/soap/v1/Documents
CLIENTCERT=FastAgentEfallia
CERTPASS=xxxxxxx
SIREN=xxxxxxxxxxx
REQUESTMODE=0
PROXYHOST=
PROXYPORT=
PROXYLOGIN=
PROXYPWD=
SSLVERIFYPEER=1
TRACE=1
EXCEPTIONS=0
TIMEOUT=600
DELETEDFOLDER=0

[NAMESPACE]
SOAPENV=http://schemas.xmlsoap.org/soap/envelope/
NS=http://sei.ws.fast.cdc.com/
XM=http://www.w3.org/2005/05/xmlmime

[TRACE]
LEVEL=7
LOGS=D:\multigest\bin\docapost\logs
```

Détail du contenu de la section [SERVICE] :

Propriété	Description
ENDPOINT	Url du service web
CLIENTCERT	Nom du certificat à utiliser pour la connexion aux services web. Le certificat est fourni par Docapost et est à stocker dans le dossier certificats du dossier d'installation du connecteur
CERTPASS	Mot de passe du certificat.
SIREN	Numéro siren du compte client Docapost.

REQUESTMODE	Définit le mode d'envoi des requêtes HTTP. 0 = mode SOAP ,1 = mode CURL.(Mode CURL déprécié, par défaut = 0).
PROXYHOST	Nom du serveur Proxy si actif pour l'accès au service web (facultatif)
PROXYPORT	Port du serveur Proxy si actif pour l'accès au service web (facultatif)
PROXYLOGIN	Login d'authentification au serveur Proxy si actif pour l'accès au service web (facultatif)
PROXYPWD	Mot de passe d'authentification au serveur Proxy si actif pour l'accès au service web (facultatif)
SSLVERIFYPEER	Active/Désactive(1 ou 0) la vérification du certificat si l'accès au service web est en HTTPS (facultatif, par défaut = 1)
TRACE	Active/Désactive(1 ou 0) la capture les informations de requête et de réponse SOAP si REQUESTMODE = 0. (facultatif, par défaut = 0)
EXCEPTIONS	Active/Désactive(1 ou 0) la génération d'exceptions de type SoapFault. (déprécié, par défaut = 0)
TIMEOUT	Délai maximum en secondes d'une connexion aux services web (facultatif, par défaut = 600)
DELETEFOLDER	Active/Désactive(1 ou 0) la suppression du dossier parapheur après retour dans Multigest.Par défaut les dossiers sont archivés (facultatif, par défaut = 0)

Détail du contenu de la section [NAMESPACE] :

Propriété Description

SOAPENV	Url de définition des espaces de nom du schéma soapenv. Par défaut http://schemas.xmlsoap.org/soap/envelope/
NS	Url de définition des espaces de nom de l'API SOAP Docapost.
XM	Url de définition des espaces de nom du schéma xmime. Par défaut http://www.w3.org/2005/05/xmlmime

Détail du contenu de la section [DEBUG] :

Propriété Description

LEVEL	Définit le niveau de logs et tracers par la combinaison des valeurs présentées ci-dessous
LOGS	Chemin du dossier de stockage des logs et tracers

NB : Valeurs combinables pour l'option LEVEL

1 = Active l'enregistrement du log des erreurs dans le fichier errors-log
xxxx-xx-xx xxh.log

2 = Active l'enregistrement du log des requêtes http des services web dans le fichier trace-log xxxx-xx-xx xxh.log (Nécessite TRACE = 1)

4 = Active l'enregistrement du log des réponses http des services web dans le fichier trace-log xxxx-xx-xx xxh.log (Nécessite TRACE = 1)

Exemple : 7 active tous les logs , 6 active les logs des requêtes et réponses http

Dans l'administration des connecteurs, créez un connecteur pour le parapheur

et donnez lui un nom à utiliser côté Process.

Pour Docapost, il n'y a besoin que de renseigner un gestionnaire, mais vous pouvez renseigner un circuit à appeler

Process – Fast-Parapheur

Envoi en signature :

Pour l'heure, il n'y a pas de connecteur spécifique pour envoyer un document à signer vers Fast-Parapheur. Il faut utiliser le [connecteur générique](#) avec la configuration json suivante :

```
{
  "circuitId":"Définit le circuit de signature, à récupérer côté Fast-Parapheur",
  "label":"Le libellé (sous-dossier) associé au document",
  "DossierID":"commentaire associé au document",
  "email_destinataire":"email de la personne à qui l'on souhaite envoyer le document après sa signature"
}
```

Retour de signature :

Configurer un agent qui utilise [le connecteur de retour universel](#) pour interroger MG9 de façon récurrente jusqu'à ce que la signature soit opérée. [Voir le processus d'exemple.](#)

Connecteur d'envoi – iParapheur (de Libriciel sans Pastel)

NB : Les connecteurs de signature s'utilisent par paire. Un connecteur d'envoi en signature et un connecteur de retour de signature. Ce dernier est toujours le [connecteur générique de retour](#).

iParapheur

Configurer les circuits à accéder.
(Détail à venir)

Il faut au moins un TypeTechnique et un SousType.

Si un script est défini côté iParapheur pour déterminer le circuit, il faut aussi connaître les métadonnées utilisées dans ce script et qu'il faudra envoyer lors de la signature.

MG9 – iParapheur

Pour utiliser un parapheur depuis Process, il faut configurer un connecteur dans MG9.

La configuration du connecteur est réalisée dans le fichier
[REP_INSTALL_MULTIGEST]\bin\iparapheur\config.ini

Exemple de fichier de configuration :

```
[SERVICE]
ENDPOINT=https://secure-iparapheur.partenaire.libriciel.fr/ws-iparapheur-no-m
tom
ENDPOINT_MTOM= /* Laisser vide, ce n'est plus utilisé
AUTHSERVICE=5 /* Entre 1, 2, 4 et 5 => 5 : authent HTTP + HTTPS (1+4)
LOGIN=ws-multigest@bac-46
PASSWORD=xxxxxxxxxxxxx
CLIENTCERT=
CACERT=
CERTPASS=
AUTHTYPE=0
REQUESTMODE=0
PROXYHOST=
PROXYPORT=
PROXYLOGIN=
PROXYPWD=
SSLVERIFYPEER=1 /* 1 : si le HTTPS le nécessite,
TRACE=1
EXCEPTIONS=1
TIMEOUT=600
DELETEFOLDER=0 /* 1 : si on ne veut pas simplement archiver
IPTYPE= /* Défini le circuit de paraphe. Préférentiellement à renseigner
directement depuis Process
IPSOUSTYPE= /* Défini le circuit de paraphe. Préférentiellement à renseigner
directement depuis Process
USEMTOM=0
MULTISESSION=1
MAXRUNNINGJOB=200

[NAMESPACE]
SOAPENV=http://schemas.xmlsoap.org/soap/envelope/
NS=http://www.adullact.org/spring-ws/iparapheur/1.0
XM=http://www.w3.org/2005/05/xmlmime

[TRACE]
LEVEL=7
LOGS=D:\multigest\bin\iparapheur\logs
```

Détail du contenu de la section [SERVICE] :

Propriété	Description
ENDPOINT	Url du service web
ENDPOINT_MTOM	Url du service web en mode MTOM.(déprécié)

AUTHSERVICE	Type d'authentification pour la connexion aux services web. La valeur correspond à la combinaison des valeurs suivantes : 1 = Authentification HTTP, 2 = Authentification forte par certificats (déprécié), 4 = mode HTTPS.
LOGIN	Login par défaut d'authentification pour la connexion aux services web
PASSWORD	Mot de passe du login par défaut
CLIENTCERT	Nom du certificat client à utiliser pour la connexion aux services web si authentification forte demandée. Le certificat est fourni par Libriciel et est à stocker dans le dossier d'installation du connecteur
CACERT	Nom du certificat d'autorité de certification à utiliser pour la connexion aux services web si authentification forte demandée. Le certificat est à stocker dans le dossier d'installation du connecteur
CERTPASS	Mot de passe du certificat client
AUTHTYPE	Type d'authentification HTTP si l'authentification HTTP est activée. 1 = BASIC, 2 = DIGEST (facultatif, par défaut = 1)
REQUESTMODE	Définit le mode d'envoi des requêtes HTTP. 0 = mode SOAP , 1 = mode CURL. (Mode CURL déprécié, par défaut = 0).
PROXYHOST	Nom du serveur Proxy si actif pour l'accès au service web (facultatif)
PROXYPORT	Port du serveur Proxy si actif pour l'accès au service web (facultatif)
PROXYLOGIN	Login d'authentification au serveur Proxy si actif pour l'accès au service web (facultatif)
PROXYPWD	Mot de passe d'authentification au serveur Proxy si actif pour l'accès au service web (facultatif)
SSLVERIFYPEER	Active/Désactive(1 ou 0) la vérification du certificat si l'accès au service web est en HTTPS (facultatif, par défaut = 1)
TRACE	Active/Désactive(1 ou 0) la capture les informations de requête et de réponse SOAP si REQUESTMODE = 0. (facultatif, par défaut = 0)
EXCEPTIONS	Active/Désactive(1 ou 0) la génération d'exceptions de type SoapFault. (déprécié, par défaut = 0)
TIMEOUT	Délai maximum en secondes d'une connexion aux services web (facultatif, par défaut = 600)
DELETEDFOLDER	Active/Désactive(1 ou 0) la suppression du dossier parapheur après retour dans Multigest. Par défaut les dossiers sont archivés (facultatif, par défaut = 0)
IPTYPE	Type technique utilisé par défaut pour la création du dossier
IPSOUSTYPE	Sous type technique utilisé par défaut pour la création du dossier
MULTISESSION	Active/Désactive(1 ou 0) la récupération des dossiers avec le compte paramétré dans l'administration des connecteur. Par défaut le compte utilisé est celui défini dans le fichier config.ini. (facultatif, par défaut = 0)
MAXRUNNINGJOB	Définit le nombres maximum de jobs (dossiers iparapheur) à traiter par itération du service automate webserveur.

Détail du contenu de la section [NAMESPACE] :

Propriété Description

SOAPENV	Url de définition des espaces de nom du schéma soapenv. Par défaut http://schemas.xmlsoap.org/soap/envelope/
NS	Url de définition des espaces de nom de l'API SOAP Libriciel.
XM	Url de définition des espaces de nom du schéma xmime. Par défaut http://www.w3.org/2005/05/xmlmime

Détail du contenu de la section [DEBUG] :

Propriété Description

LEVEL	Définit le niveau de logs et tracers par la combinaison des valeurs présentées ci-dessous
LOGS	Chemin du dossier de stockage des logs et tracers
NB : Valeurs combinables pour l'option LEVEL	
1 = Active l'enregistrement du log des erreurs dans le fichier errors-log xxxx-xx-xx xxh.log	
2 = Active l'enregistrement du log des requêtes http des services web dans le fichier trace-log xxxx-xx-xx xxh.log (Nécessite TRACE = 1)	
4 = Active l'enregistrement du log des réponses http des services web dans le fichier trace-log xxxx-xx-xx xxh.log (Nécessite TRACE = 1)	
Exemple : 7 active tous les logs , 6 active les logs des requêtes et réponses http	

Dans l'administration des connecteurs, créez un connecteur pour le parapheur et donnez lui un nom à utiliser côté Process.

Pour iParapheur, il n'y a besoin que de renseigner un gestionnaire

Process – iParapheur

Envoi en signature :

Pour l'heure, il n'y a pas de connecteur spécifique pour envoyer un document à signer vers iParapheur. Il faut utiliser le [connecteur générique](#) avec la configuration json suivante :

```
{
  "TypeTechnique": "Définit le circuit de signature, à récupérer côté iParapheur : IPTYPE",
  "SousType": "Définit le circuit de signature, à récupérer côté iParapheur : IPSOUSTYPE",
  "DossierID": "Id unique, si non défini : est généré côté MG",
  "EmailEmmetteur": "Courriel de l'émetteur utilisé pour l'envoi de notifications (si quelqu'un doit suivre).",
  "DateLimite": "Date limite du dossier. Date au format YYYY-MM-DD - facultatif",
  "Visibilité": "Visibilité du dossier. Valeurs disponibles : PUBLIC (par défaut) ou CONFIDENTIEL",
  "AnnotationPublique": "information à fournir côté parapheur de façon
```

```
publique - facultatif",
  "AnnotationPrivee": "information à fournir côté parapheur de façon privée-
facultatif",
  "metadatas": {
    "clé1": "val1",
    "clé2": "val2"
  }
}
```

Retour de signature :

Configurer un agent qui utilise [le connecteur de retour universel](#) pour interroger MG9 de façon récurrente jusqu'à ce que la signature soit opérée. [Voir le processus d'exemple.](#)

Connecteur d'envoi – iXparapheur (iXbus)

NB : Les connecteurs de signature s'utilisent par paire. Un connecteur d'envoi en signature et un connecteur de retour de signature. Ce dernier est toujours le [connecteur générique de retour](#).

iXparapheur (iXbus)

Il existe deux possibilités pour utiliser iXparapheur

- Créer des circuits de signatures dans iXbus et les utiliser depuis Process
- Créer des circuits custom à la volée depuis Process et les faire appliquer dans iXbus

Dans tous les cas, il faudra que les signataires et valideurs soient identifiés dans iXbus et appelables depuis Process (qu'ils soient utilisateurs des deux solutions ou seulement de iXbus).

MG9 – iXparapheur (iXbus)

Pour utiliser un parapheur depuis Process, il faut configurer un connecteur dans MG9.

La configuration du connecteur est réalisée dans le fichier **[REP_INSTALL_MULTIGEST]\bin\ixbus\config2.ini**

Exemple de fichier de configuration :

```
[SERVICE]
```

BASEURI=https://demodemat.ixbus.net
URLAPI=api/parapheur/v1
TOKENNAME=IXBUS_API
TOKEN=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx
LOGIN=EFALIA_Api
NATURE=Bons de commande
HTTPLOGIN=
HTTPPASSWORD=
AUTHTYPE=
PROXYHOST=
PROXYPORT=
PROXYLOGIN=
PROXYPWD=
SSLVERIFYPEER=1
USEPWDTOKEN=0
SENDCREATOR=1
CREATORCOLUMN=MAIL
KEYWORD=
ADDCOMMENT=1
DELETEDFOLDER=1
TIMEOUT=600

[TRACE]
LEVEL=7
LOGS=D:\multigest\bin\ixbus\logs

Détail du contenu de la section [SERVICE] :

Propriété	Description
BASEURI	Url racine du site web hébergeant l'application iXParapheur
URLAPI	Url de l'API REST (facultatif, par défaut = api/parapheur/v1)
TOKENNAME	Nom de la clé API (Token) envoyé dans les entêtes HTTP (facultatif, par défaut = IXBUS_API)
TOKEN	Clé de l'API (également appelée token API) envoyée à chaque requête HTTP afin d'accéder aux fonctionnalités de l'API. (récupérée depuis iXbus)
USEPWDTOKEN	Active/Désactive (1 ou 0) l'utilisation d'un TOKEN alternatif définit dans la colonne mot de passe de l'administration des connecteurs (déprécié , par défaut = 0)
PROXYHOST	Nom du serveur Proxy si actif pour l'accès au service web (facultatif)
PROXYPORT	Port du serveur Proxy si actif pour l'accès au service web (facultatif)
PROXYLOGIN	Login d'authentification au serveur Proxy si actif pour l'accès au service web (facultatif)
PROXYPWD	Mot de passe d'authentification au serveur Proxy si actif pour l'accès au service web (facultatif)
SSLVERIFYPEER	Active/Désactive (1 ou 0) la vérification du certificat si l'accès au service web est en HTTPS (facultatif)

TIMEOUT	Délai maximum en secondes d'une connexion aux services web (facultatif, par défaut = 600)
DELETEFOLDER	Active/Désactive(1 ou 0) la suppression du dossier parapheur après retour dans Multigest.Par défaut les dossiers sont archivés (facultatif, par défaut = 0)
LOGIN	Rédacteur du dossier par défaut. Nécessite SENDCREATOR = 0
NATURE	Nature du circuit par défaut.
SENDCREATOR	Active l'envoi dynamique d'un redacteur en fonction des données de la GED.0 = Désactivé, 1 = Créateur du processus, 2 = Acteur précédent, 3 = Créateur du document
CREATORCOLUMN	Définit la donnée utilisateur à transmettre pour le redacteur si SENDCREATOR > 0. LOGIN = login multigest, LOGIN_WINDOWS= login windows multigest, MAIL = email multigest
ADDCOMMENT	Active/Désactive (1 ou 0) l'ajout du commentaire workflow aux mots clés envoyés à la création du dossier (facultatif, par défaut = 0)

Description de la section [DEBUG]

Propriété Description

LEVEL	Définit le niveau de logs et tracers par la combinaison des valeurs présentées ci-dessous
LOGS	Chemin du dossier de stockage des logs et tracers
NB : Valeurs combinables pour l'option LEVEL	
1 = Active l'enregistrement du log des erreurs dans le fichier errors-log xxxx-xx-xx xxh.log	
2 = Active l'enregistrement du log des requêtes http des services web dans le fichier trace-log xxxx-xx-xx xxh.log	
4 = Active l'enregistrement du log des réponses http des services web dans le fichier trace-log xxxx-xx-xx xxh.log	
Exemple : 7 active tous les logs , 6 active les logs des requêtes et réponses http	

Dans l'administration des connecteurs, créez un connecteur pour le parapheur et donnez lui un nom à utiliser côté Process.

Pour iXparapheur, il n'y a besoin que de renseigner un gestionnaire. Vous pouvez cependant renseigner un circuit si vous ne voulez pas le spécifier côté Process.

(NB : Les circuits désignés dans Process prendront le pas sur ceux désignés dans l'interface des connecteurs MG9 qui prendront eux-mêmes le pas sur ceux définis dans le config.ini détaillé ci-dessus)

Process – iXparapheur (iXbus)

Classe : `com.clog.workey.connectors.v2.MGParapherIXParapheurSendDocument`

Depuis Process, il existe 2 méthodes pour signer via iXparapheur :

- [Depuis Process, spécifier les circuits définis dans iXparapheur et MG9 fait passe-plat](#)

- Créer des circuits personnalisés directement dans Process et les faire jouer côté iXparapheur (pas de définition de circuit que ce soit dans iXparapheur ou dans MG9)

Utilisation d'un circuit défini dans iXparapheur

En entrée :

1. Le compte parapheur à utiliser (par exemple, "MonNomDeConnecteurIXbusDansMG9") (champ texte monovalué). C'est le nom du parapheur [défini côté MG9](#).
2. Champ texte monovalué contenant "circuit"
3. Champ texte monovalué contenant le nom du circuit iXparapheur à utiliser
4. Composant MGFoldre ou MGDre contenant le document à signer. (champ monovalué)
5. Composant MGFoldre ou MGDre contenant des annexes à inclure dans le parapheur (optionnel, champ multivalué)

En sortie :

1. Id du dossier parapheur créé depuis MG9 (a priori, inutile dans Process, mais peut avoir un intérêt pour le debug) (champ texte monovalué)

Création d'un circuit custom à utiliser dans iXparapheur

En entrée :

1. Le compte parapheur à utiliser (par exemple, "MonNomDeConnecteurIXbusDansMG9") (champ texte monovalué). C'est le nom du parapheur [défini côté MG9](#).
2. Champ texte multivalué détaillant la succession de valideurs et signataires iXparapheur
Il contient une suite de "Visa" et "Signature". Ex: "Visa; Visa; Signature; Signature; Signature"
3. Champ texte multivalué contenant les identifiants des utilisateurs iXparapheur devant viser ou signer, suivant l'ordre établi dans le second champ (les deux champs contiennent donc le même nombre d'éléments).
4. Composant MGFoldre ou MGDre contenant le document à signer. (champ monovalué)
5. Composant MGFoldre ou MGDre contenant des annexes à inclure dans le parapheur (optionnel, champ multivalué)

En sortie :

1. Id du dossier parapheur créé depuis MG9 (a priori, inutile dans Process, mais peut avoir un intérêt pour le debug) (champ texte monovalué)

Connecteur d'envoi – Xparaph (SPL-Xdemat)

NB : Les connecteurs de signature s'utilisent par paire. Un connecteur d'envoi en signature et un connecteur de retour de signature. Ce dernier est toujours le [connecteur générique de retour](#).

Xparaph

Comme pour iXparapheur, il est possible de définir l'identité du signataire dans Xparaph directement depuis Process.

MG9 – Xparaph

Pour utiliser un parapheur depuis Process, il faut configurer un connecteur dans MG9.

La configuration du connecteur est réalisée dans le fichier **[REP_INSTALL_MULTIGEST]\bin\xparaph\config.ini**

Exemple de fichier de configuration :

Attention

En construction

[SERVICE]

TIMEOUT=600

[TRACE]

LEVEL=7

LOGS=D:\multigest\bin\ixbus\logs

Détail du contenu de la section [SERVICE] :

Propriété Description

Description de la section [DEBUG]

Propriété Description

LEVEL Définit le niveau de logs et tracers par la combinaison des valeurs présentées ci-dessous

LOGS Chemin du dossier de stockage des logs et tracers

NB : Valeurs combinables pour l'option LEVEL

1 = Active l'enregistrement du log des erreurs dans le fichier errors-log
xxxx-xx-xx xxh.log

2 = Active l'enregistrement du log des requêtes http des services web dans le
fichier trace-log xxxx-xx-xx xxh.log

4 = Active l'enregistrement du log des réponses http des services web
dans le fichier trace-log xxxx-xx-xx xxh.log

Exemple : 7 active tous les logs , 6 active les logs des requêtes et réponses
http

Dans l'administration des connecteurs, créez un connecteur pour le parapheur
et donnez lui un nom à utiliser côté Process.

Pour Xparaph, il n'y a besoin que de renseigner un gestionnaire.

Process – Xparaph

Classe : **com.clog.workey.connectors.v2.MGParapherXparaphSendDocument**

Depuis Process, il faut définir le circuit à l'aide des champs en entrée :

1. Le compte parapheur à utiliser (par exemple, "MonNomDeConnecteurXparaphDansMG9") (champ texte monovalué). C'est le nom du parapheur [défini côté MG9](#).
2. Champ texte monovalué désignant le signataire dans Xparaph (ex : "CG55023-34")
3. Champ texte monovalué pour spécifier l'identité du déposant
4. Champ texte monovalué pour désigner le contexte de Xparaph à utiliser (valeurs possibles : FON(défaut), PER, SPH, DIR, DLP, EXE)
5. Composant MGFolder ou MGDdoc contenant le document à signer. (champ monovalué)
6. Composant MGFolder ou MGDdoc contenant des annexes à inclure dans le parapheur (optionnel, champ multivalué)

En sortie :

1. Id du dossier parapheur créé depuis MG9 (a priori, inutile dans Process, mais peut avoir un intérêt pour le debug) (champ texte monovalué)