Ce document a pour but de faciliter l'intégration du transmetteur SwissDecTX dans une application salariale. Il contient des conseils et des informations relatives aux travaux à effectuer durant la phase d'intégration et de préparation pour la certification. Toutefois, il ne se substitue en aucun cas aux documents officiels SwissDec, qui feront toujours foi en cas de doute. En particulier, le présent document n'a ni la prétention d'être complet, ni celle d'être totalement rigoureux ou précis. La documentation SwissDec, en particulier les directives pour la transmission salariale, la documentation relative au format des données, les spécifications pour la transmission et le protocole de test officiel, décrivant pas-à-pas la procédure de test appliquée lors de la certification, sont à lire (et à observer) scrupuleusement. Le transmetteur lui-même est entièrement documenté dans un fichier séparé (SwissDecTX.chm)

Pré-requis

- Compte sur le SwissDec Lab (swissdec.ch)
- Compte sur les serveurs de test (itserve.ch)
- Monitoring ID spécifique à votre projet (délivré par SwissDec)
- 3 certificats de test (CA, receiver, transmitter avec mot-de-passe, délivrés par SwissDec)
- La documentation SwissDec complète, en particulier les directives pour la transmission salariale.
- Le transmetteur SwissDecTX et sa documentation (disponible sur www.softwarecomponents.ch)

Les comptes, identifiants, certificats et mots de passe nécessaires pour les tests ainsi que la documentation et les divers exemples disponibles fournis par SwissDec s'obtiennent tous directement au travers du site swissdec.ch et/ou de votre expert SUVA régional, chargé de la certification.

Types de données XML à générer depuis le programme salaire

- Déclaration
- Déclaration de remplacement
- Déclaration (marquée en tant que données de test)
- Déclaration de remplacement (marquée en tant que données de test)

Le format des diverses déclarations énumérées ci-dessus est quasiment identique. Les déclarations de remplacement et/ou de test se distinguent simplement par l'ajout de balise(s) spécifique(s) dans l'entête du document. Voir la documentation SwissDec, en particulier les directives pour la transmission salariale UC006 <TestCase> et UC007 <Substitution> (TransmitterRequirements_f.pdf ainsi que SalaryDeclarationContainerTechDoc_fr.pdf).

Le mode « remplacement » et le mode « test » doivent obligatoirement être implémentés par le programme salaire (ajouter, le cas échéant, la ou les balise(s) requise(s) par ces modes dans la déclaration XML), la partie transmission, tant manuelle qu'électronique, reste quant à elle strictement identique. Le transmetteur SwissDecTX transférera les déclarations sans jamais ajouter ou retirer de balises XML.

Types de transmission

- **Transmission manuelle**, via browser (IE, Chrome, FireFox…), directement sur une page web SwissDec, sous deux formes distinctes :

    1. Déclarations sur un transmetteur *online*, utile pour tester la validité des données. Ces données non-signées sont celles générées par l'application salariale, exemple : le fichier `ICHAGCompany.xml` faisant partie de la documentation technique disponible sur le SwissDec Lab, comportant éventuellement les balises de test et/ou de remplacement mentionnées plus haut. Il n'y a rien de particulier à faire ici, on peut simplement envoyer la déclaration XML telle que produite par le programme salaire, le fichier `ICHAGCompany.xml` étant un exemple d'une telle déclaration. Cette méthode sert surtout à valider l'intégrité de la déclaration salariale.

    2. Enveloppes SOAP signées, év. cryptées, (EIV) transmises manuellement via un récepteur *online.* Ces données sont créées par le transmetteur SwissDecTX (voir `DeclareSalaryLocal` dans la documentation technique du transmetteur) à partir de la déclaration créée par le programme salaire et sauvées sur disque, puis transmises manuellement par l'utilisateur, via un simple browser web, sur un récepteur online particulier. C'est le cas de figure « EIV » qui doit obligatoirement être implémenté par toutes les applications. Il est important de noter que les enveloppes signées générées par le transmetteur SwissDecTX pour le cas EIV sont munies d'un « migros-data » (TTL) de 3600 secondes. L'operateur a donc exactement 60 minutes pour ouvrir son browser et envoyer la déclaration, faute de quoi cette dernière devient irrémédiablement caduque (il faudra en générer une nouvelle).

- **Transmission électronique** directe via le transmetteur SwissDecTX

    Enveloppes SOAP signées, év. cryptées, transmises directement sur un service web (c'est le mode principal). Ces données signées/cryptées sont créées par le transmetteur SwissDecTX en respectant à la lettre les spécifications SwissDec pour la signature numérique et le cryptage des informations. Voir `DeclareSalary` et `DeclareSalaryDeferred` dans la documentation technique du transmetteur pour transmettre une déclaration crée par l'application, soit de manière directe, soit en deux phases (envoi dit « asynchrone » avec récupération ultérieure du résultat final). Ce sont les cas de figure « PIV synchrone » et « PIV asynchrones » décrits dans la documentation SwissDec, qui doivent tous deux obligatoirement être implémentés par toutes les applications salariales.

    La transmission s'effectue par HTTPS et ne requiert en principe aucune configuration spéciale, il faut simplement que l'ordinateur qui transmet soit connecté « normalement » à l'internet, directement ou au travers d'un proxy.

    Le transmetteur SwissDecTX supporte l'authentification auprès d'un proxy web, parfois nécessaire en environnement d'entreprise (voir `SetHttpProxy` et `SetHttpProxyWithCredentials` dans la

documentation du transmetteur), ainsi que la détection automatique des réglages nécessaires via WPAD (Web Proxy Auto-Discovery).

En clair cela revient à dire qu'il n'y a normalement rien à faire de particulier pour assurer la connexion et que si l'on se trouve dans un environnement réseau contrôlé, où l'accès internet est bloqué, le transmetteur SwissDecTX supporte les fonctions nécessaires pour s'authentifier auprès du proxy/gateway permettant de « sortir » sur l'internet.

L'application salariale peut prévoir ce cas en exposant une page de configuration où l'administrateur du réseau pourra entrer les paramètres nécessaires pour accéder à l'internet, typiquement l'adresse du proxy HTTP, le port TCP à utiliser et, le cas échéant, un nom d'utilisateur et un mot de passe si le proxy exige ces informations.

Il suffira alors d'appeler l'une ou l'autre des deux fonctions mentionnées ci-dessus pour permettre au transmetteur SwissDecTX d'atteindre les serveurs SwissDec sur le web.

Si l'application n'est utilisée que dans des environnements « ouverts », tels que les connections ADSL/VDSL usuelles, il n'est peut-être pas nécessaire de prévoir le cas du proxy web (ce cas n'est pas exigé par SwissDec à ce jour). Sachez simplement que le transmetteur SwissDecTX est déjà prêt pour ce cas de figure et que dans certains cas le proxy sera pris en compte automatiquement.

Destinations

- Serveur(s) de test
- Serveur(s) de production

Plusieurs URLs sont possibles, il faut être en mesure de changer l'adresse du serveur relativement aisément depuis l'application, un outil de configuration, un fichier de paramètres… Le transmetteur SwissDecTX offre un moyen programmatique (voir `SetUrl`) et un outil de configuration externe (`SwissDecTX.Config.exe`), d'autre part la configuration du transmetteur est stockée dans le registre, donc il est aisé de modifier l'URL extérieurement, voir HKCU\Software\ARSD\SwissDecTX\Url.

Toutes les combinaisons possibles de types de documents, types de transmissions et de destinations doivent être supportées. Suivant la destination (test/production/demo) les certificats nécessaires peuvent être différents.

Adresses utiles (SwissDec v2.2)

- Receveur (test) pour la transmission électronique directe « PIV synchrone et asynchrone » :

  https://tst.itserve.ch/itserve/lohnstandard/testing/refapps2/receiver-webservice/20051002-0.0/SalaryDeclarationService

- Site the configuration du receveur test ci-dessus :

https://tst.itserve.ch/itserve/lohnstandard/testing/refapps2/receiver/20051002-0.0/login.xhtml

- Receveur pour la transmission manuelle EIV depuis un browser :

  https://tst.itserve.ch/itserve/lohnstandard/testing/refapps2/eiv-web/20051002-0.0/

- Transmetteur online (test), surtout utile pour la validation des données salariales (accepte des données XML non signées, telles que ICHAGCompany.xml) :

  https://tst.itserve.ch/itserve/lohnstandard/testing/refapps2/transmitter/20051002-0.0/login.xhtml

D'autres adresses peuvent être utilisées lors des tests en vue de la certification. Les URLs ci-dessus sont mentionnées ici à titre purement indicatif, contactez SwissDec ou itServe en cas de doute.

Répartition des tâches entre le programme salaire et le transmetteur SwissDecTX

Le programme salaire crée les données XML contenant la déclaration salariale, au format requis par SwissDec. Au moment où les travaux concernant la transmission commencent, ces documents XML ont typiquement déjà reçu l'aval de l'expert SUVA régional concernant leur forme et leur contenu. Le programme salaire maintient un journal des transmissions dans lequel sont archivés, verbatim, tous les messages envoyés et les réponses reçues dans leur forme brute, c'est-à-dire les enveloppes SOAP des messages émis et des réponses correspondantes. L'implémentation du journal des transmissions est à la discrétion du développeur, toutefois le transmetteur SwissDecTX offre un mécanisme de journalisation automatique, où les messages envoyés et reçus sont stockés sur disque dans un répertoire déterminé par vous-même. En outre, le transmetteur SwissDecTX supporte les fonctions nécessaires pour créer les documents à archiver, il est donc possible de créer votre propre mécanisme pour le maintien du journal. L'application peut maintenir ce journal comme bon lui semble, par exemple en stockant les enveloppes SOAP dans sa base de données ou à tout autre endroit, pourvu qu'il soit accessible dans le futur. Le maintien d'un journal des transmissions est exigé et le journal maintenu par le transmetteur SwissDecTX est admissible pour la certification.

Pour les cas « EIV » (transmissions manuelles) l'application invoque le transmetteur SwissDecTX (voir `DeclareSalaryLocal`) qui crée le message final, signé, daté et éventuellement crypté, mais sans le transmettre. Du point de vue de l'utilisateur, il peut par exemple s'agir d'une fonction de type « Save As… », où l'utilisateur a l'opportunité de spécifier un chemin et un nom de fichier pour sauvegarder la déclaration prête à l'envoi. Le document ainsi crée sera ensuite envoyé manuellement par ce même utilisateur, depuis un browser web, sur une page « EIV » similaire au receveur test EIV mentionné en page 4. Les déclarations ainsi créées expirent après 60 minutes, il faut donc les transmettre sans tarder.

Pour les cas « PIV » (transmissions directes) on distingue deux sous-cas : tout d'abord la transmission dite *synchrone* où le transmetteur envoie les données signées, datées et éventuellement cryptées, puis retourne la réponse immédiatement (voir `DeclareSalary`). Le second cas est celui de la transmission dite *asynchrone* ou le programme salaire envoie la déclaration (voir `DeclareSalaryDeferred`) et reçoit un

ticket (« job key ») en retour qu'elle stocke, puis passe ce ticket à une autre fonction du transmetteur (voir `GetStatusFromDeferredDeclaration`), typiquement dans les 24h suivant l'émission initiale, pour récupérer les résultats finaux.

Dans les deux cas PIV, l'application reçoit au final une URL et un mot de passe que l'utilisateur devra impérativement utiliser, depuis un browser web, afin de confirmer l'envoi définitif et la « libération » des données. La libération doit s'effectuer dans les délais prescrits par SwissDec, faute de quoi la déclaration ne sera pas prise en compte par le serveur.

Tout comme chacun des messages envoyés, les réponses du serveur doivent être stockées dans le journal des transmissions, d'autre part l'URL et le mot de passe final doivent être présentés à l'utilisateur sous une forme lui permettant d'accéder facilement à la page web ou la libération des données s'effectue.

En cas d'erreur, le serveur retournera divers éléments d'information incluant un statut et un ou plusieurs message(s) d'erreur ou « warning(s) », qui devront tous être présentés à l'utilisateur.

Il peut s'agir de message d'ordre technique (erreur survenue lors de la transmission) ou un message résultant d'une erreur logiques (par exemple un montant incohérent contenu dans les données, ou une date improbable).

Le transmetteur SwissDecTX valide tout ce qu'il transmet et reçoit, au moyen des schémas « XSD » officiels fournis par SwissDec. Il détecte de ce fait une grande partie des erreurs et toutes les « malformations », par exemple une balise manquante ou une erreur de syntaxe XML. Il détecte aussi toute une série d'erreurs techniques telles que les signatures invalides, manquantes, la corruption des données en cours de transmission, les interruptions de connections… (tous ces cas sont vérifiés formellement lors de la procédure de certification).

En revanche le transmetteur ne peut pas valider les données d'un point de vue logique, par exemple un employé né dans le futur : 05.02.2110 est une date parfaitement valide mais sera certainement rejetée, en 2010, en tant que date de naissance d'une personne. Il est donc nécessaire d'afficher les résultats à l'utilisateur sous une forme simple et compréhensible pour qu'il/elle puisse le cas échéant prendre les mesures correctives nécessaires avant de resoumettre la déclaration au moyen d'un message de remplacement.

Pour résumer, en amont, le programme salaire crée les déclarations au format XML et indique, si besoin est, si il s'agit d'une déclaration de remplacement et/ou de test au moyen de balises XML appropriées.

Le transmetteur s'occupe de la validation technique d'après schémas, de la signature, du cryptage optionnel, de la création des documents nécessaires à l'archivage dans le journal des transmissions ainsi que de la transmission elle-même pour les deux cas « PIV » synchrone et asynchrone, tout en détectant toutes les conditions d'erreur prévues par le programme de certification SwissDec.

Enfin, en aval, l'application archive les messages envoyés et les réponses correspondantes dans son journal (c.-à-d. les documents XML correspondant aux paramètres `messageSent` et `messageReceived` passés au transmetteur) et présente les informations nécessaires à l'utilisateur (extraits des informations retournées dans les paramètres `result`, `plausibility` et `jobFinished`). Optionnellement, le maintien d'un journal simple, sur disque, peut être confié au transmetteur lui-même. Il suffit d'activer et de

paramétrer le journal depuis l'outil de configuration du transmetteur en indiquant l'emplacement du dossier dans lequel stocker les messages et le mode de stockage : le transmetteur peut créer un sous-répertoire par transmission (avec un nom approprié comportant entre autres la date etc) et y placer les fichiers XML correspondant au message envoyé et à la réponse correspondante. Le transmetteur peut aussi créer un fichier archive (au format Zip) pour chaque transmission et stocker le message transmis et la réponse sous forme compressée (c'est le mode que nous recommandons, sachant que ce journal n'existe que pour des raisons d'audit). L'application peut facilement énumérer le contenu de ce répertoire et extraire les informations nécessaires à l'affichage du journal, si désiré, en interprétant les différents champs dont le nom des fichiers (ou des sous-répertoires) sont composés (on a la date, l'heure, le type de message et le résultat de l'opération sous forme condensée : acceptation, refus, erreur…, c.-à-d. assez d'informations pour afficher une liste représentant le journal.). Le mécanisme de journalisation intégré au transmetteur SwissDecTX est admissible pour la certification, il n'est donc pas obligatoire d'implémenter un journal des transmissions dans la compta.

L'appel du transmetteur est extrêmement simple. Par exemple, pour le cas EIV, une seule ligne de code suffit pour invoquer la fonction créant le fichier EIV nécessaire à la transmission manuelle. Une seule ligne est également nécessaire pour effectuer une transmission synchrone (premier cas « PIV ») et deux appels seulement pour le cas « PIV » asynchrone.

Bien sûr, il reste le travail amont/aval décrit plus haut mais la partie transmission se résume essentiellement à appeler une fonction ou deux et à interpréter le résultat, d'autre part la journalisation automatique optionnelle rend la tenue du journal extrêmement simple (il n'y a rien à faire d'autre que d'activer cette fonction).

L'emplacement du journal et le type de journalisation (sous-répertoires ou archives Zip) sont gouvernés par des clés de registre. L'outil de configuration permet de créer ces clés mais il est également possible de les créer programmatiquement ou durant l'installation de votre application. L'emplacement du répertoire peut être spécifié librement (toutefois il est fortement **déconseillé** de tenir le journal dans un sous-répertoire de « \Program Files\ » sachant que cette partie du disque est essentiellement « read-only » pour les utilisateurs normaux). Les paramètres de journalisation sont stockés dans

HKLM\SOFTWARE\ARSD\SwissDecTX\TransmissionLog

..où on trouvera deux valeurs: *Location* (REG_SZ) qui pointe vers le répertoire à utiliser et *Type* (REG_DWORD) qui peut valoir 0 (sous-répertoires) ou 1 (archives Zip, recommandé).

S'agissant d'un log d'audit, l'outil de configuration ne permet pas de désactiver ou de changer les paramètres du log après-coup. On peut bien sur modifier les valeurs dans le registre, ou simplement effacer la clé « TransmissionLog » pour désactiver la journalisation automatique mais il faut garder en tête que la tenue d'un journal est obligatoire pour obtenir la certification.

En général, le transmetteur ne touche pas au contenu de la déclaration. Il y a toutefois deux exceptions : la première est que SwissDecTX ajoute sa signature (nom et version) dans l'entête des documents qu'il transmet (dans le tag `<ct:Name>`). Le but est d'assure une traçabilité en cas de problème. La seconde exception est que le transmetteur peut optionnellement générer automatiquement le contenu du tag `<ct:RequestID>` : si la valeur est vide, ou égale à un astérisque (« * ») ou encore à la chaine « *reqId* » - comme dans les exemples SwissDec – le transmetteur insèrera un identifiant unique de 20 caractères.

A propos du transmetteur

Le transmetteur SwissDecTX n'a pas à proprement parler « d'interface utilisateur ». Il s'agit d'un composant qui se présente sous différentes formes et qui peut être appelé directement depuis quasiment n'importe quel programme ou environnement de développement sous Windows. Le transmetteur lui-même est écrit en langage C# et dépend de .NET 2.0 SP1. Le transmetteur fonctionne aussi parfaitement dans un environnement .NET 3.0 (Windows Vista) ou .NET 3.5 SP1 (Windows 7) ou encore .NET 4.x, mais sans nécessiter la (lourde) installation de l'un de ces trois derniers « frameworks » sur les plateformes plus anciennes telles, que Windows 2000, XP ou Vista.

Le transmetteur s'installe automatiquement au moyen d'un programme, `SwissDecTX.Setup.exe`, qui téléchargera et installera automatiquement les prérequis nécessaires depuis le web, si nécessaire.

Il n'y a donc rien de spécial à faire en regard de .NET sur les machines des clients : il suffit d'exécuter le programme d'installation pour installer le transmetteur sur n'importe quelle PC.

Comme dit plus haut, le transmetteur dépend de .NET 2.0 SP1 et a de ce fait les mêmes exigences que ce dernier en matière de PC et d'OS : Windows 2000 avec Service Pack 4 est la version la plus ancienne supportée par .NET 2.0 et de ce fait par le transmetteur SwissDecTX.

Le fonctionnement sous Windows 98/SE/Me n'est pas entièrement exclu, mais nécessiterait probablement des travaux supplémentaires et ne peut être garanti. En l'état, le fonctionnement sur des versions de Windows antérieures à Windows 2000 SP4 n'est pas supporté. Nous contacter le cas échéant.

Il existe plusieurs niveaux d'intégration possible, mentionnées ci-dessous par ordre croissant de technicité :

- Interface « ligne de commande »

  C'est le niveau le plus élémentaire et le plus universel. L'application lance un petit exécutable avec des paramètres sur la ligne de commande (fonction à exécuter, nom des fichiers en entrée et en sortie) et la transmission s'effectue. On peut ainsi transmettre « à la main » une déclaration depuis une invite de commande. Le programme s'appelle `SwissDecTX.exe` et est installé en standard avec le transmetteur. L'endroit où il s'installe est écrit par l'installateur dans le registre (voir HKLM\SOFTWARE\ARSD\SwissDecTX\InstallPath) donc il suffit d'aller lire cette valeur[1], de concaténer le nom du programme, d'ajouter les paramètres nécessaires et de lancer le tout au moyen d'une fonction de type « run » ou « exécuter », qui existe dans tous les environnements de développement. C'est une excellente première approche, qui permet d'obtenir un résultat immédiat avec très peu de programmation : si votre compta génère déjà les déclarations XML, une heure suffit pour intégrer le transmetteur au moyen de l'interface ligne de commande et effectuer les premières transmissions.

[1] Ne pas se fier à « C:\Program Files\SwissDecTX » ou autre, d'une part « Program Files » est localisé et son nom change en fonction de la langue du système d'exploitation, d'autre part l'administrateur (p.ex. celui s'occupant du réseau de l'entreprise qui sera votre client, et donc pas vous !) peut décider d'installer les programmes à un tout autre endroit : le transmetteur ira donc à cet endroit et non pas où on s'attendrait à le trouver normalement. Il faut *toujours* lire la clé de registre pour connaitre l'emplacement où se trouve le transmetteur, puis concaténer le nom de l'exécutable, resp. de la DLL, si possible en utilisant une fonction intelligente telle que, par exemple, `PathAppend()` de Win32 (shlwapi) ou `System.IO.Path.Combine()` en .NET

Fichier à utiliser : SwissDecTX.exe (installé automatiquement avec le transmetteur)

- Interface « DLL »

  L'interface de type DLL est destinée aux environnements de développement capables d'intégrer les librairies dynamiques. Ces environnements incluent les langages C et C++ mais aussi les anciennes versions de VB ainsi que beaucoup d'outils, anciens ou modernes. Le transmetteur installe toujours une petite DLL, nommée `SwissDecTX.dll`, qui expose les fonctions de transmission sous une forme appelable directement (C, stdcall). La documentation du transmetteur inclut un fichier nommé `SwissDecTX.API.h` qui contient les définitions de toutes les fonctions exposées par la DLL, laquelle support les jeux de caractères Unicode et ANSI.

  Les programmes C/C++ peuvent utiliser cette interface directement et utiliser toutes les techniques usuelles telles que *late binding* (via LoadLibrary), ou *static binding* (au moyen de la librairie SwissDecTX.lib fournie, avec *delay-load* ou non, peut-être en ajoutant le chemin approprié[2] au PATH pour que la DLL soit trouvée). Une chose à **ne pas** faire est d'extraire la DLL et de la placer, par exemple, dans le répertoire de votre application. C'est un moyen très sûr de créer des problèmes dans le futur si le transmetteur est mis à jour mais que la DLL, maintenant séparée, reste de l'ancienne version.

  Fichier à utiliser : SwissDecTX.dll (installé automatiquement avec le transmetteur)

  Pour les langages autres que C/C++ l'interfaçage avec une DLL est souvent possible mais il requiert un certain doigté car les paramètres et leur type doivent souvent être spécifiés explicitement et il se peut qu'il ne soit pas immédiat de trouver la bonne correspondance entre, par exemple, le type « date » d'un langage de programmation donné avec le type « date » utilisé par l'interface DLL du transmetteur. On préférera l'interface COM, voir paragraphe suivant, si l'environnement de développement le supporte.

- Interface « COM »

  COM, pour Component Object Model est un modèle introduit en 1993 par Microsoft, en partie pour tenter d'éliminer le problème connu sous le nom de « DLL hell » et d'autre part pour profiter d'un moyen sûr et moderne apte au développement d'applications et de systèmes modulaires.

  De nombreux systèmes ont été bâtis sur COM, à commencer par OLE2 et une pléthore d'autre choses incluant, entre autres, ActiveX et bien d'autres encore. COM est solidement ancré dans la

[2] Ne pas se fier à « C:\Program Files\SwissDecTX » ou autre, d'une part « Program Files » est localisé et son nom change en fonction de la langue du système d'exploitation, d'autre part l'administrateur (p.ex. celui s'occupant du réseau de l'entreprise qui sera votre client, et donc pas vous !) peut décider d'installer les programmes à un tout autre endroit : le transmetteur ira donc à cet endroit et non pas où on s'attendrait à le trouver normalement. Il faut *toujours* lire la clé de registre pour connaitre l'emplacement où se trouve le transmetteur, puis concaténer le nom de l'exécutable, resp. de la DLL, si possible en utilisant une fonction intelligente telle que, par exemple, `PathAppend()` de Win32 (shlwapi) ou `System.IO.Path.Combine()` en .NET

plateforme Windows et il constitue même *le futur* de cette plateforme, puisque la plupart des nouvelles fonctions de Windows 7 (DirectWrite, Direct2D, Taskbar API, Sensor & Location API etc) sont délivrées uniquement sous la forme de composants et d'interfaces COM.

Le modèle COM jouit d'une mauvaise réputation, en partie parce qu'il est difficile à programmer correctement, néanmoins pratiquement tous les environnements de développement plus ou moins modernes (comme PROGRESS, VB6 ou Delphi) supportent parfaitement son usage, ou tout au moins la « consommation » de composants COM, ce qui nous intéresse ici.

Il suffit d'intégrer le transmetteur via un fichier nommé `SwissDecTX.Transmitter.tlb`, fourni avec la documentation, dans votre environnement de développement, typiquement au moyen d'une fonction du type « Import Type Library » présente dans la plupart d'entre eux.

A partir de là on crée une instance du transmetteur et on appelle ses fonctions, souvent avec l'aide « IntelliSense » disponible directement dans l'éditeur de code. C'est le moyen le plus direct et le plus facile pour intégrer le transmetteur, par exemple dans Visual Basic ou Delphi : une fois importé, un icône symbolisant le transmetteur apparait dans une barre d'outil et il suffit d'en ajouter une instance au projet, comme on le ferait pour n'importe quel bouton ou « editbox », pour intégrer le transmetteur en quelques secondes.

L'outil de développement est en général assez intelligent (et dispose d'assez d'informations) pour proposer des choix appropriés pour l'ordre et le type des paramètres, voir pour effectuer automatiquement les conversions de type de données (p.ex. date ou décimal) si nécessaires.

L'usage de cette interface est fortement recommandé si votre outil de développement le supporte.

Fichier à utiliser : SwissDecTX.Transmitter.tlb (fourni avec la documentation du transmetteur)

- Interface « .NET »

Le transmetteur étant lui-même un composant .NET, les utilisateurs de cet environnement pourront intégrer et utiliser le transmetteur SwissDecTX sans le moindre effort. Il suffit pour cela de référencer le fichier `SwissDecTX.Transmitter.dll`, faisant partie de la documentation, depuis le projet, typiquement au moyen d'une fonction de type « Add Reference ».

On prendra soin d'éditer les propriétés de cette référence pour dire à l'environnement de **ne pas** copier cette DLL localement (ce qui implique aussi qu'elle ne fait **pas** partie des éléments constituant l'application et ne devra **pas** être installée avec l'exécutable principal de votre produit).

Dans Visual Studio 2005, 2008 ou 2010, ce réglage s'effectue simplement en indiquant la valeur « False » à la propriété « Copy Local » de la référence à la DLL `SwissDecTX.Transmitter.dll` (vérifier de visu que cette DLL n'est **pas** copiée dans le répertoire qui contient l'exécutable principal de l'application).

En effet, il est inutile d'inclure cette DLL avec l'application, elle ne sert qu'à indiquer l'allure du composant à l'environnement de développement .NET.

L'installateur automatique du transmetteur installe et configure cette DLL au bon endroit (GAC), elle ne doit donc pas être installée séparément par l'application.

L'interface .NET permet d'éviter de passer par des documents XML écrits sur disque. Le transmetteur expose des fonctions qui permettent de traiter et d'échanger directement des documents XML en mémoire.

Fichier à utiliser : SwissDecTX.Transmitter.dll (fourni avec la documentation du transmetteur)

Le développeur peut choisir librement son niveau d'intégration en fonction de l'environnement de développement dont il dispose. Certain langages ou systèmes offrent moins de choix, mais en fin de compte tout le monde devrait trouver chaussure à son pied. D'autre part il est possible de développer de nouveaux interfaçages si nécessaire – nous contacter le cas échéant.

Notons qu'il n'est jamais nécessaire d'ajouter un fichier à la distribution de votre application (hormis `SwissDecTX.Setup.exe`). Pour les deux premiers cas, `SwissDecTX.exe` et `SwissDecTX.dll` sont installés automatiquement durant le déploiement du transmetteur et dans les deux derniers cas les fichiers `SwissDecTX.Transmitter.tlb` et `SwissDecTX.Transmitter.dll` sont uniquement destinés à servir de référence à l'environnement de développement (IDE) et à fournir à ce dernier les indications nécessaires à l'appel du composant.

Ces fichiers ne doivent **pas** être inclus dans votre application ni installés séparément, ils sont nécessaires uniquement en environnement de développement.

Configuration du transmetteur

Il existe plusieurs façons de configurer le transmetteur, mais voyons tout d'abord ce qui doit être configuré :

- L'URL du serveur sur lequel envoyer les transmissions « PIV »
- Les trois certificats fournis par SwissDec
- La licence du transmetteur
- Le cryptage optionnel des données
- Le journal des transmissions (optionnel, l'application peut maintenir son propre journal)

Le moyen le plus simple de configurer le transmetteur est d'utiliser l'outil de configuration graphique, `SwissDecTX.Config.exe`, installé automatiquement avec le transmetteur. Cet outil permet de configurer les divers éléments mentionnés ci-dessus ainsi que de spécifier le « monitoring ID » à utiliser pour le test d'interopérabilité, que ce programme permet également d'effectuer par simple pression d'un bouton.

Les quatre points à configurer, listés plus haut, sont mémorisés de façon permanente dans la machine (par utilisateur), on peut donc parfaitement se contenter de configurer le transmetteur une fois pour toutes au moyen de cet outil et de ne rien faire d'autre, ces paramètres seront utilisés par défaut par le transmetteur jusqu'à ce qu'on les modifie.

L'URL s'obtient auprès de SwissDec. Celle mentionnée plus haut dans ce document (voir page 4) était valide au moment où ces lignes ont été écrites, mais il est possible qu'elle ait changé entretemps.

Les certificats de test fournis avec les exemples disponibles sur le SwissDec Lab sont les suivants :

- SwissDec Root CA Certificate : `cacert.crt`
- Client Certificate : `test.p12` (mot de passe : `testZertifikat`)
- Server Certificate : `receiver.crt`

La licence à utiliser, `SwissDecTX.Demo.lic`, est fournie avec la documentation du transmetteur.

Le transmetteur expose des fonctions (API) qui permettent de gérer une partie de ces réglages programmatiquement (URL, cryptage, sélection des certificats parmi ceux installés).

L'outil « ligne de commande » `SwissDecTX.exe` comporte un mode d'installation des certificats et offre donc un moyen d'installer ces derniers programmatiquement.

L'outil de configuration graphique permet de vérifier la configuration en un instant et d'effectuer immédiatement un contrôle du fonctionnement, toutefois cet outil n'offre pas pour l'instant de fonction permettant de spécifier l'usage d'un proxy, il faut passer par l'interface programmatique pour cela (ou profiter de la configuration automatique).

Enfin, les réglages mémorisés sont stockés dans le registre, par utilisateurs, sous la clé

```
HKCU\Software\ARSD\SwissDecTX
```

Le chemin d'installation du transmetteur ainsi que le chemin pointant vers le journal des transmissions sont également stockés dans le registre, par machine, sous la clé

```
HKLM\SOFTWARE\ARSD\SwissDecTX
```

Quelques conseils et remarques (après 5 certifications réussies !)

- On ne s'y prend jamais assez tôt.
- Lisez **scrupuleusement** les spécifications SwissDec, y compris « entre les lignes ».
- Effectuez **plusieurs fois** la série de tests de certification à blanc (le protocole de test est très précis et formellement documenté, genre : l'opérateur entre -9999.90, puis presse OK. Réponse attendue : …, réponse constatée : …, test réussi : o/n, etc. (il s'agit là d'une partie du test `001_InteroperStd`, premier parmi les 4 tests d'interopérabilité). Prévoir une interface-utilisateur pour effectuer les tests d'interopérabilité depuis votre programme, où l'opérateur entre lui-même ses chiffres et la chaîne de caractères accentués et où les résultats des calculs, la chaîne retournée par le serveur et l'heure du serveur sont affichés à l'écran. Le transmetteur, via son outil de test et configuration, offre une interface-utilisateur pour effectuer le test d'interopérabilité, lequel est recevable pour la certification.
  Vous pouvez toutefois prévoir votre propre interface au sein du programme salaire si vous le désirez : il s'agit d'afficher 6 ou 7 champs et d'appeler une seule fonction, *CheckInteroperabilityWithStringOperands,* laquelle effectue toutes les vérifications demandées (exactitude des calculs, différence de temps, chaînes de caractères correctes etc).

- Il est **indispensable** de vous procurer la dernière version du protocole de test auprès de SwissDec/itServe. Il est également indispensable d'effectuer et de réussir tous les tests à blanc longtemps (10 jours ?) avant le rendez-vous de certification, afin d'avoir le temps de corriger les détails et aussi de faire « un peu de volume » pour bien tester la journalisation des messages (si elle est effectuée par votre application), la gestion et l'affichage des différentes erreurs etc.
- Il faut soigner la présentation des messages d'erreur, en particulier les réponses indiquant la plausibilité de la déclaration. En cas de problème, l'opérateur doit être capable de s'y retrouver et de comprendre pourquoi sa déclaration a été rejetée (ou seulement partiellement acceptée) et quelle sont les mesures correctives à prendre le cas échéant. Le transmetteur offre en option un moyen simple de formater les réponses sous forme de HTML, l'affichage produit étant admissible pour la certification. L'outil de test et de configuration utilise lui-même le composant de formatage des réponses pour afficher les résultats de manière claire et lisible. Il est possible d'obtenir exactement le même affichage dans votre programme en quelque lignes de code, d'autre part il est possible de personnaliser entièrement la conversion en HTML (p.ex. changer les polices, couleurs ou même le « layout ») au moyen d'une feuille de style XSLT (nous contacter).
- Vous devrez accéder au récepteur de test, à travers votre compte sur le serveur itServe, pour configurer les différentes « fausses erreurs » qui sont délibérément provoquées par le serveur et qui font toutes l'objet d'un cas-test spécifique. Pour certain des tests vous devez tenter de transmettre avec le câble réseau déconnecté etc. L'outil de test et de configuration comporte des liens cliquables vers les sites appropriés, tels que le receveur, le receveur EIV et le transmetteur en-ligne (voir l'onglet « About »).
- Il faut se tenir prêt à changer l'URL durant les tests, car ces derniers ne s'effectuent pas nécessairement sur les serveurs de test itServe (il est normalement possible de changer l'URL aisément au moyen de l'outil `SwissDecTX.Config.exe` mais si votre application choisit d'appeler elle-même la fonction `SetUrl()` avant chaque transmission, il faut prévoir de pouvoir configurer rapidement cette adresse *aussi* dans votre application).
- L'ordinateur utilisé pour les transmissions doit être raisonnablement à l'heure (ceux de vos clients devront l'être aussi, d'ailleurs). A cet effet, Windows supporte la mise à jour automatique de l'horloge du système, à la seconde près, au travers de serveurs situés sur l'internet (p.ex. `time.windows.com`). Le protocole utilisé, NTP, requiert l'ouverture du port UDP 123 dans les pare-feu et éventuels proxies. Par défaut Windows ajuste l'heure une fois par semaine. Dans le cas de postes en réseau il est fréquent que le serveur (ou contrôleur de domaine) donne l'heure à toutes les machines connectées, dans ce cas il faut s'assurer que le serveur responsable ait accès à une horloge fiable et soit mis à l'heure régulièrement.
- N'oubliez pas que SwissDec met 65 heures de consulting à votre disposition dans le cadre du programme de certification, n'hésitez pas à poser des questions, en particulier sur le déroulement des tests de certification et des exigences y-relatives.

Finalement, n'hésitez pas à contacter info@softwarecomponents.ch si vous avez des questions ou des remarques concernant le transmetteur ou sa mise en œuvre !

Installation du transmetteur

L'installation s'effectue de manière simple et automatique. Il suffit d'exécuter le fichier SwissDexTX.Setup.exe sur la machine cible. Ce programme d'installation vérifie la présence des pré-requis et

les télécharge puis les installe si nécessaire, puis installe les composants du transmetteur. Il n'y a pas de message de fin et l'interface-utilisateur de ce programme d'installation est volontairement minimale.

Les pré-requis sont

- .NET Framework 2.0 SP1 (seulement pour Windows 2000/XP)

Windows 7, Windows Vista, Windows Server 2008, Windows Server 2008 R2 et les versions plus récentes de Windows contiennent déjà le .NET Framework 2.0 (il fait partie de .NET 3.x/4.x).

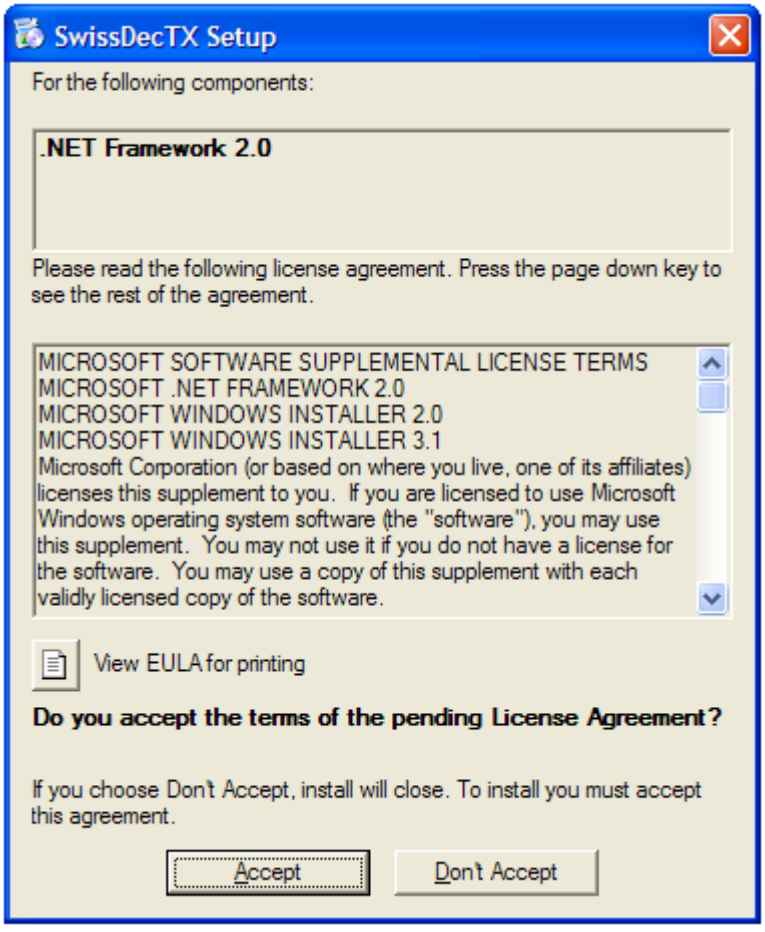
Le .NET Framework 2.0 SP1 peut être téléchargé depuis la page suivante :

http://www.microsoft.com/Downloads/details.aspx?familyid=79BC3B77-E02C-4AD3-AACF-A7633F706BA5

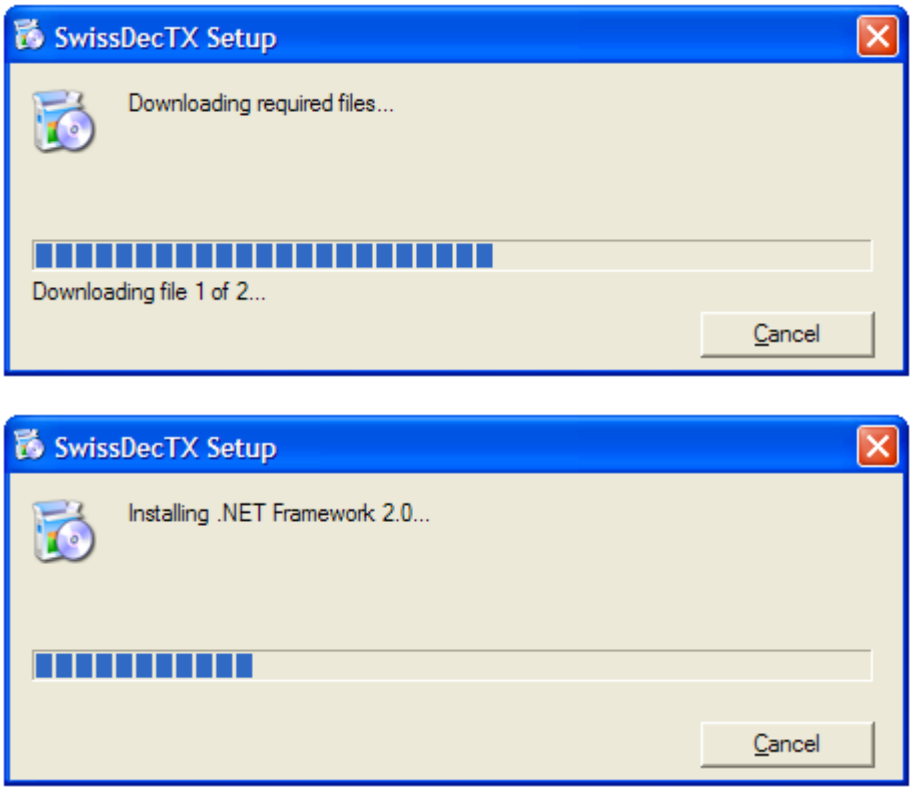
On peut, si l'on préfère, préinstaller ce framework de sorte à minimiser les messages affichés durant l'installation du transmetteur.

Sur une machine XP SP3 « vierge », c.-à-d. sans .NET préalable, l'installation se déroule comme suit :

Tout d'abord le programme d'installation détecte qu'il doit télécharger le .NET Framework 2.0. Les confirmations suivantes s'affichent :



L'installateur téléchargé ensuite les framework et l'installe:



L'installateur se termine ensuite silencieusement (**pas de message de fin**). Le tout ne prend que 2-3 minutes sur une machine moderne avec une connexion rapide. Si .NET est déjà présents l'installation du transmetteur ne prend que quelques secondes.

Il est possible de réduire le nombre de messages en préinstallant les pré-requis : .NET peut être préalablement installés manuellement ou peut être installé silencieusement par votre propre routine d'installation avant de lancer l'installeur SwissDecTX.Setup.exe, ce faisant il est possible d'intégrer l'installation du transmetteur sans nécessiter d'interaction supplémentaire pour l'utilisateur final.

Si les pré-requis sont déjà présents (par exemple sous Windows Vista et suivantes) on verra simplement s'afficher une petite fenêtre avec une barre de progrès, durant quelques secondes au plus. Notez donc bien que le téléchargement et l'installation de .NET 2 SP1 n'est nécessaire que sous W2K/XP et seulement si ce framework n'est pas déjà présent dans l'ordinateur.

Le transmetteur, sont outil de configuration et la DLL d'interfaçage sont maintenant installés. Vous trouverez ces deux derniers dans C:\Program Files\SwissDecTX et tout ce qui est nécessaire au fonctionnement du transmetteur est maintenant installé. Il n'y a pas d'entrée dans le menu démarrer ni aucun icône. L'idée est que le transmetteur est un composant sans interaction directe avec l'utilisateur final.

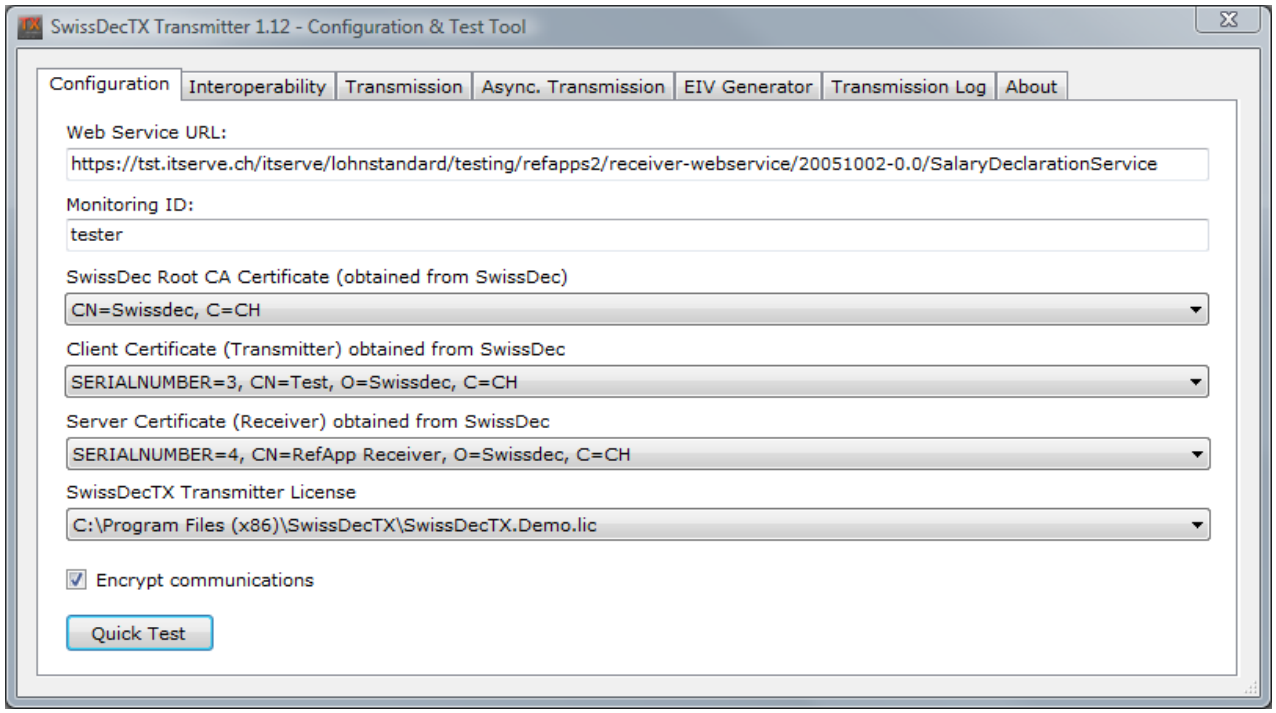
Pour le développement, le transmetteur est fourni avec un certain nombre d'autres fichiers, entre autres une présentation PowerPoint, le présent document, un fichier d'aide décrivant l'interface de programmation (API) dans tous ses détails (voir SwissDecTX.chm), des fichiers XML test, une « *type library* » pour les clients COM (.tlb), un fichier .lib pour les clients C/C++ et une DLL, *SwissDecTX.Tranmitter.dll*, destinée au référencement depuis un environnement de développement .NET.

Ces fichiers peuvent être copies n' importe où, par exemple dans un répertoire « proche » du répertoire de développement de votre application. Aucun des fichiers auxiliaires ne doit être distribué avec votre application.

Outil de test et de configuration

Le transmetteur est livré avec un outil de test et de configuration, *SwissDecTX.Config.exe*, dont voici les détails (NB : les copies d'écran datent d'une version antérieure et peuvent varier légèrement):

Configuration :



Le premier champ reçoit l'adresse du service avec lequel le transmetteur communiquera. Cette URL est fournie par SwissDec. L'exemple ci-dessus montre l'adresse du serveur de test en vigueur à ce jour. L'adresse du serveur de production sera différente.

Le second champ reçoit votre « monitoring ID ». Il s'agit d'un identifiant attribué par SwissDec qui désigne le compte vers lequel les transmissions s'effectueront sur le serveur de test. Vous pourrez examiner les résultats en-ligne en vous connectant sur la page de configuration du receveur au moyen de ce même identifiant et d'un mot de passe également attribue par SwissDec. Remarque : vous disposerez de deux jeux d'identifiant + mot-de-passe séparés : un pour le SwissDec Lab (site contenant les spécifications etc) et l'autre pour le receveur.

Les trois champs suivants indiquent quels certificats doivent être utilisés. SwissDec utilise un mode sécurisé très sûr impliquant un certificat avec clé privée coté client et un certificat avec clé privée coté serveur. La transmission est sécurisée « *end-to-end* » dans les deux sens, en plus, le « *channel* » est sécurisé par l'usage de SSL/HTTPS, c'est-à-dire que la transmission internet véhicule de façon sécurisée des données qui sont elle-même déjà entièrement sécurisées directement par le transmetteur et le receveur. A supposer qu'un

attaquant réussisse à compromettre le serveur web recevant les données, il sera presque sûrement impossible d'altérer les messages reçus et transmis car ces derniers comportent leur propre signature (et optionnellement leur propre cryptage), ce qui revient intuitivement à élever au carré le niveau de sécurité du système, par rapport à une communication SSL/HTTPS classique. Ce mode de transmission nécessite deux certificats X509, comportant chacun des clés de 1024 bit dans ce cas particulier. En outre, SwissDec ayant choisi d'émettre ses propres certificats, un certificat racine comportant une clé de 2048 bit devra également être installé.

Le premier des trois champs indique le certificat racine (« *root CA* ») de SwissDec. Il contient la clé publique permettant de vérifier la légitimité des deux autres certificats. Le certificat racine de test est nommé « *cacert.crt* ». Il suffit de choisir l'option « Browse… » (en cliquant dans le champ) et d'indiquer à l'outil de configuration l'emplacement du fichier en question. Le certificat est installé dans un « *certificate store* » géré par Windows : une fois configuré le fichier « cacert.crt » n'est plus nécessaire et peut être effacé du disque dur (ceci est également vrai pour les deux autres certificats). Notez qu'il faut être administrateur pour installer un certificat racine et que Windows demandera obligatoirement confirmation par l'affichage d'un message.

Le second des trois champs indique le certificat « transmetteur » à utiliser. Le certificat de test est nommé « test.p12 ». Ce certificat contient la clé privée qui sera utilisée pour signer les messages sortants et aussi pour décrypter les messages en provenance du serveur.

Il est protégé par un mot de passe : dans le cas du certificat test ce mot de passe est « *testZertifikat* ».

Pour la production, vous recevrez votre propre certificat de transmission avec votre propre mot de passe.

Le certificat et son mot de passe devront être considérés comme confidentiels car ils permettent de signer des déclarations en votre nom. En particulier, le mot de passe du certificat devrait être tenu raisonnablement secret et le fichier contenant le certificat ne devrait pas être laissé sur le disque dur de la machine du client après l'installation.

Le mot de passe n'est nécessaire que pour l'installation du certificat. Si les installations / configurations de votre produit sont faites manuellement sur les machines des clients, les trois fichiers contenant les certificats n'ont pas besoin d'être copiés : il suffit de les tenir par exemple sur une clé USB et de les installer directement depuis cette clé. Une sauvegarde complète de la machine copiera également le « *certificate store* » et tous les certificats qu'il contient.

Le dernier des trois certificats est le certificat du receveur (serveur). Il contient la clé publique du receveur qui permet de crypter les messages à destination de ce dernier et de vérifier l'authenticité des réponses reçues. Le certificat test est nommé « receiver.crt ».

Les trois certificats devront être configurés dans toutes les machines effectuant des transmissions. Il est possible d'installer les certificats au moyen de l'outil « *command line* » (voir le mode ICERT), par exemple depuis un script ou en exécutant programmatiquement la commande depuis la routine d'installation de votre application. On peut aussi installer les certificats simplement en les double-cliquant depuis Windows Explorer. Les trois certificats de test nécessaires pour le développement sont inclus dans le fichier `SwissDecTX_TestFiles.zip` fourni avec le transmetteur.

Le champ suivant contient le nom du fichier contenant la licence d'utilisation du transmetteur. Une licence de démonstration, « *SwissDecTX.Demo.lic* » est fournie dans le fichier .zip disponible sur http://www.softwarecomponents.ch. Cette licence permet d'effectuer des transmissions à titre d'essai sur le serveur de test de SwissDec avec le monitoring ID « tester ».

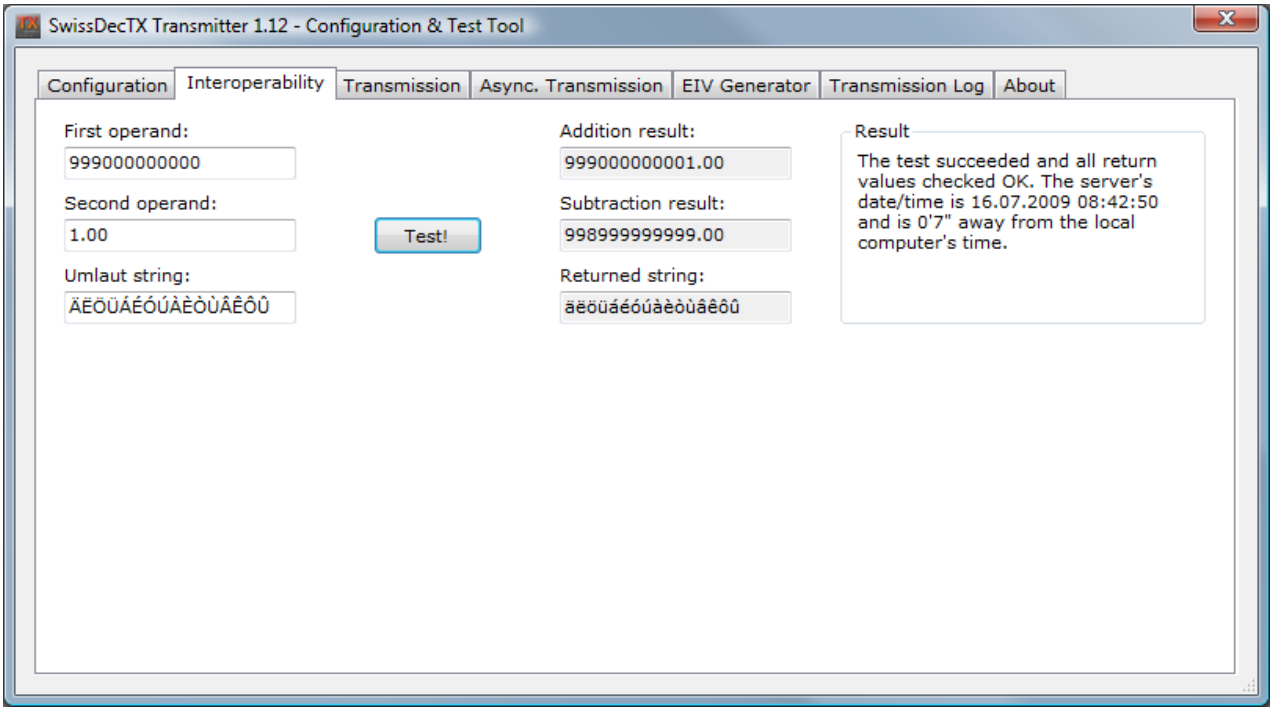
En outre, la licence est verrouillée sur certain champs d'entête (voir l'entête du fichier d'exemple « ICHAGCompany.xml » fourni par SwissDec) mais le contenu de la déclaration peut être arbitraire.

Prière de nous contacter pour l'obtention d'une licence personnalisée permettant d'effectuer des transmissions en votre nom et au moyen de votre propre « monitoring ID », ainsi que de présenter votre application pour les tests de certification. Une fois la certification obtenue, SwissDec délivrera les certificats de production et une licence finale du transmetteur – correspondant au certificat officiel – vous sera remise pas nous même.

La case à cocher permet d'indiquer au transmetteur s'il doit effectuer un cryptage des messages ou non. Le cryptage est une fonction optionnelle (désactivé par défaut) pour les versions antérieures à SwissDec 3, d'autre part les fichiers EIV ainsi que les messages envoyés archivés dans le journal des transmissions ne sont jamais cryptés.

Le bouton « Quick Test » effectue un appel au serveur (fonction « Ping ») et vérifie l'atteignabilité de ce dernier. Si cette fonction s'exécute sans erreur la connexion internet est fonctionnelle et l'URL est correcte : il convient de passer au « moment de vérité » sur l'onglet suivant :

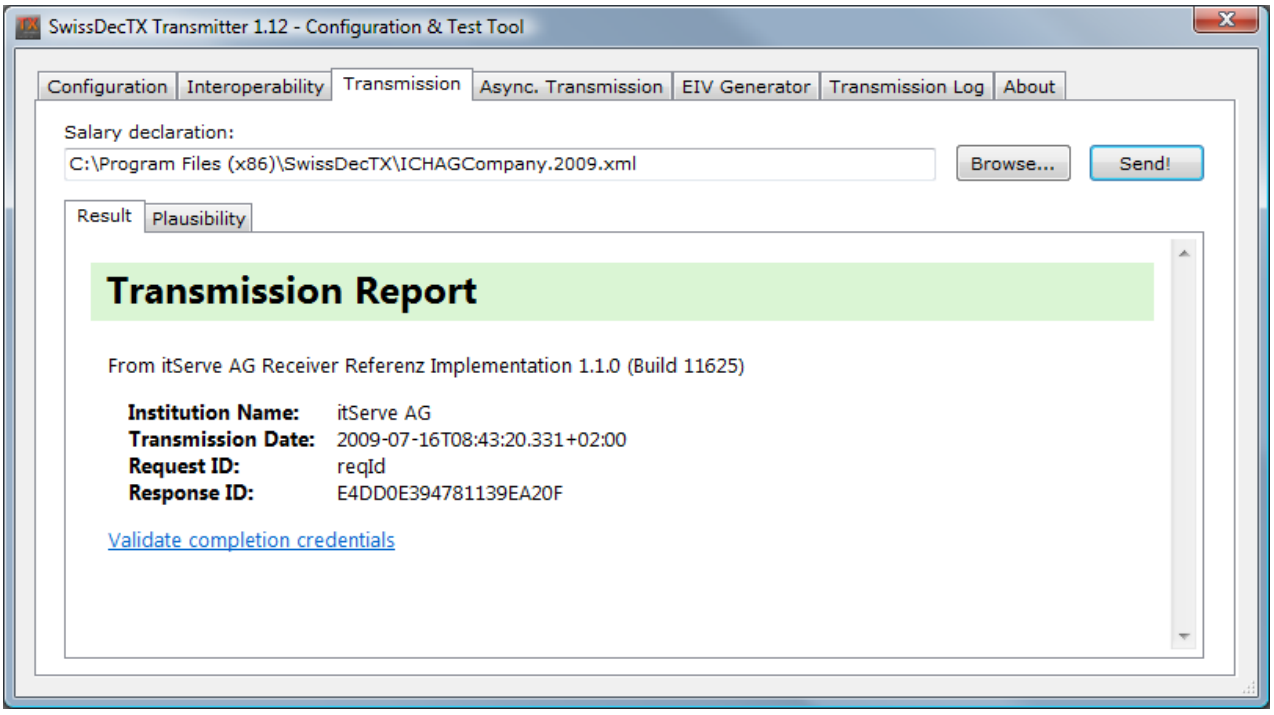
Test d'interopérabilité :



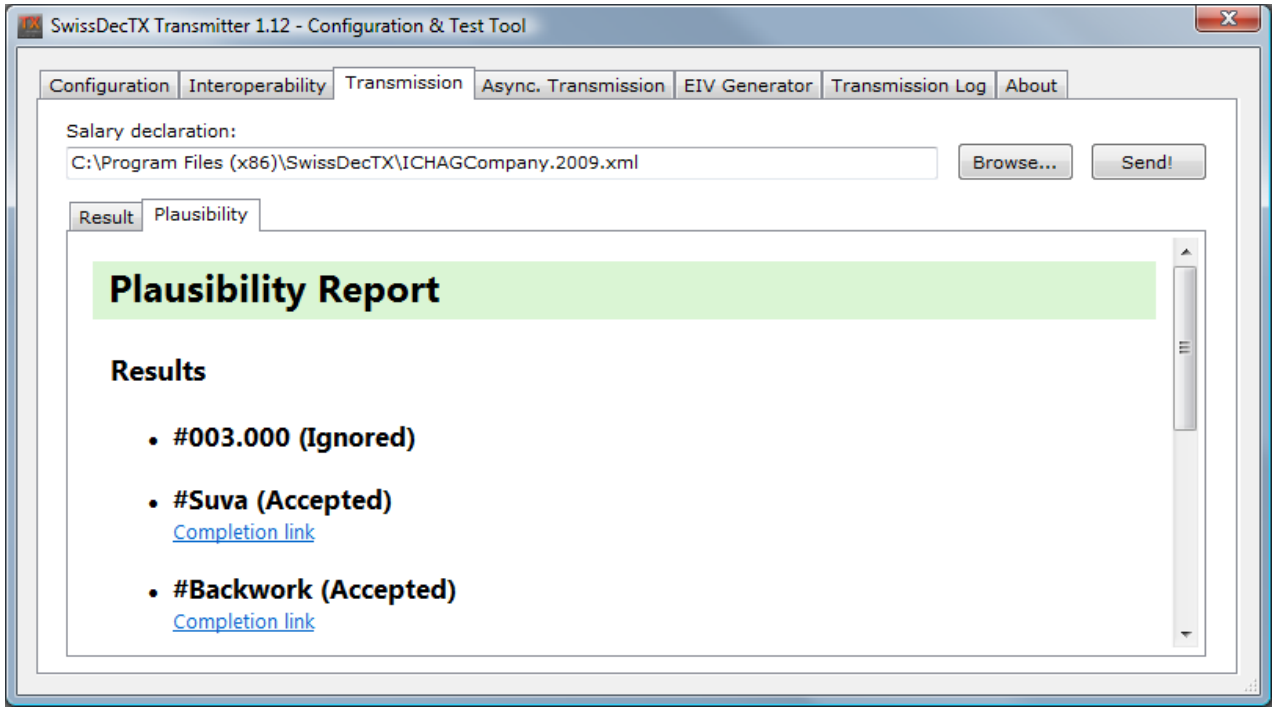
Le test d'interopérabilité s'effectue de manière très simple en cliquant sur le bouton « *Test!* ». La colonne de gauche contient les valeurs d'entrée et partie de droite affiche les résultats. Ce test s'effectue en utilisant les paramètres de la page « Configuration », en particulier l'URL, le paramètre « Monitoring ID », les certificats

et l'option de cryptage. Il est possible d'aller vérifier la teneur du message reçu par le serveur et de la réponse retournée en visitant votre compte sur la page de configuration du receveur. Si ce test se déroule sans erreurs (comme dans l'exemple ci-dessus) le plus dur est fait : le transmetteur est correctement installé et configuré, la dernière étape consistant maintenant à effectuer un « vrai » test de transmission.

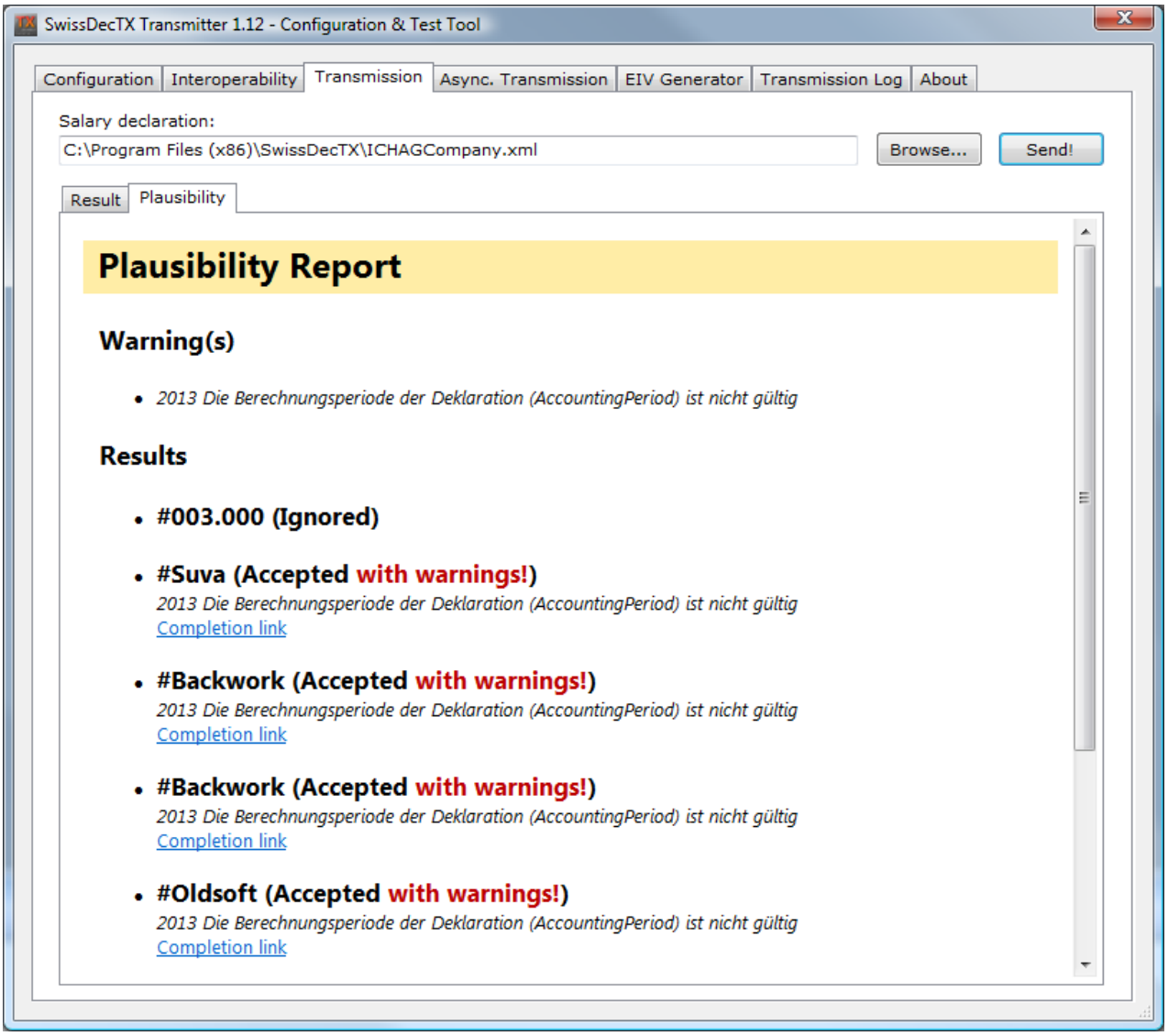
Test de transmission :



Cette page se passe d'explications : il suffit d'indiquer le nom du fichier contenant la déclaration à transmettre et de cliquer le bouton « *Send!* ». Le résultat devrait s'afficher comme ci-dessus tandis que le résultat de plausibilité s'affiche dans son propre onglet :

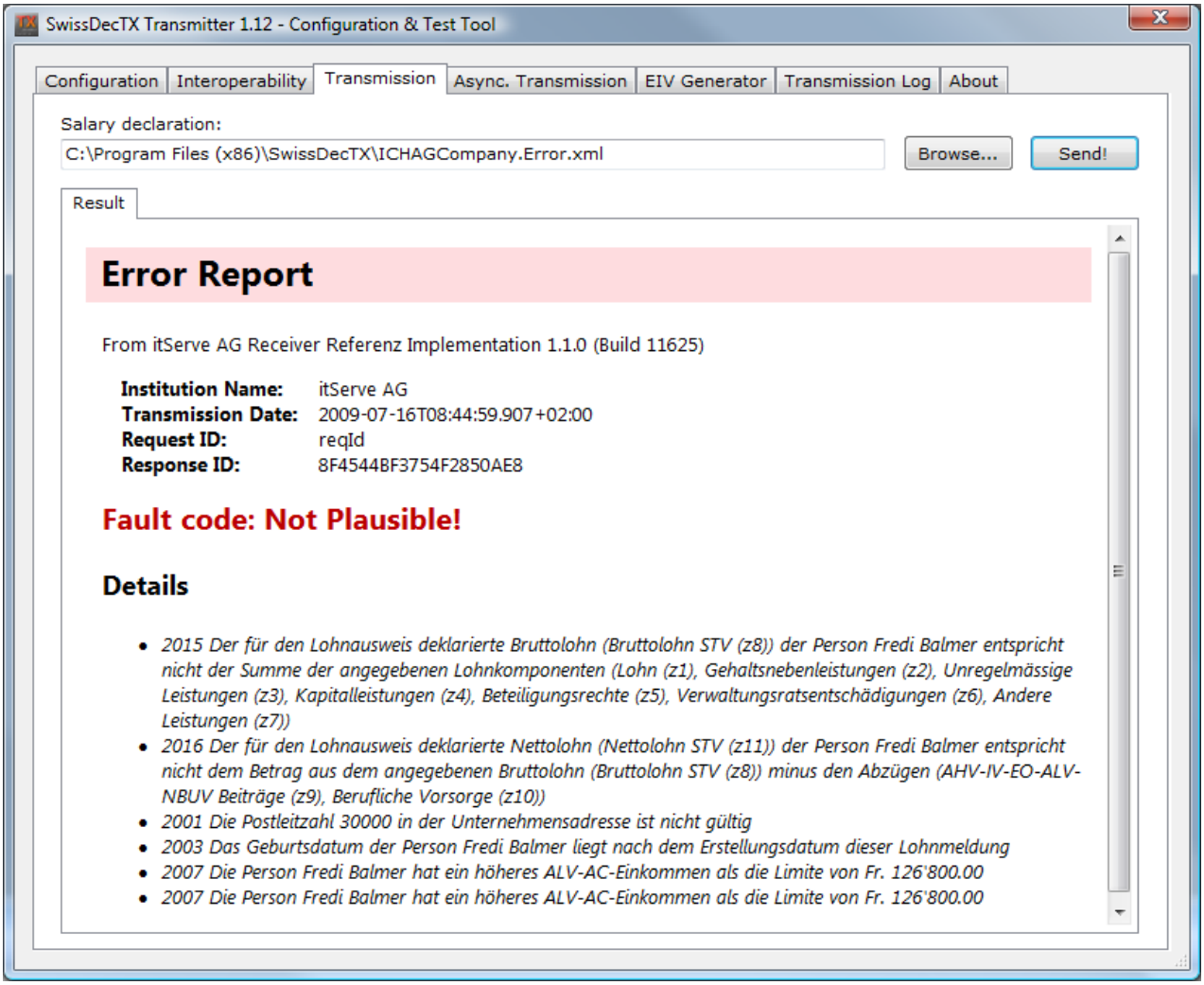

SwissDecTX Transmitter 1.12 - Configuration & Test Tool
Configuration
Interoperability
Transmission
Async. Transmission
EIV Generator
Transmission Log
About
Salary declaration:
C:\Program Files (x86)\SwissDecTX\ICHAGCompany.2009.xml
Browse...
Send!
Result
Plausibility
Plausibility Report
Results
#003.000 (Ignored)
#Suva (Accepted)
Completion link
#Backwork (Accepted)
Completion link

En cas d'erreur de type « logique » la plausibilité contiendra des messages d'avertissement :



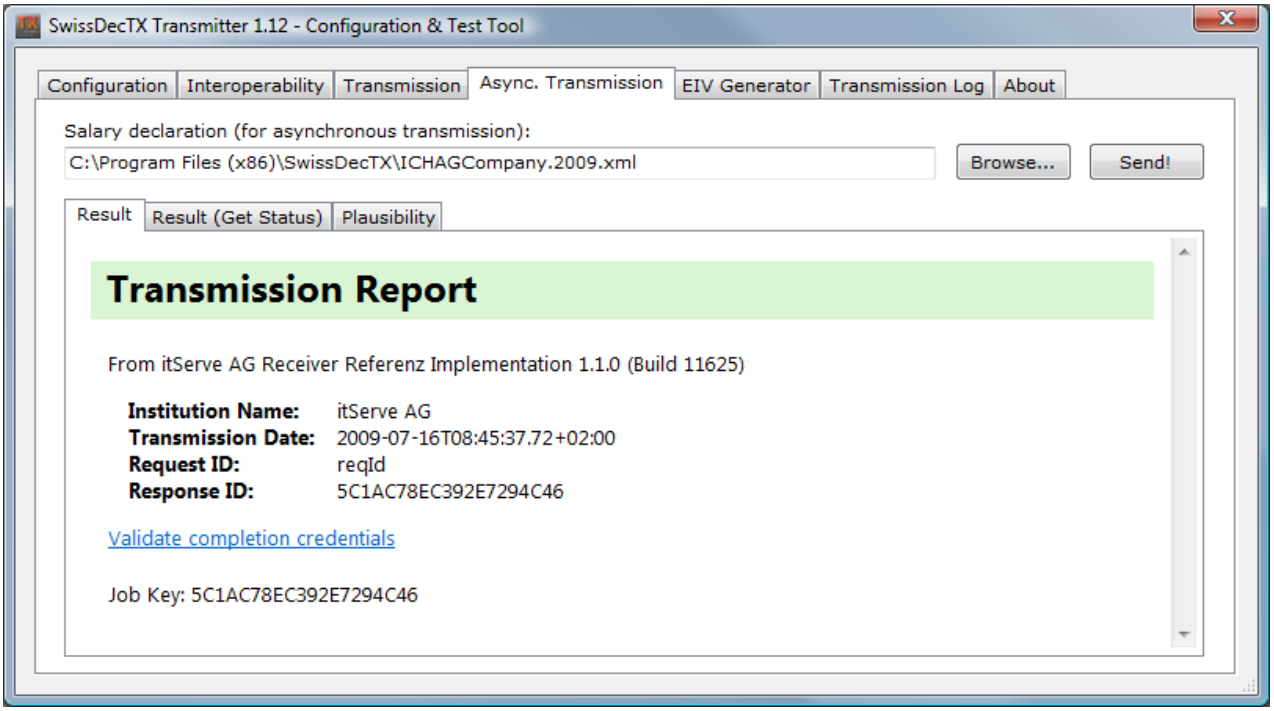
Les messages ci-dessus correspondent à l'envoi de la déclaration « ICHAGCompany.xml » qui date de 2004 : les différent services font remarquer que la période comptable n'est pas valable. (Notez au passage que la fenêtre du programme te test et configuration peut être agrandie).

En cas d'erreur grave dans les données ou d'erreur technique (réseau déconnecté…), la transmission n'est pas acceptée (ou ne peut avoir lieu) et l'outil affiche alors un rapport d'erreur, par exemple comme suit :



Transmission « PIV » asynchrone :

L'interface de test du mode de transmission PIV asynchrone est tout à fait similaire :



Le premier onglet montre le résultat de la commande *DeclareSalaryDeferred*, tandis que le second affiche le résultat de la commande *GetStatusFromDeferredDeclaration*. Dans ce test les deux commandes sont appelées l'une après l'autre mais dans le cas normal l'application sauve le « Job Key » et récupère le statut dans un second temps (typiquement dans les 24 heures). Il s'agit ici d'une fonction de test donc le programme récupère le résultat asynchrone immédiatement. Dans le cas normal, il convient d'afficher et de stocker la déclaration, le RequestID et le JobID afin de pouvoir les passer plus tard a *GetStatusFromDeferredDeclaration* pour récupérer les résultats.

Les fichiers utilisés dans les exemples ci-dessus sont dans un fichier .zip appelé « *SwissDecTX_TestFiles.zip* », fourni avec le transmetteur.

Les informations affichées sont extraites des réponses du serveur (retournées par le transmetteur dans les paramètres (ou fichiers XML) « *result* » et « *plausibility* »). L'outil de test et de configuration utilise une transformation XSLT pour convertir les réponses en HTML, lequel est ensuite affiche dans un « browser control », le tout en quelques lignes de code.

Il y a certainement d'autres façons d'afficher ces résultats, par exemple sous forme de liste ou de tableau – selon les besoins et le style de votre application – mais la méthode de la conversion HTML et l'affichage dans un browser offre des avantages certains : il est possible de styler le HTML comme bon vous semble (couleurs, polices etc), d'autre part les messages retournés par le serveur étant tous optionnels, de quantité et de longueur variable, l'affichage correct des information dans une interface « rigide » peut présenter un challenge, sachant également que les informations contiennent des URLs à cliquer, que l'utilisateur devra visiter pour confirmer et libérer les données.

Afin de faciliter l'usage de transformations XSLT également depuis les langages de programmation et les environnements de développement ne disposant pas des outil nécessaires (à savoir un processeur XSLT intégré) , nous avons décidé d'inclure un composant « utilitaire » au transmetteur.

Ce composant se présente sous la forme d'une classe appelée « *SwissDecTX.TransmitterUtilities* », documentée dans le fichier d'aide « *SwissDecTX.chm* » attenant à la documentation du transmetteur, dont les fonctions sont aussi exposées sous forme de class « COM » et également par l'interface « DLL » et par l'outil « *command-line* » (voir le mode « XSLT ») – on peut donc bénéficier de ces fonctions depuis tous les environnements de développement.

La sémantique est très simple : on passe 3 paramètres, le document d'entrée (contenant soit le résultat de la transmission, soit la plausibilité), suivi de la feuille de style XSLT gouvernant la transformation et enfin le document de destination. L'appel s'effectue donc en une ligne de code et on obtient en sortie du HTML prêt à être affiché : il suffit de disposer d'un « contrôle » HTML dans son environnement de développement.

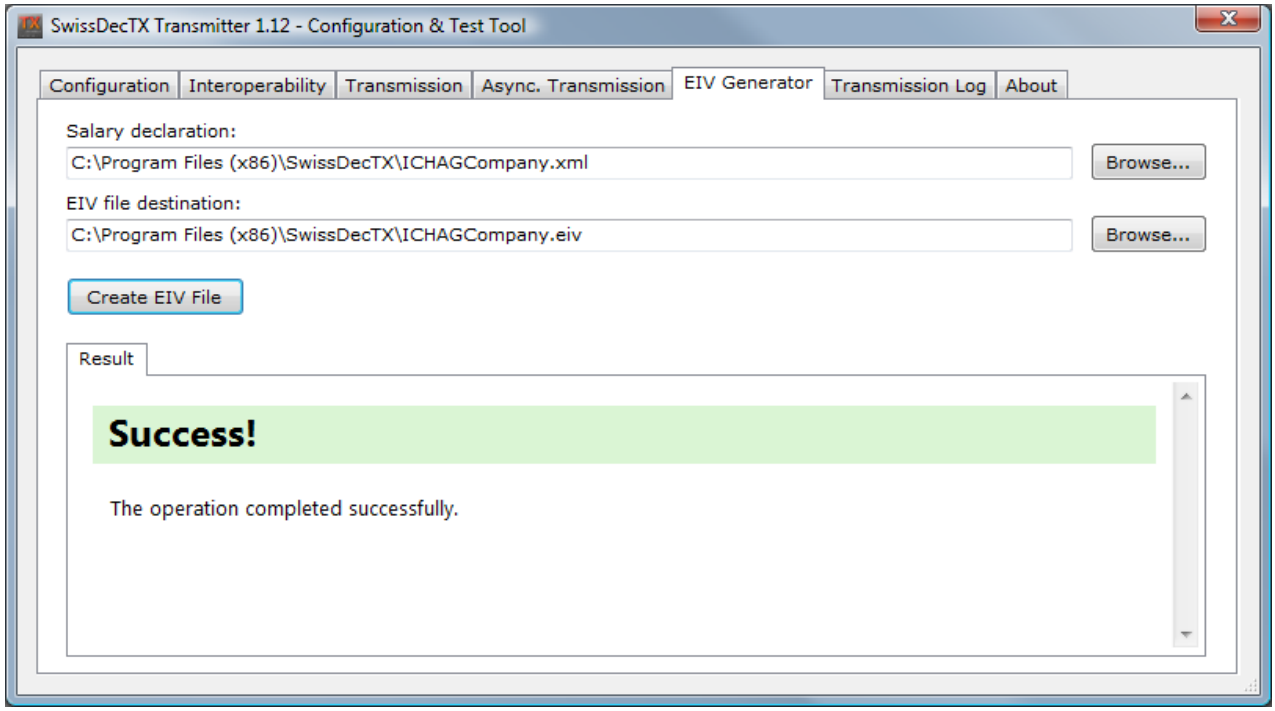
L'affichage ainsi produit est admissible pour la certification SwissDec.

Le composant utilitaire contient une feuille de style intégrée (quatre en fait, une pour chaque langue nationale) et on peut utiliser ces dernières simplement en passant un « null » (ou la chaine vide "") en lieu et place du nom de la feuille de style, ce que fait l'outil de test et de configuration décrit dans ces lignes.

On obtient alors un affichage similaire à celui montré sur les copies d'écran figurant dans ce document. Les titulaires de licence du transmetteur SwissDecTX recevront séparément, sur simple demande, le code-source XSLT de la feuille de style intégrée, leur permettant de la modifier à loisir et de l'inclure dans leur application.

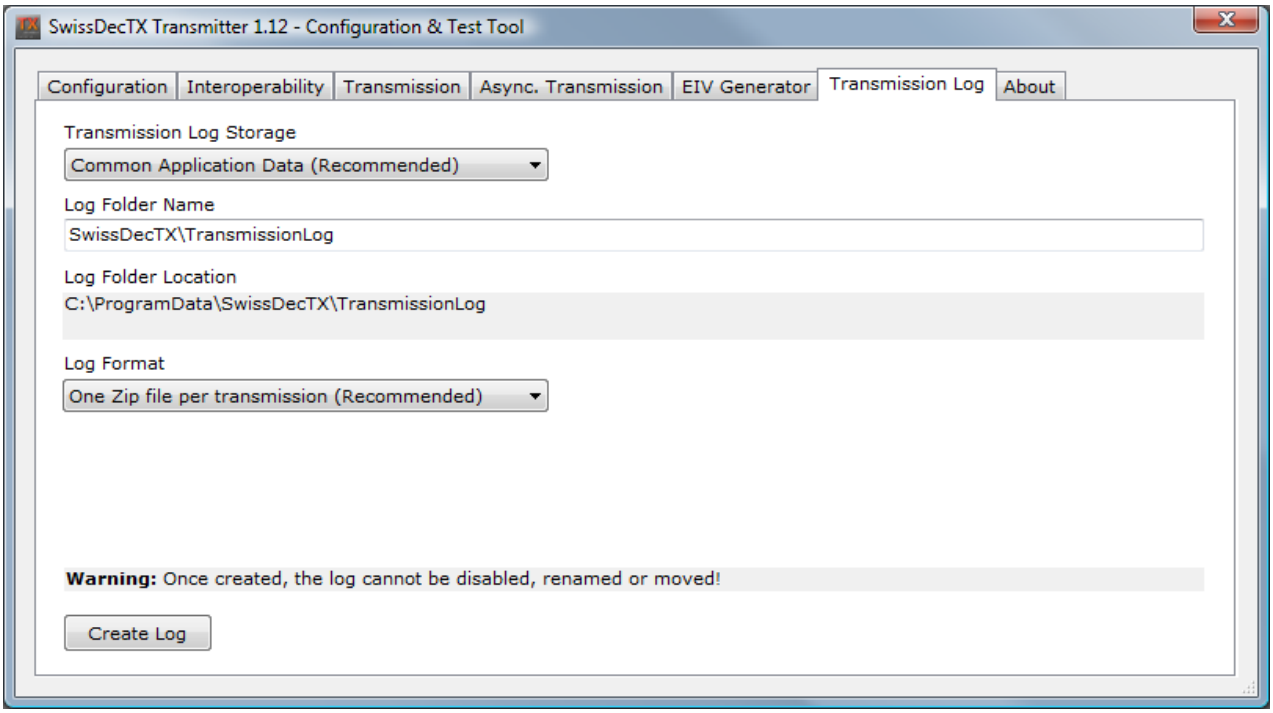
Il est possible d'obtenir un aspect et un formatage entièrement différent en modifiant le style CSS intégré, le code XSLT et / ou le HTML produit.

Génération d'un message EIV :



Afin de tester la fonction de génération de message EIV, l'outil de test et configuration expose une interface utilisateur autour de la fonction *DeclareSalaryLocal* du transmetteur et permet de créer un fichier « EIV » destiné à la transmission manuelle sur le récepteur en-ligne. Les fichiers ainsi crées ont une durée de vie (TTL) de 3600 secondes (contre 300 secondes pour les transmissions PIV).

Configuration de la journalisation automatique des transmissions :



L'outil offre différents choix pour la gestion du log, à savoir :

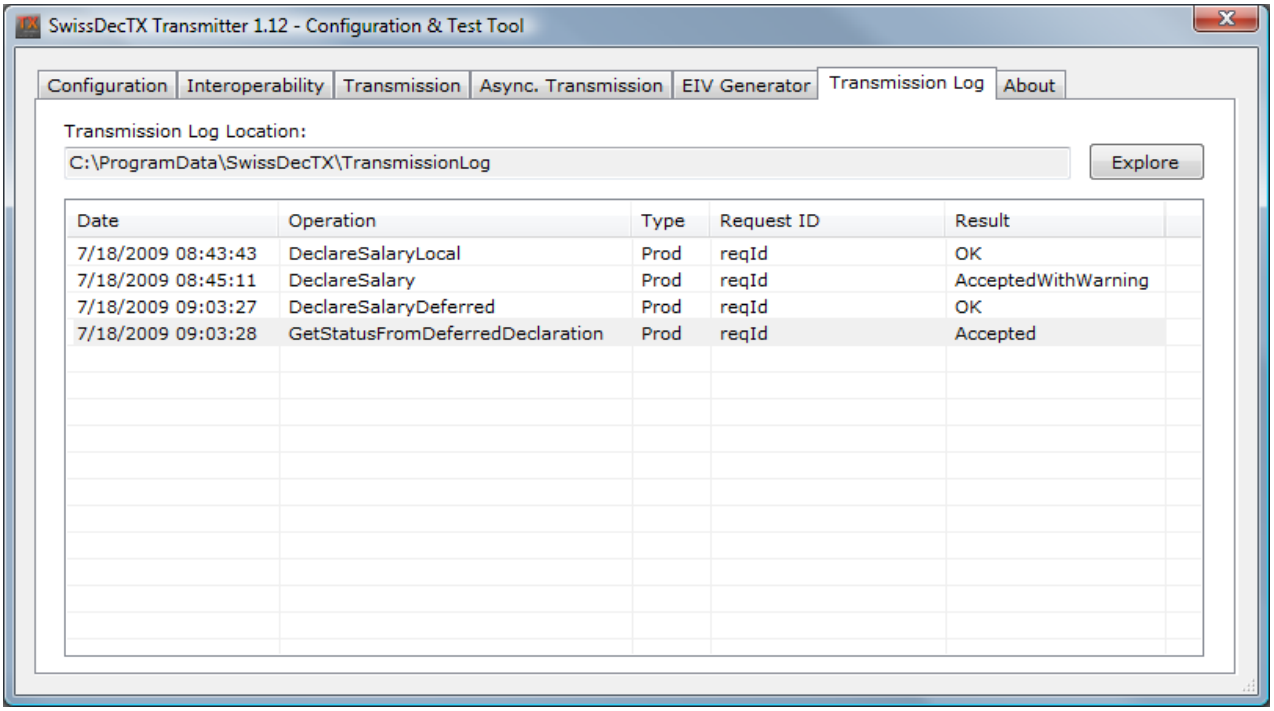
- *Logging is Disabled*
  L'application se charge elle-même du maintien du journal, en archivant les messages transmis et reçus retournes par le transmetteur (*messageSent, messageReceived*).
- *Per User Application Data*
  Le journal sera stocké dans un sous-répertoire lie à l'utilisateur.
- *Common Application Data (Recommended)*
  Le journal sera stocké dans un sous-répertoire commun prévu pour les données des programmes. Il est **très recommandé** de stocker le journal à cet endroit, par exemple dans un sous répertoire du genre VotreSociete\VotreProduit situé sous cet emplacement.
- *Custom Folder Path*
  Le journal sera stocké dans un répertoire spécifié entièrement, p.ex. C:\Compta\Journal

D'autre part on indique également le mode de stockage parmi les choix suivants:

- *One subfolder per transmission, flat files*
- *One Zip file per transmission (Recommended)*

L'option 'Zip' est préférée pour gagner de la place.

Une fois le journal créé les paramètres ne peuvent plus être modifiés et l'interface-utilisateur se mue en la simple indication du répertoire utilisé pour le stockage du journal et d'une liste du contenu du répertoire :



Le journal des transmissions ainsi créé est admissible pour la certification SwissDec.

Pour mémoire, l'application salariale doit implémenter d'une façon ou d'une autre des interfaces-utilisateur autour des fonctions suivantes :

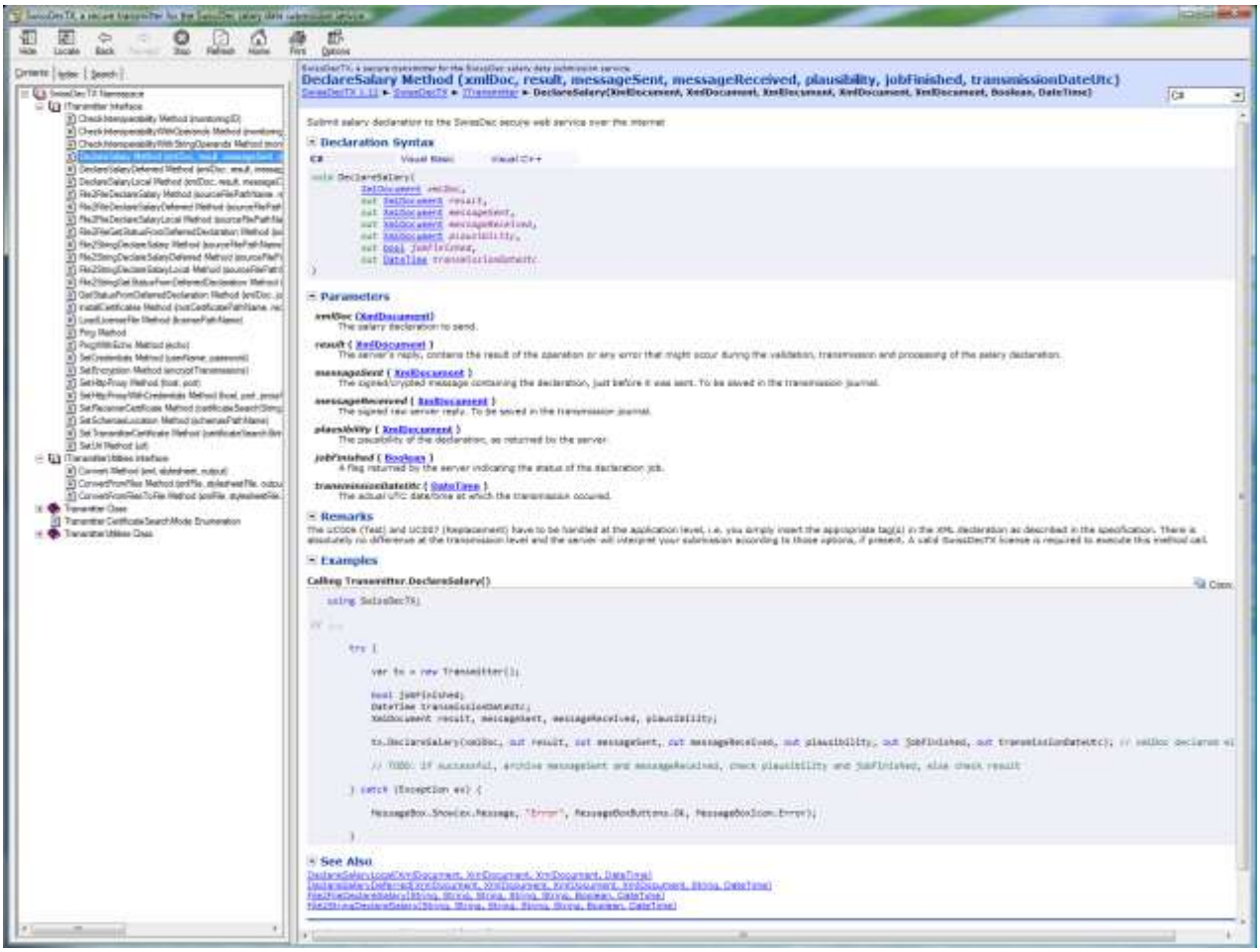
- Test d'interopérabilité (*CheckInteroperabilityWithOperands*) – optionnel : le test d'interopérabilité implémenté dans l'outil de configuration est suffisant et recevable pour la certification.
- **Mode de transmission synchrone** (*DeclareSalary*).
- **Mode de transmission asynchrone** (*DeclareSalaryDeferred/GetStatusFromDeferredDeclaration*).
- **Création des fichiers « EIV »** pour transmission manuelle (*DeclareSalaryLocal*).
- Journalisation des transmissions, archivant tous les messages SOAP envoyés et reçus. Si le mode de journalisation automatique du transmetteur est utilisé, l'application peut se contenter d'afficher la liste des messages journalisés ou ne rien faire du tout, le log étant accessible depuis Windows Explorer, via l'outil de test et configuration fourni (et installé par défaut dans chaque machine). Le journal des transmissions intégré est recevable pour la certification.
- Il faut prévoir la configuration du transmetteur (installation des certificats, configuration de l'URL du service, installation de la licence) soit sous forme programmatique soit sous la forme d'instructions écrites à destination de la personne effectuant l'installation de votre programme. L'outil de configuration fourni est suffisant et recevable pour la certification.
- **Affichage des résultats.** La manière proposée en option par le transmetteur (XML->HTML) et décrite dans ces pages est admissible pour la certification.

La plupart des aspects se résument à appeler la fonction appropriée du transmetteur ou du composant auxiliaire et ne devraient pas présenter de difficulté technique particulière. La mise en œuvre et

l'exploitation de SwissDecTX version 3 (satisfaisant aux exigences et au protocole SwissDec 3) est virtuellement identique.

Documentation détaillée relative à la programmation (API)

Un fichier d'aide (SwissDecTX.chm) décrit chaque fonction dans ses moindres détails avec la syntaxe d'appel en C#, VB et C++ ainsi qu'un exemple en pseudo-code pour la plupart des fonctions: consultez-le !



Supplément (API de création XML simplifiée)

Dès la version 1.15, le transmetteur SwissDecTX installe également un composant COM natif (situé dans SwissDecTX.dll) qui expose des fonctions permettant la création de documents XML (telle que la declaratio) de façon relativement aisée. Ce composant ne fait pas partie du transmetteur à proprement parler et n'est pas couvert par le support compris dans la licence : en effet, l'usage de ce composant se fait en amont et indépendamment des transmissions et la création des déclarations XML est du ressort de la comptabilité salariale. Il s'agit donc d'un supplément fourni à bien plaire, dans le but de faciliter la création des documents XML et dont l'usage est entièrement optionnel. Ceci dit, si vous le souhaitez, nous sommes disponibles pour toute assistance et consulting « en amont », par exemple pour vous aider lors de la création des documents ou autres tâches si besoin est.

Ce composant a été créé dans le but de masquer autant que faire se peut la complexité liée à l'utilisation des « parseurs » XML modernes. Il repose sur la présence the Microsoft XML Parser 6.0 qui devra donc

impérativement être installé. Le composant expose des fonctions simples et permet la création d'un document arbitrairement complexe et sa validation au moyen de schémas XSD.

L'archive téléchargeable sur www.softwarecomponents.ch contient un fichier zip (« Samples_VBS_MSXML6.zip ») contentant deux exemples écrits en VB-Script démontrant la création d'une déclaration XML minimale (mais valide) et la transmission d'une déclaration existante. Le formatage des réponses du serveur (plausibilité) est également démontré dans le fichier VB fourni.

En bref, la *création* d'un document au moyen du composant « XML Helper » optionnel se passe comme suit :

```
Set xml = CreateObject("SwissDecTX.XmlUtilities")

xml.CreateDoc "DeclareSalary",
"http://www.swissdec.ch/schema/sd/20051002/SalaryDeclarationServiceTypes",
"Schemas\SalaryDeclaration.xsd", "Schemas\SalaryDeclarationContainer.xsd",
"Schemas\SalaryDeclarationServiceTypes.xsd"

xml.SetAttribute "xmlns:ct",
"http://www.swissdec.ch/schema/sd/20051002/SalaryDeclarationContainer"

xml.SetAttribute "xmlns:sd",
"http://www.swissdec.ch/schema/sd/20051002/SalaryDeclaration"

xml.SetAttribute "xmlns:xsi", "http://www.w3.org/2001/XMLSchema-instance"

xml.StartTag "ct:RequestContext"
  xml.StartTag "ct:UserAgent"
    xml.AddTextTag "ct:Producer", "itServe AG"
    xml.AddTextTag "ct:Name", "Test"
    xml.AddTextTag "ct:Version", "0.1"
    xml.AddTextTag "ct:Certificate", "None"
  xml.EndTag
  'etc...
xml.EndTag

xml.StartTag "ct:Job"
  'etc…
xml.EndTag

xml.Validate
xml.SaveAs "MySalaryDeclaration.xml", True
xml.Cleanup

Set xml = Nothing
```

Le code VB devrait être lisible et facile à transcrire dans n'importe quel langage de programmation. Pour terminer, **un petit transmetteur basique** complet en 4 ou 5 lignes :

```
Set tx = CreateObject("SwissDecTX.Transmitter")

Dim jobFinished, transmissionDateTime

tx.File2FileDeclareSalary "ICHAGCompany.xml", "result.xml",
"messageSent.xml", "messageReceived.xml", "plausibility.xml", jobFinished,
transmissionDateTime

Set tx = Nothing
```

Prière de voir l'exemple fourni avec le transmetteur.