



Diginove

Dexelia Web

Numérisation en client léger

RESUME

Présentation de Dexelia Web

DIGINOVE, Espace Beauvalle Bât B
4 rue du Mahatma Gandhi
13090 Aix en Provence
Tél : +33 (0)4 42 69 09 93
contact@diginove.com

TABLE DES MATIERES

1	Contexte	4
2	Mono-document	4
3	Multi-Documents.....	6
4	Eléments techniques	8
	Environnement technique.....	8
	SDK de développement.....	9
	4.1.1 Principe de fonctionnement.....	9
	4.1.2 Mode d'échange avec ScanServer.....	12

TABLE DES FIGURES

Figure 1 : Écran de configuration du scanner.....	5
Figure 2 : Écran de numérisation mono-document	6
Figure 3 : Écran multi documents.....	7
Figure 4 : Principe de fonctionnement monoposte	9
Figure 5 : Configuration du partage de scanner.....	10
Figure 6 : Principe de fonctionnement en mode partage de scanner.....	11

1 CONTEXTE

Dexelia-Scan numérise les documents depuis une **application autonome** ou bien intégré aux **applications métiers**, que ce soit sur **PC**, sur **tablette** ou dans un navigateur **Web**. Dexelia-Scan est compatible avec de nombreuses marques de **scanners bas, moyens et hauts volumes** (ex : Canon, Fujitsu, Kodak, ...).

L'application permet également de réaliser des **traitements** avancés sur les images numérisées ainsi qu'une **indexation** du contenu. Elle intègre :

- **Simplicité** d'utilisation : seules les fonctions nécessaires sont présentées aux opérateurs. Les profils de numérisation sont adaptés au métier : CNI, facture, bulletins de salaire, ...
- **Déploiement** sur de nombreux environnements : client léger, TSE ou Citrix
- De nombreux traitements d'image sont disponibles : codes à barre 1D et 2D, OCR pleine page ou zoné, détourage, redressement, sens de lecture, découpage, binarisation, filtrage, ...
- **Compression** : la technologie Pix2Pdf (ex. : 59 Ko au lieu de 1050 Ko pour un document A4 couleur 300 dpi)
- **Multi-format** de sortie : TIFF noir & blanc G4, JPEG, PDF, PDF/A, avec intégration du texte et des métadonnées
- **Communication** avec les applications de GED, CRM, CMS par CMIS, Web Services, Servlet, ...

L'intégration du module client léger à une application métier peut être réalisée selon deux modes de fonctionnement distincts :

- Mode **mono-document** : l'application lance la numérisation successivement pour chaque document.
- Mode **multi-documents** : l'application métier définit la liste des documents à numériser et active la numérisation sur cette liste.

Ces deux modes de fonctionnement sont décrits ici. L'intégration dans une application est réalisée au travers du SDK Dexelia Scan Web, composé d'un module qui permet d'intégrer rapidement et facilement un processus de numérisation dans une application métier basée sur une architecture Web (Client Léger).

Ce composant s'intègre directement dans une page Web (dans la page ou dans une fenêtre modale) et fournit d'une part, l'interface utilisateur pour afficher, contrôler et valider les images numérisées et d'autre part, les traitements nécessaires pour produire et transférer les documents dans l'application hôte.

Ce module pilote les scanners dotés d'un driver Twain et intègre des traitements d'images pour améliorer les images et produire les fichiers à intégrer dans l'application métier : GEIDE, ERP, CRM, ...

2 MONO-DOCUMENT

La numérisation est paramétrée et lancée pour un document unique. Le paramétrage concerne le mode de numérisation (profil de numérisation), les traitements d'image réalisés (lors de la numérisation ou avant publication) et le mode de publication (type de fichier, protocole d'échange).

L'interface du composant de numérisation Dexelia Scan Web est composée de 2 onglets.

L'onglet [Configuration] liste les scanners installés sur le poste de travail. Le dernier scanner précédemment utilisé est automatiquement présélectionné.

L'utilisateur a la possibilité de rafraichir la liste de scanners disponibles (bouton [Recharger]), si le scanner n'était par exemple pas en fonction au lancement, et sélectionner un des scanners disponibles sur le poste (bouton [Connecter]).

En fonction des paramètres de configuration, l'utilisateur peut changer ou pas certains éléments du profil de numérisation (profondeur de l'image, résolution, faces numérisées ...)

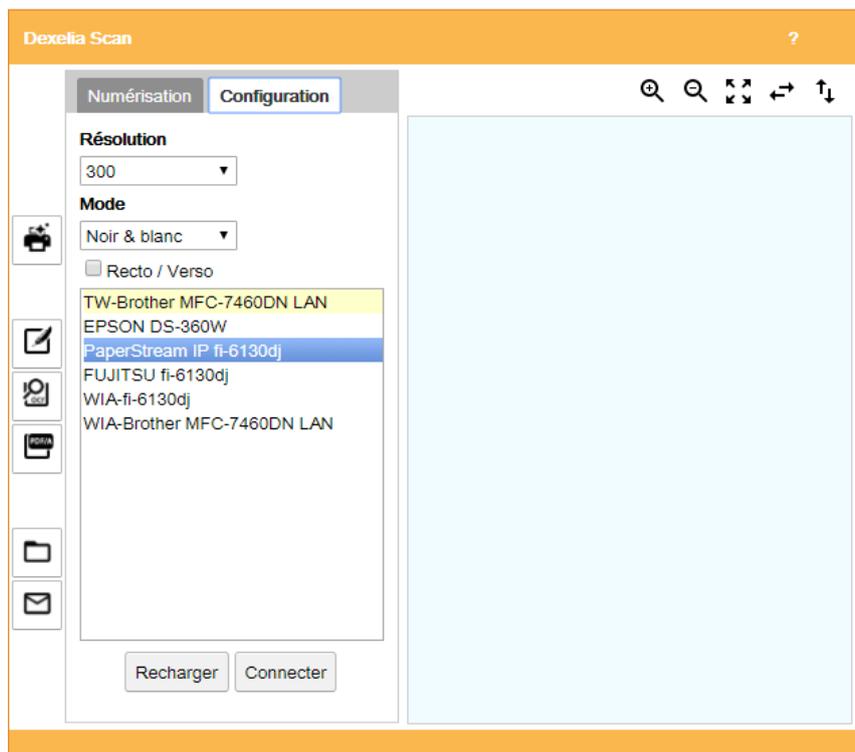


Figure 1 : Écran de configuration du scanner

L'onglet [Configuration] est présélectionné lors de l'affichage initial si on n'a pas pu se connecter au scanner défini par défaut. Si la connexion s'est correctement réalisée, l'onglet [Numérisation] est directement affiché.

L'onglet [Numérisation] permet de piloter le processus de numérisation de façon simple et efficace :

- Le bouton  déclenche la numérisation des pages.
- Le bouton  permet d'insérer des fichiers image depuis un répertoire du disque utilisateur. L'insertion peut également être faite par « glisser / déposer » depuis l'explorateur de fichiers.
- Les images s'affichent sous forme d'une liste de vignettes.
- Lorsqu'une image est sélectionnée, elle est affichée en vraie grandeur. L'opérateur peut alors zoomer et se déplacer dans l'image pour en contrôler le contenu et la qualité.
- Des traitements de base sont disponibles pour chaque image : suppression et rotation.



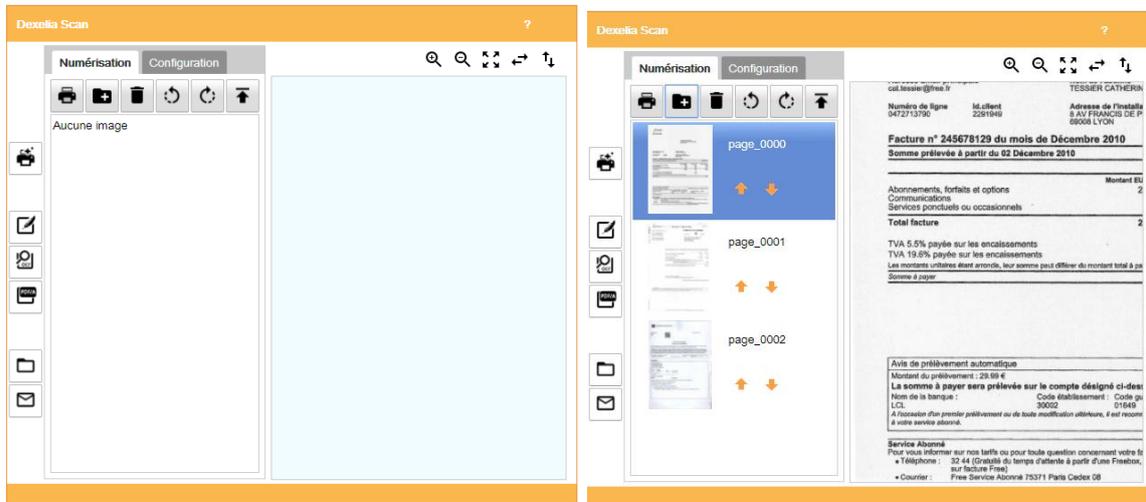


Figure 2 : Écran de numérisation mono-document

La numérisation peut être réalisée en une plusieurs fois et mixer des images numérisées ou importées. Il est également possible de réordonner les images par un simple « glisser / déposer ».

Lorsque l'ensemble des pages du lot sont numérisées, le bouton  valide le lot est déclenche le transfert des images numérisées vers l'application métier hôte, selon le paramétrage du composant : mono-pages ou multipages JPG, TIF, PDF, PDF/A ; en mode FTP, http, mail ou fichier local; avec ou sans OCR ; comprimé MRC ou non ; ...

Dans la Figure 2, ces options sont accessibles par les boutons alignés verticalement à gauche. Selon l'application, ces options sont accessibles ou non.

Il est possible de tester le mode mono document à l'adresse suivante :

<https://shared.diginove.com/dexelia¹>

3 MULTI-DOCUMENTS

Dans ce mode, il est possible de numériser des lots de documents en appliquant des éléments de configuration regroupé en filière. La filière décrite par un fichier Json permet de définir :

- Un profil de numérisation ou une liste de profils de numérisation utilisable (recto/verso, résolution, couleur/niveaux de gris/noir et blanc,...)
- Le nom du prétraitement d'image à appliquer lors de l'acquisition des images. Ces prétraitements permettent par exemple de définir la méthode de séparation (Code à barres, patch,...)
- Le schéma de nommage du lot
- La nature et la liste des informations d'indexation à saisir ou à extraire

¹ L'utilisation de l'application nécessite l'installation des drivers TWAIN correspondants au scanner ainsi que du module ScanServer (Cf. 4 Eléments techniques).

- Le format et la méthode de publication (format des images, format du fichier compagnon, destination)

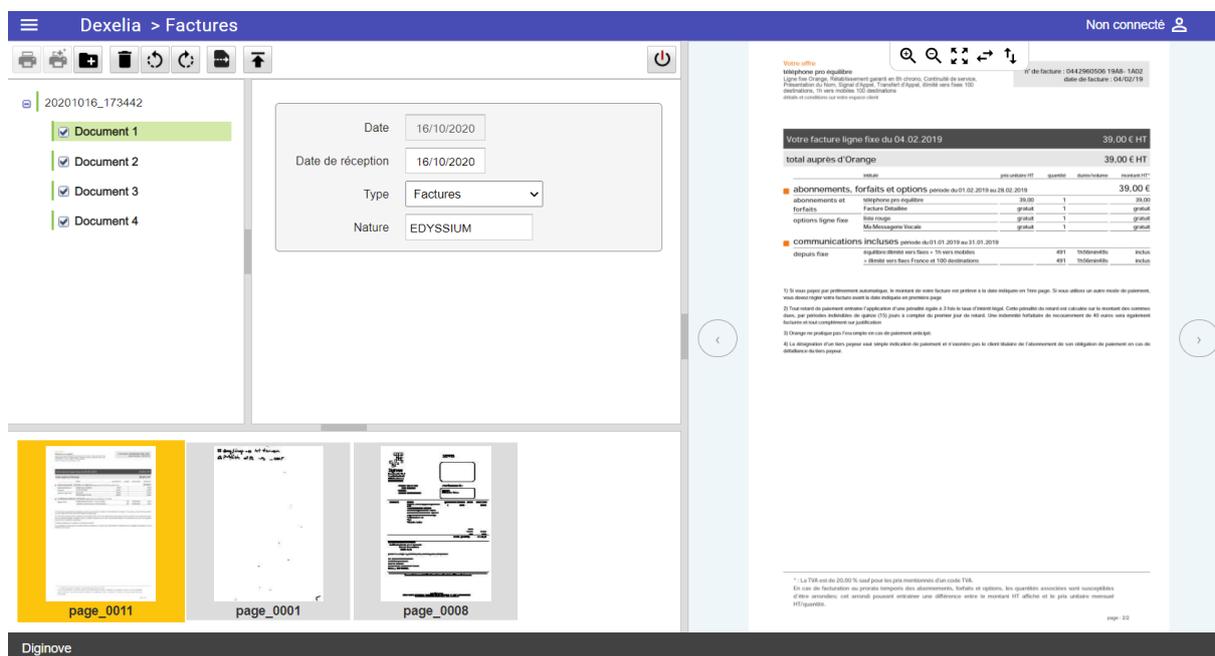


Figure 3 : Écran multi documents

La partie gauche présente l’arborescence du lot à numériser, ici pour la filière « Factures »

Si des images sont présentes, le document présente une case cochée. Un document conforme aux critères (obligatoire, nombre de pages) a une barre verticale verte sur la gauche. Sinon, elle est orange.

L’ergonomie de visualisation peut être modifiée à l’aide d’un fichier CSS.

Le document sélectionné est surligné en vert. La liste des imagerettes de ce document apparaît alors. Il est possible de déplacer par « glisser / déposer », les images d’un même document ou de déplacer vers un autre document. Comme pour le mode mono document, il est possible d’appliquer des traitements aux images.

Il est possible de tester le mode multi documents à l’adresse suivante :

<https://shared.diginove.com/dexelia/DexeliaMDL.html>²

² L’utilisation de l’application nécessite l’installation des drivers TWAIN correspondants au scanner ainsi que du module ScanServer (Cf. 4 Éléments techniques).

4 ELEMENTS TECHNIQUES

Environnement technique

La numérisation Web fonctionne sous tous les navigateurs, en mode https³, dans un environnement Windows 7 - 10 ou Windows server 2013 et au-delà.

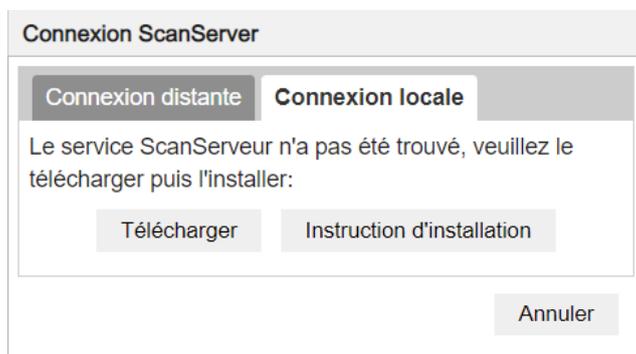
La publication du ou des fichiers image peut être réalisée selon différents moyens : http ou https (requête http put ou post), ftp, sftp ou ftps, Web Services, Webdav, CMIS, ...

Le pilotage du scanner nécessite un driver TWAIN, fourni par le constructeur, installé en mode administrateur du poste. De plus, le fonctionnement de la numérisation Web nécessite l'installation d'un composant sur le poste de l'utilisateur en mode administrateur. Cette installation peut être réalisée soit en même temps que le driver du scanner, soit lors de la première utilisation.

Il n'y a pas de contrainte particulière sur la configuration du poste de l'utilisateur. Le logiciel occupe environ 300 M octets sur le disque dur.

Installation

Lors de la première utilisation, l'application ne trouvant pas le module ScanServeur, il sera possible de télécharger l'exécutable permettant de l'installer. Les instructions d'installation sont accessibles en utilisant le bouton « Instruction d'installation ».



L'installation nécessite les droits administrateurs sur la machine.

Concernant les drivers TWAIN, s'ils ne sont pas déjà installés, ceux-ci peuvent être téléchargés sur le site « support » du constructeur.

³ Si http était possible à l'origine, l'utilisation de https est devenue nécessaire sur les versions récentes des navigateurs qui interdisent par défaut l'accès à localhost ou 127.0.0.1 en http.

SDK de développement

4.1.1 Principe de fonctionnement

Diginove fournit un SDK facilitant l'intégration dans une application. Ce SDK est développé dans un environnement Eclipse / GWT, mais l'approche est la même pour d'autres environnements.

Le schéma ci-dessous en explique les principes, pour un poste sous Windows :

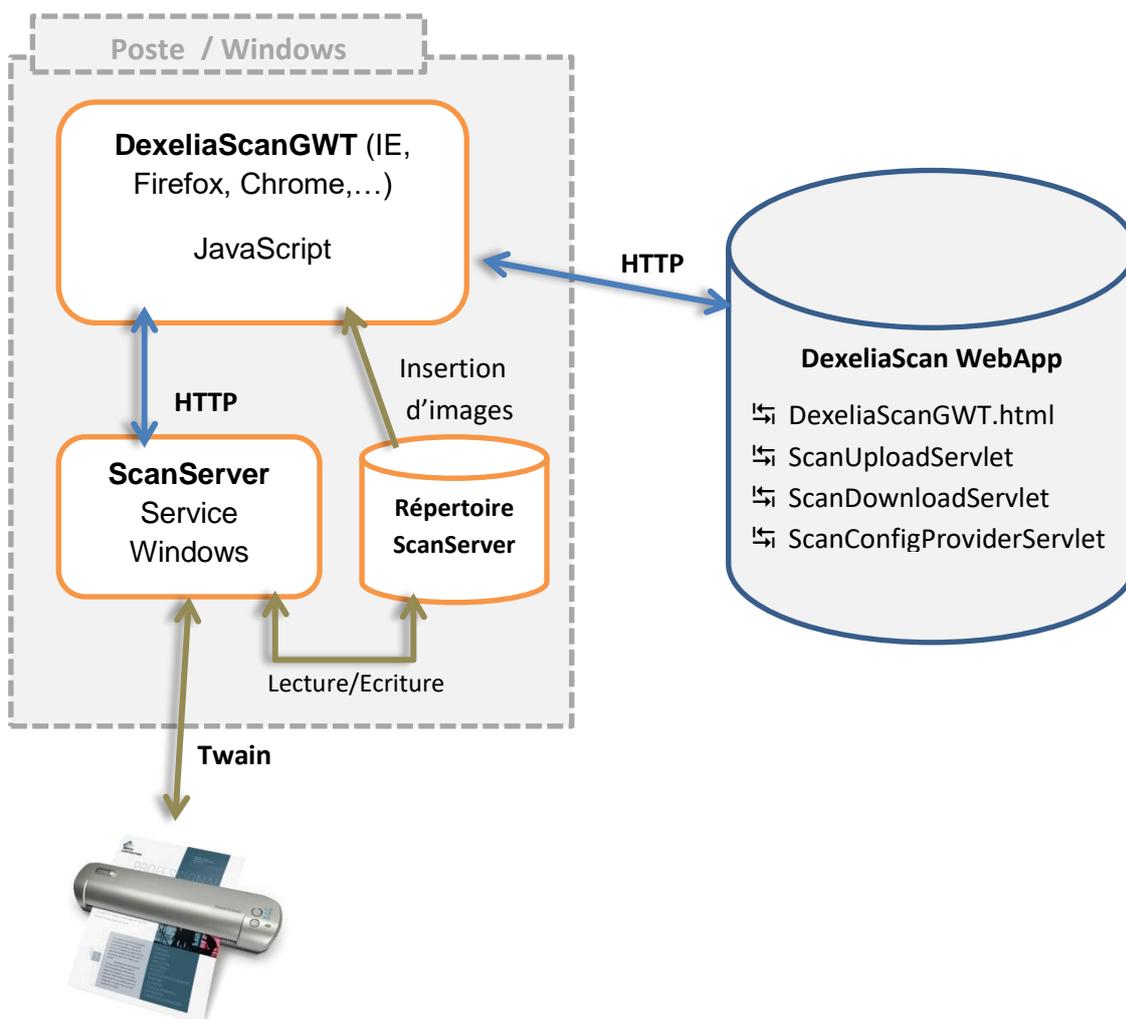


Figure 4 : Principe de fonctionnement monoposte

L'application DexeliaScanGWT est exécutée par un navigateur. La page DexeliaScanGWT.html, ainsi que toutes les ressources nécessaires, sont servies par un serveur appelé ici DexeliaScan WebApp.

L'application dialogue avec un serveur http local (ScanServer) chargé de piloter le scanner et de réaliser les accès disque. Un répertoire est dédié au ScanServer dans l'espace de l'utilisateur (enregistrement des images scannées, stockage temporaire des documents publiés, ...).

ScanServer est associé à un service Windows, chargé de le démarrer ou de l'arrêter, dans la session Windows active.

Ce mode de fonctionnement permet de fonctionner dans un cadre :

- Multi-utilisateurs : une seule installation du ScanServer est nécessaire pour tous les utilisateurs.
- Multi-session : il est possible de passer d'une session Windows à une autre.
- Metaframe (TSE ou Citrix) : un plugin TSE et Citrix est alors nécessaire, mais le fonctionnement est garanti sans ralentissement lors de la numérisation ou de la publication.

En complément l'application utilise également différents servlets :

- ScanConfigProviderServlet

Ce servlet permet de récupérer soit :

- Les fichiers de configuration du scanner relativement au nom de la source Twain et au contexte utilisateur :
- La liste des profils de numérisation relativement au nom de la source Twain et au contexte utilisateur :
- Un profil de numérisation :
- Le package d'installation du ScanServer

- ScanUploadServlet

Ce servlet est utilisé pour uploader les fichiers lors de la publication, si celle-ci est réalisée par http.

- ScanDownloadServlet

Ce servlet est utilisé pour télécharger un fichier publié, dont l'URL est renvoyé par ScanUploadServlet.

Il est également possible d'utiliser l'application depuis un mobile ou un poste non Windows. Pour cela, le ScanServer du poste Windows va être utilisé pour partager le scanner et fournir le service de numérisation sur ces postes qui n'ont pas accès au scanner. Dans ce cas, au lieu d'installer ScanServer, il faut déclarer l'adresse IP du poste partageant le scanner.



Figure 5 : Configuration du partage de scanner

Voici le principe :

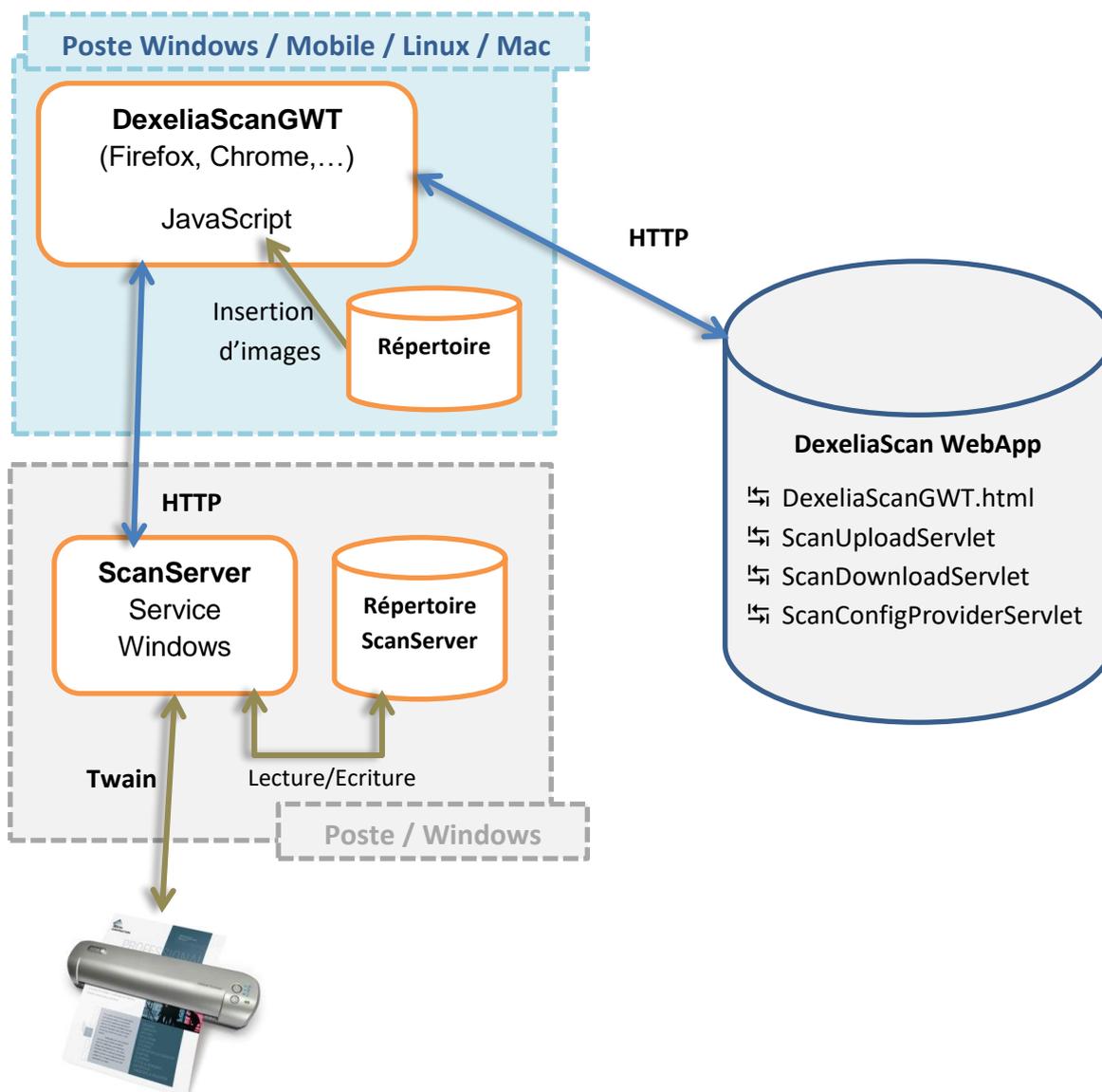


Figure 6 : Principe de fonctionnement en mode partage de scanner

ScanServer est un exécutable capable de piloter un scanner (Twain) et d'exécuter des scripts de traitement d'image. Il supporte plusieurs modes de communication (ligne de commande, socket, named pipe). Dans le cadre de la numérisation Web, celui-ci est démarré en mode socket http sur le port 9003 (configurable). Dans ce mode, ScanServer fait également office de serveur http capable de « servir » les répertoires d'images et les images elles-mêmes.

Au chargement de l'application, celle-ci tente de communiquer avec ScanServer (commande ping), s'il n'y a pas de réponse, le dialogue suivant propose soit de saisir l'adresse du poste exécutant ScanServer (Connexion Distante) soit de réaliser son téléchargement (Connexion locale).

L'installateur ScanService_Setup.exe installe un service nommé ScanServerService chargé du démarrage et de l'arrêt de ScanServer ainsi que de sa surveillance (redémarrage en cas d'arrêt). Le service est paramétré en démarrage automatique, le but étant de s'assurer que ScanServer soit toujours disponible. L'installateur doit être exécuté en mode administrateur. Lorsque l'utilisateur n'a pas les droits administrateur, le module doit être préalablement installé, en même temps que le driver du scanner, qui requiert également les droits administrateurs.

Le répertoire d'installation est par défaut C:\Program Files (x86)\Dexelia Scan Service. Lors de l'installation le répertoire de travail {%USERPROFILE%}\ScanServer est créé ainsi que les sous-répertoires suivants :

- images : répertoire contenant les images scannées
- thumb : répertoire contenant les « imagerettes » afférentes
- out : répertoire de génération des fichiers de publication (pdf)

4.1.2 Mode d'échange avec ScanServer

Les commandes ScanServer sont envoyées au format XML avec la structure suivante :

```
<commande name="{idCommand}" attributeKey1="attributeValue1"
attributeKey2="attributeValue2" ... >
  <param name="param1">valueParam1</param>
  <param name="param2">valueParam2</param>
  ...
</commande>
```

Sauf exception, les commandes sont généralement envoyées directement dans l'url (HTTP GET), toutefois, lorsque la taille du XML constituant la commande risque de ne pas être compatible avec une requête de type GET on peut basculer sur une requête de type PUT (sauvegarde d'un fichier, ...). Les commandes sont transmises au ScanServer en utilisant l'url :

```
http://localhost:9003/=commande=</maCommandeXML>
```

Code retour des commandes

ScanServer retourne lors de l'exécution d'une commande le XML du type

```
<scanserver>
  <response name="{idCommand}">{CodeRetour}</response>
</scanserver>
```

avec {idCommand} le nom de la commande exécutée et {CodeRetour} une des valeurs suivantes :

- notopened_nok : Source TWAIN non connectée exécution de la commande impossible
- badparam_nok : Un des paramètres n'est pas valide
- notsupported_nok : La commande n'est pas supportée par le driver TWAIN
- badvalue_nok : Valeur non valide ou en dehors des plages supportées
- badcommand_nok : Commande non reconnue par ScanServer
- ok : La commande a bien été exécutée
- nok : La commande n'a pas été exécutée

Plusieurs commandes peuvent être envoyées en même temps :

```
<command name="cmd1"/>
<command name="cmd2">
  <param name="paramCmd2">valueParam</param>
</command>
```

Les codes retour de chaque commande sont retournés :

```
<scanserver>
  <response name="cmd1">ok</response>
  <response name="cmd2">ok</response>
</scanserver>
```

Un exemple significatif des commandes est listé ci-dessous :

Initialisation de ScanServer

Ping : Permet de valider la présence du ScanServer.

Version : Récupération du numéro de version du ScanServer

SetContext : Permet de spécifier au ScanServer le contexte d'utilisation auquel est associée la licence

Context : Récupération du nom du contexte courant

Thumb : Permet de spécifier au ScanServer si oui ou non il doit générer les imageries (0 : non, 1 : oui) lors de l'insertion d'images.

Connexion et initialisation du Scanner

Open : Connexion à une source TWAIN. Le nom de la source doit être spécifié dans le paramètre « scanner », le paramètre « doinmainthread » doit être en général défini à false et le paramètre « twain2 » à true.

Close : Déconnexion de la source TWAIN courante

RestDSM : Réinitialisation de la source TWAIN (rechargement de la liste des scanners, ...).

ListScanner : Récupération de la liste des sources TWAIN de la machine

GetScanner : Récupération du nom de la source TWAIN connectée

SetupScanner : Initialisation de ScanServer relativement à un modèle de scanner.

Pilotage du Scanner

ListResolution : Récupération de la liste des résolutions supportées par le scanner connecté.

ListPixelFormat : Récupération de la liste des formats couleur supportés.

GetRectoVerso : Récupération du réglage du scanner vis-à-vis du recto/verso

RectoVerso : Réglage du scanner vis-à-vis du recto/verso

GetPixelFormat : Récupération du format couleur courant : 0, 1 ou 2 (Cf. ci-dessus ListPixelFormat)

PixelFormat : Spécification du format couleur à utiliser : 0, 1 ou 2 (Cf. ci-dessus ListPixelFormat)

GetResolution : Récupération de la résolution courante (en DPI)

Resolution : Spécification de la résolution à utiliser (DPI)

Scan / ScanUI : Lancement de la numérisation. ScanUI permet d'invoquer le dialogue de configuration TWAIN.

Le paramètre outDir définit le répertoire devant recevoir les images scannées, le répertoire thumbDir celui des imageries relativement au répertoire de l'utilisateur connecté (Ex : C:\Users\toto\ScanServer).

Gestion des images

Le répertoire de référence utilisé par ScanServer est le sous répertoire « ScanServer » du répertoire de l'utilisateur connecté (Ex : C:\Users\toto\ScanServer). ScanServer implémentant un mini serveur de fichier, la liste des images est directement récupérée par une requête `http://localhost:9003/images/`, le résultat est une page html.

DeleteFile : Suppression d'un ou plusieurs fichiers

ChooseDir : Permet à l'utilisateur de choisir un répertoire sur son disque

SendMail : Création d'un mail avec les documents spécifiés en pièce attachées

Process : Exécution d'un script de commande sur des fichiers. Les paramètres sont les suivants :

- inputFile : Les fichiers ou les répertoires à traiter (séparateur ;)
- outputFile : Le répertoire de sortie
- scriptFile : le script relativement au répertoire d'installation de SpiScanService
- extensions : les extensions des fichiers à prendre en compte
- parameters : des paramètres spécifique au script (Ex ci-dessous 270 = angle de rotation)

Parmi les scripts livrés :

- Suppression de toutes les images d'un répertoire et des imageries associées (raz.js)
 - inputFile = .\images\
○ outputFile = .\images\
○ scriptFile = raz.js
○ extensions = JPG, TIF, PDF, GIF, BMP
○ parameters = thumb\
- Rotation d'une image ou d'une imagerie (rotate.js)
 - inputFile = la ou les images à traiter (Ex : ".\images\page_0000.bmp ; .\images\page_0001.bmp")
 - outputFile = .\images\
○ scriptFile = rotate.js
○ extensions = Les extension à prendre en compte (Ex : BMP)
 - parameters = l'angle de rotation (90, 180, 270, ...)
- Insertion d'un fichier image (insertImage.js)
- Publication (pdf.js)

Cas de l'insertion d'un fichier

Le contenu du fichier est transmis au ScanServer via le paramètre « inputContent ». Le contenu doit être encodé en base64 (objet file l'entête "data:application/pdf;base64," doit être supprimée).

- inputFile = le nom du fichier à traiter
- outDir = le répertoire des images (Ex : .\images\
- thumbDir = le répertoire des imageries (Ex : .\thumb\
- scriptFile = insertImage.js
- imgPrefix = Le préfixe à ajouter aux images (Ex : page_)
- inputContent = le contenu du fichier à traiter encodé en base 64

Cas de la publication

Le script pdf.js produit un fichier PDF à partir d'un répertoire ou d'une liste de fichiers (paramètre outputFile). Le XML <setup> montre les options utilisables lors de la génération du fichier.