swissdec-Adapter

Benutzerhandbuch

Versionsgeschichte			
Version 1.0	2013-08-07	Marco Stettler	Version 1.0
Version 1.2	2014-01-30	Christoph Brunner	Recovery Tool & Patch Release 2.0_2
Version 1.3	2015-03-02	Marco Stettler	Patch Release 2.0_4

Inhaltsverzeichnis

1. Einführung 1
1.1. swissdec-Adapter 1
1.2. Übersicht swissdec 2
1.3. Schnittstellen 2
1.4. Referenzen 5
2. Anforderungen an den Betrieb 6
2.1. Systemanforderungen 6
2.2. Verzeichnisstruktur 9
2.3. Migration 9
3. Installation 10
3.1. Voraussetzungen prüfen 10
3.2. Rechte zum Ausführen (nur Unix) 10
3.3. Umgebung konfigurieren 11
3.4. Java JCE installieren 11
3.5. Receiver Applikation 11
3.6. Integrationsapplikation 18
3.7. Logging-Konfiguration 20
3.8. Wartungsfenster 20
3.9. Schlussbemerkung 21
4. Installation testen 22
4.1. Installation 22
4.2. Konfiguration 22
4.3. Ausführen der Tests 23
5. Starten des Adapters 24
5.1. Manueller Start 24
5.2. Starten als Windows-Dienst 24
5.3. Unix Init-Skript 26
6. Konfigurations-Referenz 28
6.1. Tomcat Konfiguration 28
6.2. Receiver Konfiguration 28
6.3. Integrationsapplikation Konfiguration 29
7. Monitoring 31
7.1. Eingebautes Monitoring 31
8. Recovery-Tool 33
8.1. Programmstart 33
8.2. Optionen 33
9. Probleme und Lösungen 38
A. Anhang 40
A.1. Referenzierte Dokumente 40
A.2. Glossar 40
B. Beispiele 41
B.1. Konfiguration 41

Abbildungsverzeichnis

1.1. Installationsskizze swissdec-Adapter 2
1.2. Dateisystemschnittstelle swissdec-Adapter - sM-Client 3
3.1. Security Overview 13
3.2. Webservice Security Overview 14
3.3. SSL Handshake 16
3.4. SSL Handshake mit Mutual Authentication 16
3.5. Transport Layer Security Overview 17
5.1. tomcat6w.exe, Tomcat Service Properties 25
5.2. Tomcat Service Java Properties 25
5.3. Windows Dienst Verwaltung 26

Tabellenverzeichnis

1.1. Pfade der Schnittstelle swissdec-Adapter - sM-Client 3
1.2. swissdec-Adapter Verzeichnisse 4
2.1. swissdecAdapter.zip 9
3.1. swissdec-Adapter Receiver Grundkonfiguration 12
3.2. Legende zu Abbildung Security Overview 13
3.3. Webservice Security Signature 14
3.4. Webservice Security Encryption 15
3.5. swissdecAdapter Webservice Security Konfiguration 15
3.6. swissdecAdapter Transport Layer Konfiguration 17
3.7. swissdec-Adapter Integration Grundkonfiguration 19
3.8. Log-Dateien 20
4.1. TestTool Konfiguration 22
6.1. swissdec-Adapter Receiver Konfiguration 28
6.2. swissdec-Adapter Integration Konfiguration 29
8.1. Recovery Tool, Generelle Optionen 33

Liste der Beispiele

3.1. Standardconnector- Konfiguration Tomcat 12
3.2. Tomcat Connector Konfiguration für SSL-Terminierung mit Mutual Authentication 17
3.3. Standardconnector- Konfiguration Tomcat 18
B.1. Standardkonfiguration Receiver 41
B.2. Standardkonfiguration Integration 42
B.3. Standardkonfiguration Tomcat Receiver 43
B.4. Standardkonfiguration Tomcat Integration 44
B.5. Standardkonfiguration Logging 45

1. Einführung

Dieses Dokument beschreibt die *swissdec* -Adapter Installation, Konfiguration und den Betrieb. Der swissdec-Adapter ist eine Java Applikation, bestehend aus zwei Webapplikationen, um swissdec Meldungen für die Domänen Tax (Lohnausweise) sowie TaxAtSource (Quellensteuerabrechnungen) auszutauschen. Er wurde zur Integration mit dem *sM-Client* entwickelt, kann jedoch auch standalone betrieben werden.

1.1. swissdec-Adapter

Der swissdec-Adapter bietet folgende Funktionalität:

- Empfang und Versand von protokollkonformen *swissdec*-Meldungen
- sicherer Transport der Lohndaten mittels SSL/TLS und WS-Security
- Integritätsprüfung mittels WS-Security Signatur

Anmerkung

Die swissdec Plattform ist synchron. Das bedeutet, dass jede Meldung von einem Unternehmen vom Distributor sofort verarbeitet und an die Endempfänger weitergeleitet wird. Das Unternehmen (Lohndatenquelle) erlebt das ganze System somit als Einheit. Sollte ein End-Empfänger nicht in geforderter Qualität betrieben werden, vermindert dieser Empfänger die Zuverlässigkeit des ganzen Systems. Alle Teilnehmer müssen sich deshalb auf eine minimale Zuverlässigkeit einigen.

Wir möchten eine kundenorientierte Sicht einnehmen, das heisst eine Internetlösung wird sich nicht nach üblichen Bürozeiten richten. Andererseits müssen beim Betrieb eines Endempfängers die ökonomische Aspekte (Finanzierbarkeit) ebenfalls berücksichtigt werden.

Die Verfügbarkeiten der Systeme sind daher als zukünftige Zielwerte zu verstehen, das heisst die Bedeutung der Lösung nimmt zu und damit auch seine Verfügbarkeit. Ziel ist eine pragmatische Lösung ("lightweight construction" und "best effort"). Definierte Zeitbereiche:

- 7 Tage pro Woche mal 24 Stunden
- Spitzenzeiten: 6 Uhr bis 20 Uhr

Definierte Wertebereiche:

- Spitzenzeiten: Verfügbarkeit der Endempfänger (m2m) mindestens 99,52%
- Randzeiten: Verfügbarkeit der Endempfänger (m2m) mindestens 93,00%

1.2. Übersicht swissdec

Die swissdec-Plattform ist eine Service orientierte Client - Server Plattform mit dem Distributor als Intermediär. Folgendes Diagram gibt eine kurze Übersicht über die swissdec Architektur:

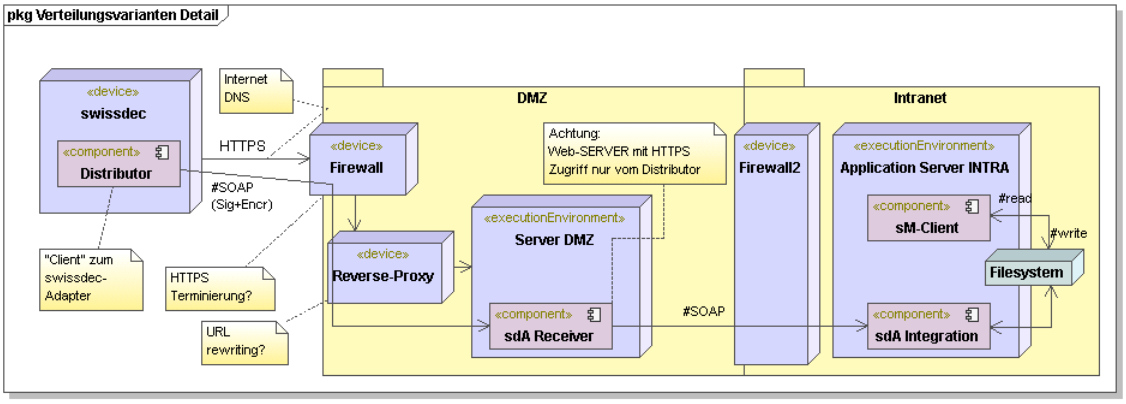


Abbildung 1.1. Installationsskizze swissdec-Adapter

1.3. Schnittstellen

1.3.1. swissdec

Die swissdec-Schnittstelle wurde im itServe Produkt STEP bereits implementiert und bei mehreren Kunden installiert sowie von der swissdec abgenommen. Die swissdec Dokumentation ist auf der Webseite [http://www.swissdec.ch] des Vereins verfügbar.

1.3.2. sM-Client

Zwischen dem swissdec-Adapter und dem *sM-Client* kommt eine Filebasierte-Schnittstelle zum Einsatz. Der Adapter schreibt die eingehenden Meldungen in ein Inbox-Verzeichnis, das vom *sM-Client* gepollt wird.

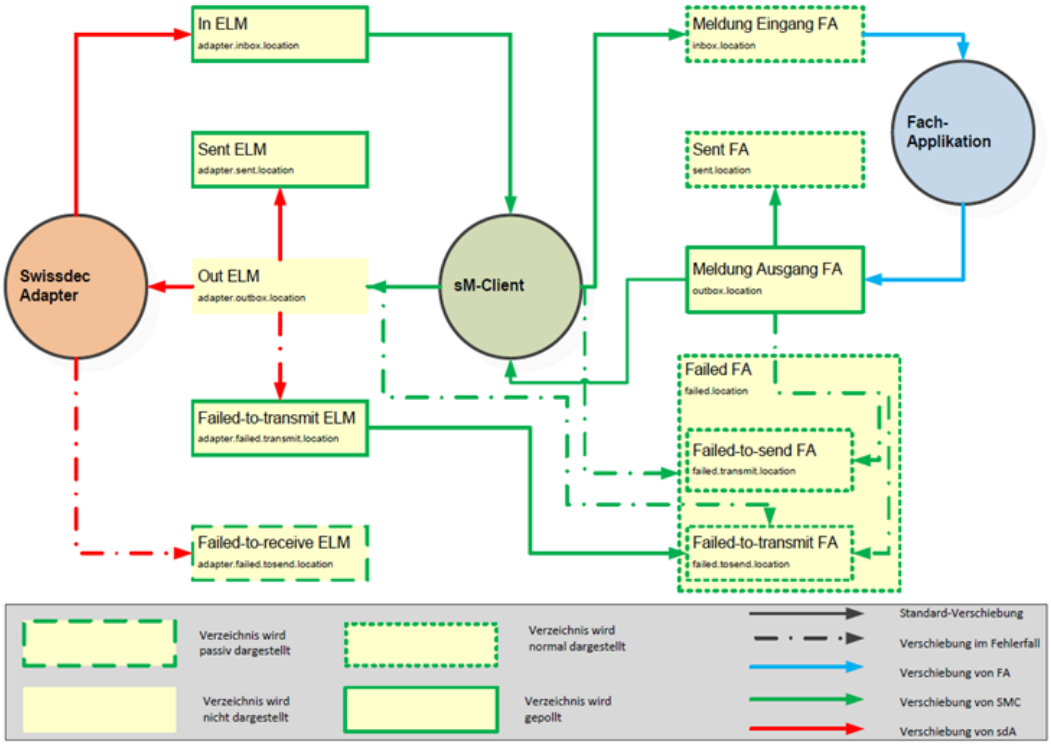


Abbildung 1.2. Dateisystemschnittstelle swissdec-Adapter - sM-Client

sM-Client (Message-handler-elm.properties)	**PathID**	**swissdec-Adapter (swissdecAdapter.properties)**
Im sM-Client sind diese Verzeichnisse relativ zu base.dir.adapter		sdA braucht absoluten Pfad in dieser Konfigurationsdatei
adapter.inbox.location	PATH_ELM_IN	released.filesystem.parent.xml
adapter.outbox.location	PATH_ELM_OUT	result.filesystem.parent.xml
adapter.sent.location	PATH_ELM_SENT	sent.filesystem.parent.xml
adapter.failed.receive	PATH_ELM_FAILED_TO_RX	failed.filesystem.parent.xml
adapter.failed.transmit.location	PATH_ELM_FAILED_TO_TX	undeliverable.filesystem.parent.xml

Tabelle 1.1. Pfade der Schnittstelle swissdec-Adapter - sM-Client

PathID	Beschreibung
PATH_ELM_IN	• sind im swissdec XML-Standard (Schemavalid) • haben den Dateinamen: ["Tax"|"TaxAtSource"]_DeclarationId_ ["MIXD"|"SPLT"]_Timestamp.zip • konsumierte Meldungen werden vom sM-Client verschoben/gelöscht
PATH_ELM_OUT	• Antworten werden nur für Quellensteuerabrechnungen erstellt (Quellensteuerabrechnungsresultat) • sind im swissdec XML-Standard (Schemavalid), gezippt • haben den Dateinamen: DeclarationId.zip • konsumierte Meldungen werden vom swissdec-Adapter verschoben • nicht konsumierte Meldungen werden vom swissdec-Adapter verschoben
PATH_ELM_SENT	• Versendete Quellensteuerabrechnungsresultate • sind im swissdec XML-Standard (Schemavalid), gezippt • haben den Dateinamen: DeclarationId.zip • wird vom sM-Client gelesen, um den Prozess abschliessen zu können
PATH_ELM_FAILED_TO_RX	• Lohnausweis und QST-Abrechnung, Fehler beim Empfang • die Antworten des swissdec-Adapters werden auch geschrieben • nicht in jedem Fall vorhanden
PATH_ELM_FAILED_TO_TX	• QST-Abrechnungsresultate, Fehler beim Versand

Tabelle 1.2. swissdec-Adapter Verzeichnisse

Die komplette Schnittstellenspezifikation entnehmen Sie bitte der [DETAILSPEZ]

1.4. Referenzen

- [ENDRECREQ] swissdec Anforderungen Endempfänger
- [DETAILSPEZ] Detailspezifikation swissdec-Adapter

2. Anforderungen an den Betrieb

Der swissdec-Adapter besteht aus zwei Java Applikationen. Beide Applikationen (Receiver und Integration) sind als Java-Webapplikationen implementiert und benötigen einen Java Servlet Container. Das Installationspaket beinhaltet einen vorkonfigurierten Apache Tomcat und kann daher einfach entpackt werden.

Die Receiver-Applikation implementiert das swissdec Protokoll und muss aus dem Internet via HTTPS erreichbar sein. Typischerweise wird diese in der *DMZ* installiert.

Die Integrationsapplikation stellt die Kommunikation mit dem *sM-Client* sicher. Diese Schnittstelle ist dateisystembasiert. Das heisst, die Integrationsapplikation und der *sM-Client* benötigen ein gemeinsames Dateisystem. Die Integrationsapplikation kann sowohl auf dem gleichen System wie die Receiver als auch auf einem dedizierten System oder dem System des *sM-Client*s installiert werden.

Die Kommunikation zwischen den beiden Applikation geschieht mittels HTTP/SOAP. Es muss sichergestellt werden, dass eine solche Verbindung von der Receiver-Applikation (Client, initiiert immer die Verbindung) zur Integrationsapplikation geöffnet werden kann.

2.1. Systemanforderungen

2.1.1. Unterstützte Platformen

Der swissdec-Adapter ist eine reine Java Applikation. Er läuft unter allen Plattformen, die Java in der Version 1.6 unterstützen. Getestete und unterstützte Plattformen:

- Windows 2008 Server 64bit
- Windows 7
- Ubuntu 12.04 LTS

2.1.2. Als Dienst starten

Aufgrund der synchronen swissdec Architektur muss der swissdec-Adapter permanent laufen. Um dies sicherzustellen sollte der swissdec-Adapter als Service (auf Windows) oder Start-Skript (auf Unix) konfiguriert werden.

2.1.3. CPU

Die Anforderungen an die CPU sind gering. Es wird der Einsatz von aktuellen 64-Bit Multi-Core Prozessoren empfohlen (Intel Core i5/i7, Intel Xeon, oder ähnliche).

Anmerkung

Durch die swissdec-Architektur ergibt sich die Problematik, dass der Distributor auf alle Endempfänger warten muss, bevor die Quittung an den Kunden gesendet wird. Die swissdec hat deshalb folgende Perfomance-Anforderungen spezifiziert:

- [ENDRECREQ], Kapitel 3.10: Eine Lohnmeldung mit 100 Personen sollte in weniger als 20 Sekunden verarbeitet sein.
- [ENDRECREQ], Kapitel 3.10: Eine Lohnmeldung mit 100 Personen muss in einer Minute verarbeitet werden können.
- [ENDRECREQ], Kapitel 3.10: Eine Lohnmeldung mit 2000 Personen sollte in einer Minute verarbeitet sein.

2.1.4. RAM

Es sollten mindestens 1 Gigabyte für den swissdec-Adapter reserviert werden. Damit lassen sich Meldungen bis etwa 6'000 Mitarbeiter empfangen. Falls grössere Meldungen empfangen werden sollen, empfehlen wir 4 Gigabytes zu reservieren. Damit lassen sich Meldungen bis etwa 20'000 Mitarbeiter empfangen.

2.1.5. Festplatte

Die Installation benötigt in etwa 150 Megabytes. Der benötigte Festplattenplatz zur Laufzeit variert deutlich und ist abhängig von der Anzahl und der Grösse der empfangenen Meldungen. Als Startwert wird 1 Gigabyte freier Festplattenplatz empfohlen.

2.1.6. Grössenbeschränkungen

Das zur Verfügung stehende RAM, beziehungsweise der Java Virtual Machine zugewiesene Memory beschränkt die Grösse der Meldungen. Um die Signatur der Meldungen prüfen beziehungsweise erstellen zu können, muss die gesamte Meldung im Arbeitsspeicher gehalten werden.

Anmerkung

- 32Bit JVM ist auf 1.6 GB limitiert
- Eine Meldung mit 6'000 Personen benötigt etwa 1 GB Memory
- Eine Meldung mit 20'000 Personen benötigt etwa 4 GB Memory

2.1.7. Netzwerk

Der swissdec-Adapter startet einen Webservice-Endpunkt, der aus dem Internet erreichbar sein muss. Es muss also ein Pfad aus dem Internet zum swissdec-Adapter geschaffen werden. Dazu müssen folgende Dinge konfiguriert werden:

- URL: Um aus dem Internet erreichbar zu sein, benötigt der swissdec-Adapter zwei URLs. Die swissdec schreibt vor, dass pro Domäne (Quellensteuer/TaxAtSource, Steuern/ Tax/Lohnausweis) eine eindeutige Adresse vergeben werden muss. Mehr dazu im Abschnitt 2.1.7.2, „Reverse Proxy“.

- SSL/TLS: Um die Verbindung abzusichern, schreibt die swissdec SSL/TLS mit Klienten-Authentifizierung vor. Das heisst nebst „normalem“ HTTPS muss vom Klienten ein Zertifikat verlangt und geprüft werden. Mehr dazu im Abschnitt 2.1.7.3, „SSL/TLS“.

- Firewall: Der swissdec-Adapter öffnet keine Verbindungen ins Internet. Daher muss nur der eingehende Pfad konfiguriert werden. Die Verbindung kann auf den Distributor eingeschränkt werden. Mehr dazu im Abschnitt 2.1.7.1, „Firewall“.

2.1.7.1. Firewall

Es müssen eingehende Verbindungen auf den swissdec-Adapter erlaubt werden. Per default wird der swissdec-Adapter auf Port 8080 gestartet. Siehe dazu auch Kapitel 6, *Konfigurations-Referenz*.

Anmerkung

Da der swissdec-Adapter von einem Klienten (Distributor) bedient wird, kann die eingehende Verbindung auf dessen IP restriktiert werden. Für den produktiven Distributor ist dies die IP:

- 194.11.148.11

Der Testdistributor der swissdec Referenz Applikationen hat die IPs:

- 193.247.121.163

  193.247.102.165

2.1.7.2. Reverse Proxy

Falls ein Reverse Proxy im Einsatz ist, müssen folgende URLs von aussen erreichbar sein:

```
http://<HOST>:<PORT>/webservice/tax/SalaryDeclarationConsumerService
```

```
http://<HOST>:<PORT>/webservice/tas/SalaryDeclarationConsumerService
```

2.1.7.3. SSL/TLS

Die swissdec schreibt für Endempfänger nebst „normalem“ HTTPS vor, dass der Klient (Distributor) sich authentifizieren muss. Dies als Schutz der Endempfänger. Grundsätzlich liegt die Entscheidung, wie und wo das SSL/TLS mit Mutual Authentication terminiert wird, beim Endempfänger (Betreiber). Folgende Lösungen sind dabei denkbar:

- Die Terminierung findet auf dem Entry-Server oder der Firewall statt.
- Die Terminierung findet auf dem Tomcat des swissdec-Adapters statt. Die Konfiguration hierzu finden Sie im Anhang.

Die Zertifikate für beide Lösungen erhalten Sie via *sM-Client* Support (https://jira.ctp-consulting.com/browse/SMCSUPPORT).

2.1.8. Systemzeit

Es ist wichtig, dass die Systemzeit des Systems, auf dem der swissdec-Adapter Receiver läuft, korrekt ist. Dies weil auf Web Service Security Ebene ein Timestamp generiert wird. Wenn die Zeit des Distributors und des Endempfängers auseinander driftet, werden Meldungen abgelehnt. Es wird empfohlen, die Zeit automatisch mittels *NTP (Network Time Protocol)* zu aktualisieren.

2.2. Verzeichnisstruktur

Die folgende Tabelle zeigt die Verzeichnisstruktur einer installierten swissdec-Adapter Applikation (Receiver oder Integration):

Verzeichnis	**Beschreibung**
bin	Skripts zum Starten und Stoppen des swissdec-Adapters
conf	Konfigurationsdateien
derbydb	Eingebettete Datenbank für den swissdec-Adapter, nur Integrationsapplikation
jce	Java Crypto Extensions für Oracle/Sun und IBM JDK der Version 1.6.
lib	Zusätzliche Java-Bibliotheken
logs	Logfiles
temp	Temporäre Dateien zur Laufzeit
webapps	Die Webapplikationen zum swissdec-Adapter
work	Temporäre Dateien zur Laufzeit

Tabelle 2.1. swissdecAdapter.zip

2.3. Migration

Da die Version 2.0 des swissdec-Adapters der erste produktive Release ist, empfiehlt sich auch für die Pilot-Teilnehmer eine Neuinstallation. Eine Migration der Pilotumgebung ist nicht vorgesehen.

3. Installation

Der swissdec-Adapter besteht aus zwei Java Applikationen. Beide Applikationen (Receiver und Integration) sind als Java-Webapplikationen implementiert und benötigen einen Java Servlet Container. Das Installationspaket beinhaltet einen vorkonfigurierten Apache Tomcat und kann daher einfach entpackt werden.

Die Receiver-Applikation implementiert das swissdec Protokoll und muss aus dem Internet via HTTPS erreichbar sein. Typischerweise wird diese in der *DMZ* installiert.

Die Integrationsapplikation stellt die Kommunikation mit dem *sM-Client* sicher. Diese Schnittstelle ist dateisystembasiert. Das heisst, die Integrationsapplikation und der *sM-Client* benötigen ein gemeinsames Dateisystem. Die Integrationsapplikation kann sowohl auf dem gleichen System wie die Receiver als auch auf einem dedizierten System oder dem System des *sM-Client*s installiert werden.

Die Kommunikation zwischen den beiden Applikation geschieht mittels HTTP/SOAP. Es muss sichergestellt werden, dass eine solche Verbindung von der Receiver-Applikation (Client, initiiert immer die Verbindung) zur Integrationsapplikation geöffnet werden kann.

Anmerkung

Wenn die Applikationen auf verschiedenen Systemen installiert werden, müssen die folgenden Kapitel für beide Systeme berücksichtigt werden.

3.1. Voraussetzungen prüfen

Bitte prüfen Sie vor der Installation wie im Kaptiel 2.1 Systemvorausetzungen beschrieben die Systeme und nehmen Sie wenn nötig Änderungen vor.

Auf dem System muss ein JDK (Java Development Kit) der Version 1.6 oder 1.7 installiert sein.

3.2. Rechte zum Ausführen (nur Unix)

Wenn Sie den swissdec-Adapater unter Unix installieren, sind folgende zusätzlichen Schritte zwingend. Diese sind für beide Applikationen (Receiver und Integration) nötig.

1. Wechseln Sie in das Verzeichnis „<SWISSDECADAPTER_HOME>/bin“

2. Wechseln Sie die Berechtigungen der Skripte auf „ausführbar“

```
chmod +x *.sh
```

3.3. Umgebung konfigurieren

Die folgenden Umgebungsvariabeln müssen gesetzt sein:

- JAVA_HOME – Installationsverzeichnis des JDK 1.6 / 1.7

Windows

```
set JAVA_HOME=C:\Program Files\Java\jdk1.6.0_09
```

UNIX (Variiert nach verwendeter Konsole, dieses Beispiel ist für bash)

```
export JAVA_HOME=/usr/java/jdk1.6.0_09
```

Normalerweise wird diese Umgebungsvariable bei der Installation des JDK gesetzt, sowohl in Windows wie in UNIX.

3.4. Java JCE installieren

Aufgrund der Amerikanischen Gesetzgebung ist es nicht erlaubt, dass Oracle/IBM in Java Tools für die „unlimitierte Kryptographie“ im Installationspaket mitliefert. Da der swissdec- Adapter auf diese starke Kryptographie angewiesen ist, müssen die „Unlimited Strength Java(TM) Cryptography Extension Policy Files“ nachinstalliert werden. Das swissdec-Adapter Paket enthält diese Dateien für die Version 1.6 und 1.7 im Verzeichnis

- jce/oracle6
- jce/oracle7
- jce/ibm

Um diese Dateien zu installieren kopieren Sie alle Dateien im gewünschten Verzeichnis (JDK-implementationsspezifisch) in das Verzeichnis

- <JDK_HOME>/jre/lib/security

Diese Dateien können vom Hersteller der JDK-Implementation (Oracle/Sun oder IBM) via Download bezogen werden.

Anmerkung

Das OpenJDK Projekt liefert die unlimitierte Kryptographie im JDK mit. Wenn Sie OpenJDK einsetzen, sind keine zusätzlichen Installationsschritte notwendig.

3.5. Receiver Applikation

3.5.1. Tomcat Konfiguration

Der mitgelieferte Servlet-Container "Tomcat" kann bis auf eine Ausnahme mit der mitgelieferten Konfiguration gestartet werden. Bitte prüfen Sie, ob der Standardport 8080 auf dem Computer

noch nicht belegt ist. Falls dies der Fall sein sollte, bitte im <SWISSDECADAPTER_HOME>/conf/server.xml im Abschnitt "Connector" den Wert für "port" auf einen freien Port ändern. Diese Option gibt den TCP-Port an, auf dem der Tomcat horcht und eingehende Verbindungen erwartet.

```
<Connector port="8080" protocol="HTTP/1.1"
   connectionTimeout="20000"
   redirectPort="8443"/>
```

Beispiel 3.1. Standardconnector- Konfiguration Tomcat

3.5.2. Receiver Konfiguration

Sobald das ZIP-Archiv entpackt ist, müssen Sie die Konfiguration anpassen. Im Kapitel 6, *Konfigurations-Referenz* finden Sie eine komplette Liste aller möglichen Einstellungen. Der folgende Abschnitt beschreibt die Pflichtkonfiguration, die in der Datei <SWISSDECADAPTER_HOME>/conf/swissdecAdapter.properties vorgenommen werden muss.

Option	**Beschreibung**
userAgent.institutionName	Deskriptiver Name der Institution (zum Beispiel „KSTV Bern“)
institution.tas.id	Technische Angabe, für welche Institution QST- Meldungen empfangen werden (zum Beispiel „BE“, „ZH“, „GE“, ...). MUSS richtig konfiguriert sein, da sonst Meldungen abgelehnt werden
institution.tax.id	Technische Angabe, für welche Institution Lohnausweise empfangen werden (zum Beispiel „BE“, „ZH“, „GE“, ...). MUSS richtig konfiguriert sein, da sonst Meldungen abgelehnt werden
salarydeclaration.service.host	Der Host, auf dem die swissdec-Adapter Integration läuft.
salarydeclaration.service.port	Der Port, auf dem die swissdec-Adapter Integration läuft.

Tabelle 3.1. swissdec-Adapter Receiver Grundkonfiguration

3.5.3. Security Konfiguration

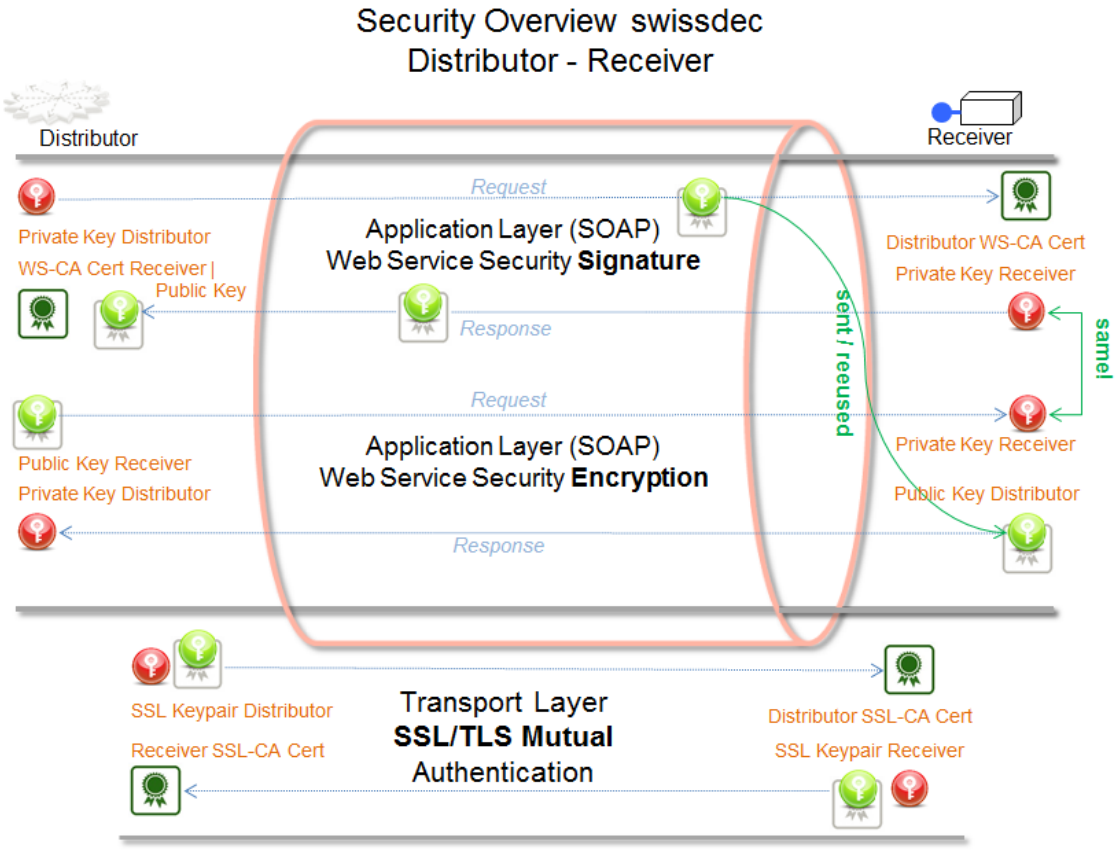


Abbildung 3.1. Security Overview

	Privater Schlüssel (Private Key)
	Zertifikat mit öffentlichem Schlüssel (Public Key). Im Folgenden werden "öffentlicher Schlüssel" (public Key) gleichgesetzt mit dem Zertifikat, welches den öffentlichen Schlüssel beinhaltet.
	CA Zertifikat

Tabelle 3.2. Legende zu Abbildung Security Overview

3.5.3.1. Webservice Security

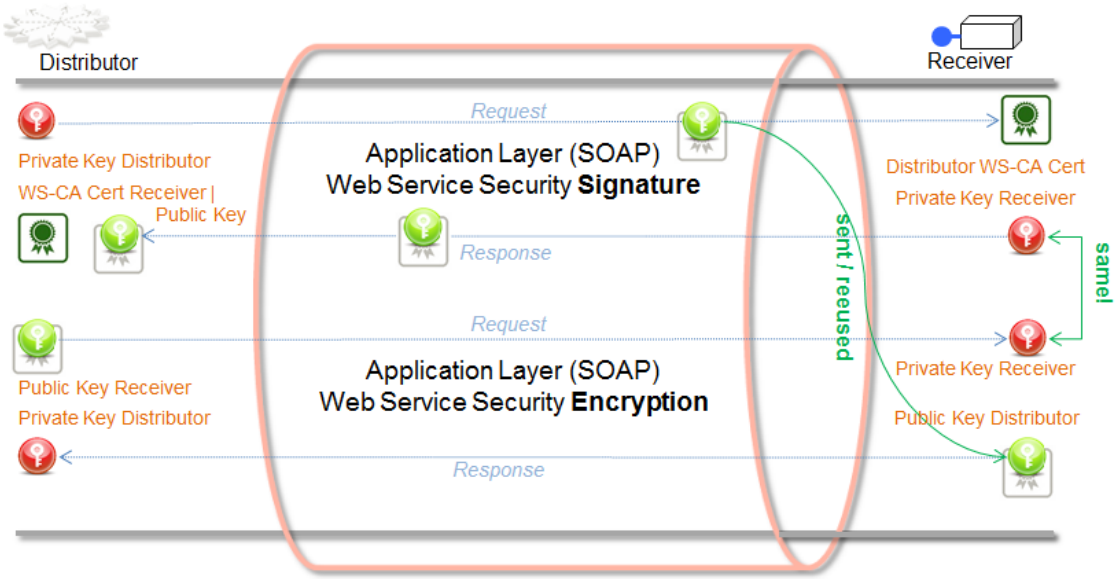


Abbildung 3.2. Webservice Security Overview

3.5.3.1.1. Signatur

Request	Der Klient (swissdec Distributor) signiert den Request mit seinem privaten Schlüssel (Private Key) und schickt seinen öffentlichen Schlüssel (Public Key) mit dem Request mit. Auf Serverseite (swissdec-Adapter) wird der mitgeschickte öffentliche Schlüssel gegen swissdec CA-Zertifikat geprüft und die Signatur verifiziert. Serverseitig wird also nur das swissdec CA-Zertifikat benötigt, um die Signatur zu prüfen.
Response	Der swissdec-Adapter signiert die Antwort mit seinem privaten Schlüssel. Dem swissdec Distributor muss dessen öffentlichen Schlüssel, beziehungsweise das CA-Zertifikat bekannt sein.

Tabelle 3.3. Webservice Security Signature

Für den Webservice Security Signatur Prozess wird auf Serverseite also das CA-Zertifikat des Klienten sowie das eigene Schlüsselpaar (privater und öffentlicher Schlüssel) benötig.

3.5.3.1.2. Verschlüsselung

Request	Der Klient (swissdec Distributor) verschlüsselt den Request mit dem öffentlichen Schlüssel (Public Key) des Empfängers. Auf Serverseite (swissdec-Adapter) wird der Request mit dem eigenen privaten Schüssel dechiffriert.
Response	Der swissdec-Adapter verschlüsselt die Antwort mit dem öffentlichen Schlüssel des swissdec Distributors, der im Request für die Webservice Security Signatur mitgeschickt wurde. Damit wird sichergestellt, dass nur der Inhaber des dazugehörigen privaten Schlüssels die Antwort dechiffrieren kann.

Tabelle 3.4. Webservice Security Encryption

Für den Webservice Security Encryption Prozess wird auf Serverseite nur der private Schlüssel des Endempfängers benötigt, der der gleiche ist wie für die Webservice Security Signatur.

3.5.3.1.3. Konfiguration

Aus der Abbildung 3.2 geht hervor, dass serverseitig (swissdec-Adapter) für die Webservice Security zwei Dinge benötig werden:

- swissdec CA-Zertifikat
- Schlüsselpaar (privater und öffentlicher Schlüssel) des Endempfängers

Da das CA-Zertifikat nur bekannt sein muss, wird es nicht explizit konfiguriert. Es reicht, wenn es im konfigurierten *Keystore* vorhanden ist. Der private Schlüssel des Endempfängers muss jedoch explizit im *Keystore* adressiert werden. Die folgenden Konfigurationsoptionen (swissdecAdapter.properties) steuern das Verhalten des swissdec-Adapters für die Webservice Security.

Option	**Beschreibung**
org.apache.ws.security.crypto.merlin.file	Pfad zum *Keystore* mit den Zertifikaten für die Webservice-Security.
wss4j.signatureUser	Zertifikats- Alias im *Keystore* , mit welchem Key die Antworten signiert werden (Private Key).
wss4j.privateKeyPassword	Passwort zum privaten Schlüssel (wss4j.signatureUser).
org.apache.ws.security.crypto.merlin.keystore.password	Passwort zum *Keystore* .

Tabelle 3.5. swissdecAdapter Webservice Security Konfiguration

Anmerkung

Die produktiven Zertifikate können via *sM-Client* Support (http://jira.ctp-consulting.com) bestellt werden.

Warnung

Sobald die Webservice Security Zertifikate für die Produktion installiert sind, kann sowohl von den Refapps wie auch vom TestTool keine Übermittlungen mehr akzeptiert werden. Dies weil die Signaturen der Meldungen nicht mehr verifiziert werden können, da der produktive Distributor Zertifikate einer anderen CA (Certificate Authority) verwendet.

3.5.3.2. Transport Layer (SSL/TLS)

Der Client baut eine Verbindung zum Server auf. Für gewöhnlich authentifiziert sich zuerst der Server gegenüber dem Client mit einem Zertifikat. Dann schickt entweder der Client dem Server eine mit dem öffentlichen Schlüssel des Servers verschlüsselte geheime Zufallszahl, oder die beiden Parteien berechnen mit dem Diffie-Hellman-Schlüsselaustausch ein gemeinsames Geheimnis. Aus dem Geheimnis wird dann ein kryptographischer Schlüssel abgeleitet. Dieser Schlüssel wird in der Folge benutzt um alle Nachrichten der Verbindung mit einem symmetrischen Verschlüsselungsverfahren zu verschlüsseln und zum Schutz von Nachrichtenintegrität und -authentizität durch einen Message Authentication Code abzusichern.

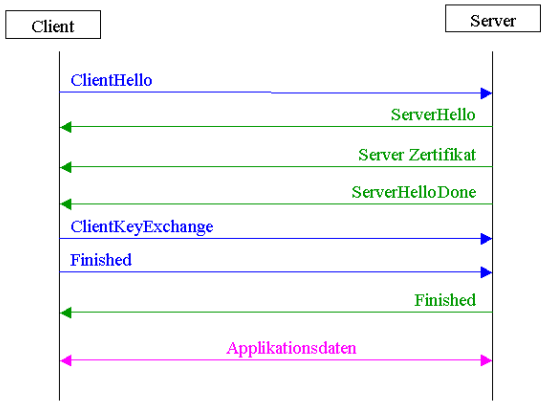


Abbildung 3.3. SSL Handshake

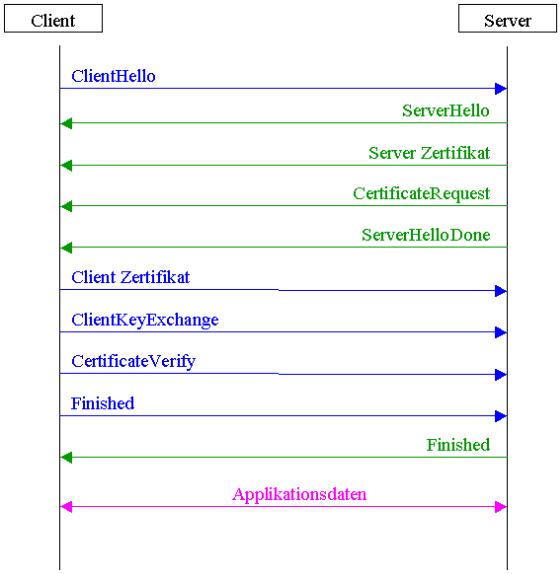


Abbildung 3.4. SSL Handshake mit Mutual Authentication

Bei der Mutual Authentication (gegenseitige Authentifizierung) verlangt der Server vom Klienten ebenfalls, dass sich dieser mit einem Zertifikat authentifiziert. Dies ist im swissdec-Standard vorgeschrieben um sicherzustellen, dass alle Parteien einander gegenseitig vertrauen.

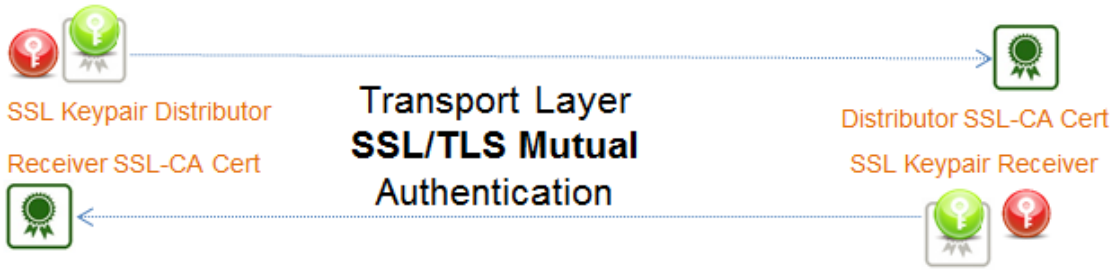


Abbildung 3.5. Transport Layer Security Overview

In der vereinfachten Ansicht wird ersichtlich, dass auf beiden Seiten (Klient/swissdec Distributor, Server/swissdec-Adapter) sowohl ein Schlüssel-Paar (privater Schlüssel mit dazugehörigem öffentlichen Schlüssel) um sich auszuweisen sowie das CA-Zertifikat des Gegenübers zum verifizieren benötigt wird.

Die SSL/TLS Konfiguration für den swissdec-Adapter befindet sich in der Datei "conf/server.xml". Falls die Terminierung auf dem Adapter gemacht werden soll, befindet sich eine beispielhafte Konfiguration in der selben Datei.

```
<Connector port="8443" protocol="HTTP/1.1"
        connectionTimeout="20000"
        scheme="https" secure="true" SSLEnabled="true"
        keystoreFile="[path]/server.jks" keystorePass="keystore pass"
        truststoreFile="[path]/truststore.jks" truststorePass="truststore pass"
        clientAuth="true" sslProtocol="TLS"/>
```

Beispiel 3.2. Tomcat Connector Konfiguration für SSL-Terminierung mit Mutual Authentication

Option	**Beschreibung**
port	Der TCP-Port, auf welchem der Server horcht.
keystoreFile	Der Pfad zur *Keystore* -Datei. Relativ zum <SWISSDECADAPTER_HOME> Verzeichnis.
keystorePass	Passwort zur *Keystore* -Datei.
truststoreFile	Der Pfad zur Truststore-Datei. Relativ zum <SWISSDECADAPTER_HOME> Verzeichnis.
truststorePass	Passwort zur Truststore-Datei.
clientAuth	Der Wert "true" verlangt eine Klientenauthentifizierung (Mutual Authentication).

Tabelle 3.6. swissdecAdapter Transport Layer Konfiguration

Anmerkung

Keystore (mit dem Schlüsselpaar) und Truststore (mit dem CA-Zertifikat des swissdec Distributors) werden der Einfachheit halber in einer Datei geliefert. Damit sind beide Angaben identisch!

Die produktiven Zertifikate können via sM-Client Support (http://jira.ctp-consulting.com) bestellt werden. Dazu muss der Hostname (DNS-Eintrag) bekannt gegeben werden, mit dem die swissdec-Adapter Installation aus dem Internet erreichbar ist.

Warnung

Der Tomcat (Servlet-Container des swissdec-Adapters) hat die Limitierung, dass das Keystore-Passwort identisch mit dem Passwort für den darin enthaltenen privaten Schlüssel sein muss.

3.6. Integrationsapplikation

3.6.1. Tomcat Konfiguration

Der mitgelieferte Servlet-Container "Tomcat" kann bis auf eine Ausnahme mit der mitgelieferten Konfiguration gestartet werden. Bitte prüfen Sie, ob der Standardport 9090 auf dem Computer noch nicht belegt ist. Falls dies der Fall sein sollte, bitte im <SWISSDECADAPTER_HOME>/conf/server.xml im Abschnitt "Connector" den Wert für "port" auf einen freien Port ändern. Bedenken Sie, dass wenn hier der Port geändert wird, in der Receiver-Applikation der Wert für "salarydeclaration.service.port" im <SWISSDECADAPTER_HOME>/conf/swissdecAdapter.properties angepasst werden muss.

```
<Connector port="9090" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443"/>
```

Beispiel 3.3. Standardconnector- Konfiguration Tomcat

3.6.2. Integrationsapplikation Konfiguration

Sobald das ZIP-Archiv entpackt ist, müssen Sie die Konfiguration anpassen. Im Kapitel 6, *Konfigurations-Referenz* finden Sie eine komplette Liste aller möglichen Einstellungen. Der folgende Abschnitt beschreibt die Pflichtkonfiguration, die in der Datei <SWISSDECADAPTER_HOME>/conf/swissdecAdapter.properties vorgenommen werden muss.

Option	**Beschreibung**
released.filesystem.parent.xml	Pfad zum Meldungseingang (PATH_ELM_IN). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren.
failed.filesystem.parent.xml	Pfad zum "failed to receive" (PATH_ELM_FAILED_TO_RX). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren.
result.filesystem.parent.xml	Pfad zum Meldungsausgang (PATH_ELM_OUT). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren.
sent.filesystem.parent.xml	Pfad zum "sent" (PATH_ELM_SENT). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren.
undeliverable.filesystem.parent.xml	Pfad zum "failed to send" (PATH_ELM_FAILED_TO_TX). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren.
institution.tas.id	Technische Angabe, für welche Institution QST- Meldungen empfangen werden (zum Beispiel „BE“, „ZH“, „GE“, ...). MUSS richtig konfiguriert sein, da sonst Meldungen abgelehnt werden
institution.tax.id	Technische Angabe, für welche Institution Lohnausweise empfangen werden (zum Beispiel „BE“, „ZH“, „GE“, ...). MUSS richtig konfiguriert sein, da sonst Meldungen abgelehnt werden

Tabelle 3.7. swissdec-Adapter Integration Grundkonfiguration

3.7. Logging-Konfiguration

Die Konfiguration für das Logging befindet sich in der Datei <SWISSDECADAPTER_HOME>/lib/log4j.xml. Die im Paket enthaltene Konfiguration schreibt 3 Log-Dateien.

Datei	**Beschreibung**
catalina.out	Log des Servlet-Containers (Tomcat)
access.log	Spezielle Ereignisse, zum Beispiel eingehende Meldungen, Wartungsfenster, etc.
swissdecAdapter.log	Log des swissdecAdapters

Tabelle 3.8. Log-Dateien

Standardmässig werden diese 3 Dateien ins Verzeichnis <SWISSDECADAPTER_HOME>/logs/ geschrieben und täglich "gerollt".

Mehr Informationen entnehmen Sie bitte folgender Webseite: http://wiki.apache.org/logging-log4j/Log4jXmlFormat

3.8. Wartungsfenster

Da die Meldungen auf der swissdec Platform synchron (Echtzeit) übertragen werden sehen die Kunden bei der Übermittlung, welche Endempfänger nicht erreichbar waren. Es besteht die Möglichkeit, geplante Betriebsunterbrüche mittels Wartungsfenster anzukündigen. Diese sollten nach Möglichkeit zwischen 20:00 und 6:00 Wochentags oder an Wochenenden geplant werden. Damit wird den Kunden eine Mitteilung anstelle eines Fehlers geliefert. Um ein Wartungsfenster zu registrieren, gehen Sie wie folgt vor:

1. Erstellen Sie ein "PlannedMaintenance" XML. Im Verzeichnis <SWISSDECADAPTER_HOME>/conf finden Sie ein Template (PlannedMaintenance.xml).

2. Speichern Sie es bei der swissdec-Adapter Integrationsapplikation ins Verzeichnis "maintenance.filesystem.parent.xml". Der Name spielt dabei keine Rolle, die Dateiendung muss ".xml" sein.

3. Das Wartungsfenster wird beim nächsten Ping des Distributors mitgeteilt. Die Erreichbarkeit des End-Empfängers wird vom Distributor aus zyklisch (alle 30 Minuten) geprüft. Dazu wird die Webservice-Operation "PingConsumer" des End-Empfänger aufgerufen, der seinerseits die Erreichbarkeit mit der Antwort bestätigt. In diesem Prozess können in der Antwort des End-Empfängers Wartungsfenster registriert werden.

4. Sobald das Wartungsfenster vorbei ist, wird das "PlannedMaintenance" XML automatisch gelöscht.

Anmerkung

- Der Dateiname spielt keine Rolle, die Dateiendung muss zwingend ".xml" sein
- Die Zeiten "Start" und "End" im "PlannedMaintenance" XML sind vom Typ xs:dateTime. Das heisst, dass auch die Zeitzone sowie Sommer-/Winterzeit darin enthalten sein müssen.

- 2013-08-31T13:40:47.0Z: das Z steht für UTC
- 2014-01-31T13:40:47.0+01:00: UTC +1 Stunde, entspricht Mitteleuropäischer Zeit (Winterzeit)
- 2013-08-31T13:40:47.0+02:00: UTC +2 Stunden, entspricht Mitteleuropäischer Sommerzeit

- Die "Messages" werden bis zur Lohnbuchhaltung kommuniziert. Bitte wählen Sie vernünftige Texte in den 3 Sprachen (Deutsch, Französisch, Italiensich).
- Es können mehrere "PlannedMaintenance" XML ins Verzeichnis "maintenance.filesystem.parent.xml" gelegt werden. Der swissdec-Adapter wird immer das nächste dem Distributor melden und bei Ablauf automatisch löschen bzw. das nächste melden.

Beispiel eines "PlannedMaintenance" XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<PlannedMaintenance xmlns="http://www.itserve.ch/step/2.0/maintenance">
   <Start>2013-08-31T13:40:47.0+01:00</Start>
   <End>2013-09-01T13:40:47.0+01:00</End>
   <Messages>
      <Message language="de">Beschreibung Deutsch</Message>
      <Message language="fr">Beschreibung Französisch</Message>
      <Message language="it">Beschreibung Italienisch</Message>
   </Messages>
</PlannedMaintenance>
```

3.9. Schlussbemerkung

Bitte stellen Sie sicher, dass die beiden Applikationen des swissdec-Adapters nach einem Systemneustart automatisch gestartet werden. Mehr dazu im Kapitel 5, *Starten des Adapters*.

4. Installation testen

Um die Installation prüfen zu können, stellt die swissdec ein Test Tool zur Verfügung. Es handelt sich dabei um einen einfachen Webservice- Klienten, mit dem Quellensteuerabrechnungen und Lohnausweise im swissdec V4 Format versendet werden können.

4.1. Installation

Das TestTool wird als ZIP-Datei geliefert und muss nur entpackt werden. Zum Ausführen wird eine Java-Installation benötigt.

Warnung

Da das TestTool swissdec-konform verschlüsselt, muss für die Java-Installation JCE installiert sein. Mehr dazu im Abschnitt 3.4, „Java JCE installieren“.

4.2. Konfiguration

Bevor Sie testen können, müssen die Dateien

- Config-TAS.xml
- Config-TAX.xml

angepasst werden. Es handelt sich dabei um Properties-Files im XML-Format.

Option	Beschreibung
endpointURL	Die zu testende Adresse des swissdec-Adapters.
wssEncryptionCert	Das Zertifikat wird zum Verschlüsseln der Meldungen benutzt (Public Key des swissdec-Adapters).
sigrootcert	Das Zertifikat wird zum Verifizieren der Signatur der Antwort benutzt (Public Key des swissdec-Adapters).
sslcert	Zertifikat mit dem Public Key zum SSL-Private Key
sslkey	SSL-Private Key für die Klientenauthentifizierung
sslpartnercacert	CA- Zertifikat zum Prüfen des SSL/TLS Serverzertifikats.

Tabelle 4.1. TestTool Konfiguration

Mehr Informationen zur Security entnehmen Sie bitte dem Abschnitt 3.5.3, „Security Konfiguration“.

Anmerkung

Die mitgelieferte Konfiguration passt zur Default-Konfiguration des swissdec-Adapters.

Die Option "sslpartnercacert" wird nur bei aktiven HTTPS benötigt.

Die Optionen "sslcert" und "sslkey" werden nur bei aktivem Mutual Authentication benötigt.

4.3. Ausführen der Tests

Um einen Test auszuführen, wird das dazugehörige Batch- beziehungsweise Shellskript ausgeführt.

Anmerkung

Die einfachsten Tests sind:

- INTG_001_Smoke_TAS
- INTG_002_Smoke_TAX

5. Starten des Adapters

5.1. Manueller Start

Beide swissdec-Adapter Applikationen (Receiver und Integration) können mit folgenden Kommandos in der Konsole gestartet/gestoppt werden:

Windows

```
<SWISSDECADAPTER_HOME>/bin/startup.bat
```

```
<SWISSDECADAPTER_HOME>/bin/shutdown.bat
```

Unix

```
<SWISSDECADAPTER_HOME>/bin/startup.sh
```

```
<SWISSDECADAPTER_HOME>/bin/shutdown.sh
```

Anmerkung

Bitte nehmen Sie vor dem ersten Start wie im Kaptiel Adapter Konfiguration beschrieben die Konfiguration vor.

5.2. Starten als Windows-Dienst

Um den Tomcat als Windows-Dienst zu installieren, befolgen Sie die folgenden Schritte. Bitte beachten Sie, dass folgende Schritte für beide Applikationen (Receiver und Integration) durchzuführen sind.

1. Stoppen Sie den swissdec-Adapter, falls er läuft.
2. Öffnen Sie die Kommandozeile und navigieren Sie zum Verzeichnis <SWISSDECADAPTER_HOME>/bin/

   Sie sollten nun unter Anderen folgende drei Dateien sehen:

   - service.bat
   - tomcat6.exe
   - tomcat6w.exe
3. Führen Sie die folgenden Befehle aus:

```
service.bat install
tomcat6w.exe //ES//
```

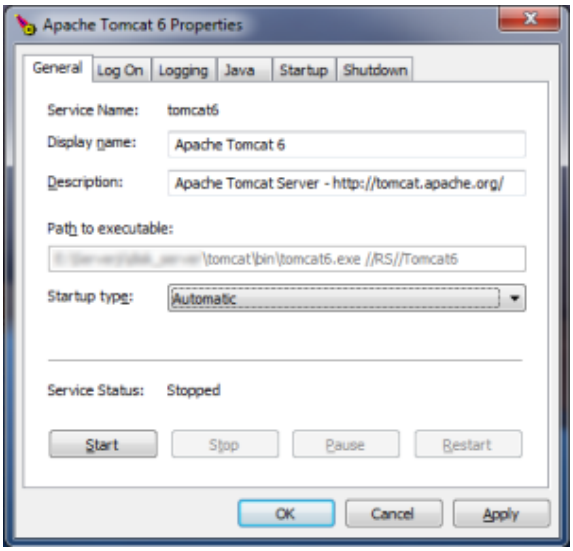


Abbildung 5.1. tomcat6w.exe, Tomcat Service Properties

Stellen Sie sicher, dass der Wert für „Startup“ auf automatic gesetzt ist.

4. Auf dem Reiter "Java" erhöhen Sie die Werte “Initial memory pool” auf 256 MB und “Maximum memory pool” auf mindestens 1024 MB.

5. Ergänzen Sie die "Java Options" wie folgt:

```
-XX:MaxPermSize=256m
-Dproperties.url=file:///C:/.../swissdecAdapter/conf/swissdecAdapter.properties
-Dderby.system.home=C:\...\swissdecAdapter\derbydb
```

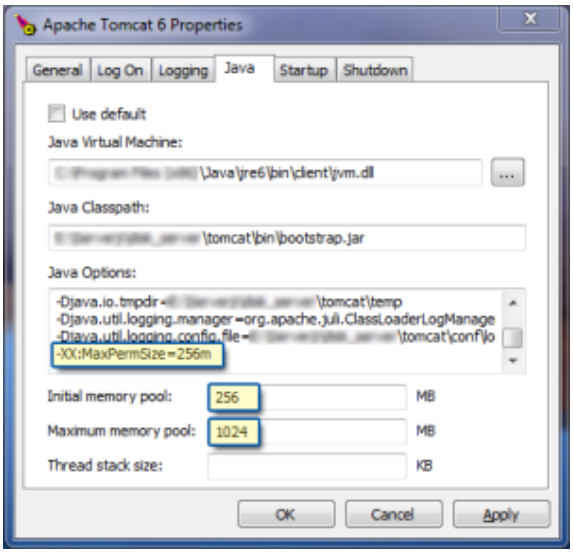


Abbildung 5.2. Tomcat Service Java Properties

Anmerkung

Beachten Sie, dass die Optionen "properties.url" und "derby.system.home" für Ihre Umgebung angepasst werden müssen.

- properties.url: zeigt auf <SWISSDECADAPTER_HOME>/conf/swissdecAdapter.properties
- derby.system.home: zeigt auf <SWISSDECADAPTER_HOME>/derbydb

6. Der Dienst "Tomcat6" sollte nun installiert sein. Prüfen Sie die Einstellungen mittels der Windows Dienst Verwaltung.

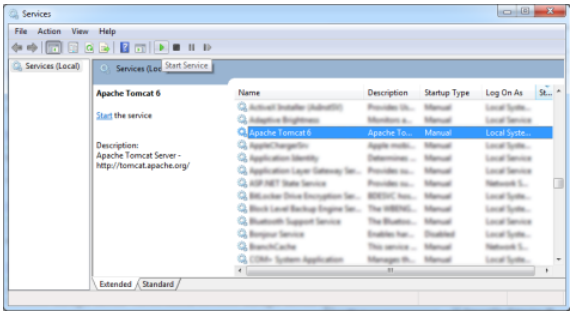


Abbildung 5.3. Windows Dienst Verwaltung

5.3. Unix Init-Skript

Bitte beachten Sie, dass beide Applikationen (Receiver und Integration) gestartet werden. Im Verzeichnis <SWISSDECADAPTER_HOME>/bin/ befindet sich ein init.d Skript (tomcat6.sh) für den mitgelieferten Tomcat. Folgende Variabeln müssen Sie im Skript anpassen:

```
JAVA_HOME=/usr/lib/jvm/java-6-openjdk
TOMCAT_USER=tomcat6
CATALINA_HOME=/opt/swissdecAdapter
CATALINA_PID=$CATALINA_HOME/bin/tomcat6.pid
```

Es wird empfohlen, für den swissdec-Adapter einen Benutzer anzulegen. Bitte beachten Sie, dass dieser in den Verzeichnissen, die im <SWISSDECADAPTER_HOME>/conf/swissdecAdapter.properties konfiguriert sind, Lese- und Schreibrechte benötigt.

```
#!/bin/sh -e
# description: Tomcat6 service
# processname: java

# Get LSB functions
. /lib/lsb/init-functions

export JAVA_HOME=/usr/lib/jvm/java-6-openjdk
export TOMCAT_USER=tomcat6
export CATALINA_HOME=/opt/swissdecAdapter
export CATALINA_PID=$CATALINA_HOME/bin/tomcat6.pid

[ -d "$CATALINA_HOME" ] || { echo "Tomcat requires $CATALINA_HOME."; exit 1; }

case $1 in

   start|stop|run)
   if su $TOMCAT_USER bash -c "$CATALINA_HOME/bin/catalina.sh $1"; then
      log_success_msg "Tomcat $1 successful"
      [ $1 == "stop" ] &amp;&amp; rm -f $CATALINA_PID
   else
      log_failure_msg "Error in Tomcat $1: $?"
   fi
   ;;

   restart)
   $0 start
   $0 stop
   ;;

   status)
   if [ -f "$CATALINA_PID" ]; then
      read kpid &lt; "$CATALINA_PID"
      if ps --pid $kpid 2>&amp;1 1>/dev/null; then
         echo "$0 is already running at ${kpid}"
      else
         echo "$CATALINA_PID found, but $kpid is not running"
      fi
      unset kpid
   else
      echo "$0 is stopped"
   fi
   ;;

esac
exit 0
```

6. Konfigurations-Referenz

6.1. Tomcat Konfiguration

Die komplette Konfigurationsdokumentation entnehmen Sie bitte folgender Webseite: http://tomcat.apache.org/tomcat-6.0-doc/config/index.html.

6.2. Receiver Konfiguration

Option	Beschreibung
userAgent.institutionName	Deskriptiver Name der Institution (zum Beispiel „KSTV Bern“)
institution.tas.id	Technische Angabe, für welche Institution QST- Meldungen empfangen werden (zum Beispiel „BE“, „ZH“, „GE“, ...). MUSS richtig konfiguriert sein, da sonst Meldungen abgelehnt werden
institution.tax.id	Technische Angabe, für welche Institution Lohnausweise empfangen werden (zum Beispiel „BE“, „ZH“, „GE“, ...). MUSS richtig konfiguriert sein, da sonst Meldungen abgelehnt werden
tax.accept.ex	Definiert, ob Lohnausweise für im Ausland wohnhafte Personen akzeptiert werden (Default = false). Bitte sprechen Sie mit den Ihrer Fachabteilung, ob diese Option gewünscht ist. Falls ja nehmen Sie bitte Kontakt auf (JIRA oder E-Mail), da dies auf dem Distributor konfiguriert werden muss.
salarydeclaration.service.protocol	Das Prokotoll, auf dem die swissdec-Adapter Integration läuft.
salarydeclaration.service.host	Der Host, auf dem die swissdec-Adapter Integration läuft.
salarydeclaration.service.port	Der Port, auf dem die swissdec-Adapter Integration läuft.
salarydeclaration.service.context	Der Context, auf dem die swissdec-Adapter Integration läuft.
salarydeclaration.service.name	Der Name, unter dem die swissdec-Adapter Integraiton läuft.
monitoring.enabled	Schalter, ob das eingebaute Monitoring aktiv ist.
monitoring.user	Benutzername zur Authentifizierung am eingebauten Monitoring.
monitoring.pass	Passwort zur Authentifizierung am eingebauten Monitoring.

Option	Beschreibung
org.apache.ws.security.crypto.merlin.file	Pfad zum *Keystore* mit den Zertifikaten für die Webservice-Security.
wss4j.signatureUser	Zertifikatsalias im *Keystore* , mit dem Key die Antworten signiert werden.
wss4j.privateKeyPassword	Passwort zum privaten Schlüssel (wss4j.signatureUser).
org.apache.ws.security.crypto.merlin.keystore.password	Passwort zum *Keystore*.
noSecurity	Zum Ausschalten der Web Service Security. Zum Beispiel beim Einsatz einer Application Firewall.

Tabelle 6.1. swissdec-Adapter Receiver Konfiguration

6.3. Integrationsapplikation Konfiguration

Option	Beschreibung
released.filesystem.parent.xml	Pfad zum Meldungseingang (PATH_ELM_IN). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren.
failed.filesystem.parent.xml	Pfad zum "failed to receive" (PATH_ELM_FAILED_TO_RX). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren.
result.filesystem.parent.xml	Pfad zum Meldungsausgang (PATH_ELM_OUT). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren.
sent.filesystem.parent.xml	Pfad zum "sent" (PATH_ELM_SENT). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren.
undeliverable.filesystem.parent.xml	Pfad zum "failed to send" (PATH_ELM_FAILED_TO_TX). Bitte absolute Pfadangabe. Muss mit dem sM-Client korrespondieren.
maintenance.filesystem.parent.xml	Pfad zum Ordner für Wartungsfensterdefinitionen. Bitte absolute Pfadangabe.
institution.tas.id	Technische Angabe, für welche Institution QST- Meldungen empfangen werden (zum Beispiel „BE“, „ZH“, „GE“, ...). MUSS richtig konfiguriert sein, da sonst Meldungen abgelehnt werden
institution.tax.id	Technische Angabe, für welche Institution Lohnausweise empfangen werden (zum Beispiel „BE“, „ZH“, „GE“, ...). MUSS richtig

Option	Beschreibung
	konfiguriert sein, da sonst Meldungen abgelehnt werden
monitoring.enabled	Schalter, ob das eingebaute Monitoring aktiv ist.
monitoring.user	Benutzername zur Authentifizierung am eingebauten Monitoring.
monitoring.pass	Passwort zur Authentifizierung am eingebauten Monitoring.
commune.splitting	Aktiviere Gemeindesplitting für QST-Abrechnungen. Bei aktivem Gemeindesplitting wird await.result implizit „false“, da Gemeinden keine Antworten geben können.
la.splitting	Aktiviere Lohnausweissplitting.
write.original.xml	Soll bei aktiven Splitting die vollständige original Datei auch ins ZIP geschrieben werden.
await.result	Sind QST-Abrechnungsresultate zu erwarten.
processing.default.hours	Nach wievielen Stunden kann ein allfälliges QST-Abrechnungsresultat erwartet werden.
housekeeping.days	Vorhaltezeit für Meldungen. Nach Ablauf werden die Meldungen in der internen Datenbank gelöscht. Danach kann auch kein QST-Abrechnungsresultat mehr angefragt werden.

Tabelle 6.2. swissdec-Adapter Integration Konfiguration

7. Monitoring

Der swissdec-Adapter kann wie jeder andere Standardprozess oder Service überwacht werden. Falls der swissdec-Adapter unter Windows als Service registriert wurde, kann die „Windows Computer Management Console“ benutzt werden, um den Status des Dienstes zu prüfen/ ändern.

7.1. Eingebautes Monitoring

Beide swissdec-Adapter Applikationen stellen Statusinformationen via Restful Services zur Verfügung.

```
http://<INTEGRATION_HOST>:<INTEGRATION_PORT>/integration/ws/monitoring
```

für die Integrationsapplikation und

```
http://<RECEIVER_HOST>:<RECEIVER_PORT>/webservice/ws/monitoring
```

für den Receiver. Um das Monitoring zu aktivieren, müssen Sie folgende Parameter pro Applikaiton in der Datei <SWISSDECADAPTER_HOME>/conf/swissdecAdapter.properties anpassen:

```
monitoring.enabled = true
monitoring.user = admin
monitoring.pass = admin
```

Warnung

Bitte stellen Sie sicher, dass das Monitoring der Receiver Applikation (http://<HOST>:<PORT>/webservice/ws/monitoring) nicht aus dem Internet verfügbar ist.

Beispielausgabe des eingebauten Monitorings:

```
<services xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:noNamespaceSchemaLocation="serviceresponse.xsd">
  <service name="SystemProperties">
    <state>OK</state>
    <message>
      <SystemProperties>
        <SystemProperty>
          <Key>java.runtime.name</Key>
          <Value>OpenJDK Runtime Environment</Value>
        </SystemProperty>
        ...
      </SystemProperties>
    </message>
    <runtimeInMs>8</runtimeInMs>
  </service>
  <service name="Database">
    <state>OK</state>
    <message>Last Distributor Ping: </message>
    <runtimeInMs>133</runtimeInMs>
  </service>
  <service name="Configuration">
    <state>OK</state>
    <message>
      <Configuration>
        <ConfigProperty>
          <Key>versioninfo</Key>
          <Value>2.0 (Build xxxxx)</Value>
        </ConfigProperty>
        ...
      </Configuration>
    </message>
    <runtimeInMs>1</runtimeInMs>
  </service>
  <service name="Settings">
    <state>OK</state>
    <message>
      <Configuration>
        <ConfigProperty>
          <Key>distributor.lastping</Key>
          <Value></Value>
        </ConfigProperty>
          ...
      </Configuration>
    </message>
    <runtimeInMs>0</runtimeInMs>
  </service>
  <executionTime>0,142"</executionTime>
</services>
```

8. Recovery-Tool

Das Recovery-Tool ist ein Hilfsprogramm um Informationen aus der swissdecAdapter Integration zu visualisieren und gegebenenfalls neu zu exportieren.

8.1. Programmstart

Um das Programm zu starten, muss einmalig die Datei RecoveryTool-2.0_2.zip entpackt werden. Danach wird das Programm mit dem Ausführen der Datei RecoveryTool.bat (Windows) oder RecoveryTool.sh (Linux) gestartet. Das Recovery Tool ist eine Konsolenapplikation, die mit der Tastatur gesteuert wird. Hilfe im Programm erhalten Sie durch den Befehl "h".

8.2. Optionen

Folgende Steuerungsoptionen sind immer verfügbar:

Option	**Beschreibung**
h	Zeigt Ihnen die möglichen Operationen an.
q	Beendet das Recovery Tool.
u	Sie kommen in der Menustruktur ein Element nach oben.

Tabelle 8.1. Recovery Tool, Generelle Optionen

8.2.1. Declarations

In diesem Menupunkt können alle Meldungen, welche in der Datenbank der swissdecAdapter Integration gespeichert sind, gesucht und angezeigt werden.

8.2.1.1. list | l

Listet alle Deklarationen auf. Mit der Eingabe der Zahl in den eckigen Klammern werden die Details der Deklaration/Person angezeigt.

```
swissdecAdapter RecoveryTool V2.0_2

Running with: Sun Microsystems Inc. 1.6.0_21 (/usr/lib/jvm/jdk1.6.0_21/jre)
Running on: Linux 3.11.0-15-generic (amd64)

Type h or help for more information.

sdArT:~$ declarations
sdArT:~/declarations$ list
found 3 actions. type h to show list.
sdArT:~/declarations/list$ help

[1] TAS  INTG_001_Smoke_TAS_20140130_15201391091643
[2] TAX  INTG_002_Smoke_TAX_20140130_15201391091658
[3] TAX  REGR_001_TaxSplittingBug_20140130_15211391091680

[q] quit (Exit RecoveryTool)
[u] up (Go one Up)
sdArT:~/declarations/list$ 2
DECLARATION DETAILS:
Domain:    TAX
DeclarationID:  D0002
RequestID:   123456
ResponseID:   R143e3864462b
Time:    2014-01-30 15:21:03.0
TestCase:   TESTCASE
External state:  ACCEPTED with Warnings
Internal state:  processed
CompanyName:  INTG_002_Smoke_TAX_20140130_15201391091658
Nb. of Persons:  2
Contact:   ContactName
E-Mail:    dko@itserve.ch
Phone:    contactPhone

sdArT:~/declarations/list$
```

8.2.1.2. search | s

Suche nach PkID (interner Schlüssel) oder DeclarationID

```
swissdecAdapter RecoveryTool V2.0_2

Running with: Sun Microsystems Inc. 1.6.0_21 (/usr/lib/jvm/jdk1.6.0_21/jre)
Running on: Linux 3.11.0-15-generic (amd64)

Type h or help for more information.

sdArT:~$ declarations
sdArT:~/declarations$ search
Enter what you are looking for: D0007
found 9 actions. type h to show list.
sdArT:~/declarations/search$ h

[1] TAS  FUNK_001_TAS_RESULT_PROC_20140131_10331391160800
[2] TAS  FUNK_002_TAS_RESULT_SUCC_20140131_10331391160831
[3] TAS  FUNK_003_TAS_SPLITTING_20140131_10341391160847
[5] TAS  FUNK_005_TAS_NORESULT_20140131_10341391160872
[6] TAS  FUNK_007_DUPLICATE_20140131_10341391160895
[7] TAS  FUNK_008_UNDELIVERABLE_20140131_10351391160904
[8] TAS  FUNK_009_WRITE_ORIGINAL_20140131_10351391160919
[9] TAS  INTG_001_Smoke_TAS_20140131_10351391160937
[47] TAS  INTG_001_Smoke_TAS_20140203_15031391436191

[q] quit (Exit RecoveryTool)
[u] up (Go one Up)
sdArT:~/declarations/search$ 5
DECLARATION DETAILS:
Domain:    TAS
DeclarationID:  D0007
RequestID:   reqId
ResponseID:   R143e7a6617c14a
Time:    2014-01-31 10:34:37.0
TestCase:   PRODUCTIVE
External state:  ACCEPTED with Warnings
Internal state:  contributions delivered
CompanyName:  FUNK_005_TAS_NORESULT_20140131_10341391160872
Nb. of Persons:  7
Contact:   Marcel René Müller
E-Mail:    ComplexEnd2End@xxxxx.ch
Phone:    031 345 23 87

sdArT:~/declarations/search$
```

8.2.2. Recovery

Im Recovery-Modus können durch Fehler betroffene Deklarationen angezeigt und exportiert werden.

8.2.2.1. newTaxRemarks | n

Zeigt alle Lohnausweise, die die neuen Standard-Remarks gesetzt haben. Dieser Befehl kann eine Weile dauern.

```
swissdecAdapter RecoveryTool V2.0_2

Running with: Sun Microsystems Inc. 1.6.0_21 (/usr/lib/jvm/jdk1.6.0_21/jre)
Running on: Linux 3.11.0-15-generic (amd64)

Type h or help for more information.

sdArT:~$ recovery
sdArT:~/recovery$ newTaxRemarks
found 4 actions. type h to show list.
sdArT:~/recovery/newTaxRemarks$ help

[0] INTG_002_Smoke_TAX_20140130_15201391091658      123.4500.0178.12  ...
[1] INTG_002_Smoke_TAX_20140130_15201391091658      756.9876.5432.10  ...
[2] REGR_001_TaxSplittingBug_20140130_15211391091680  123.4500.0178.12     ...
[3] REGR_001_TaxSplittingBug_20140130_15211391091680  123.4500.0178.12     ...

[q] quit (Exit RecoveryTool)
[u] up (Go one Up)
sdArT:~/recovery/newTaxRemarks$
```

8.2.2.2. splitting | s

Zeigt und exportiert (Option "reexport") alle vom Lohnausweis-Splitting Bug betroffenen Deklarationen.

```
swissdecAdapter RecoveryTool V2.0_2

Running with: Sun Microsystems Inc. 1.6.0_21 (/usr/lib/jvm/jdk1.6.0_21/jre)
Running on: Linux 3.11.0-15-generic (amd64)

Type h or help for more information.

sdArT:~$ recovery
sdArT:~/recovery$ splitting
found 2 actions. type h to show list.
sdArT:~/recovery/splitting$ help

[3] TAX  REGR_001_TaxSplittingBug_20140130_15211391091680
[reexport] reexport 1 found declarations

[q] quit (Exit RecoveryTool)
[u] up (Go one Up)
sdArT:~/recovery/splitting$ reexport
reexport 1 found declarations

THE FILES WILL BE SPLITTED!

do you want to reexport into directory /home/.../.../reexport? this directory is created if not exists! [y|N] y
Please enter your Canton abbreviation [AI | BE | VD | ...] BE
sdArT:~/recovery/splitting$
```

8.2.3. Statistik

Zeigt eine kurze Übersicht über erhaltene Deklarationen.

```
swissdecAdapter RecoveryTool V2.0_2

Running with: Sun Microsystems Inc. 1.6.0_21 (/usr/lib/jvm/jdk1.6.0_21/jre)
Running on: Linux 3.11.0-15-generic (amd64)

Type h or help for more information.

sdArT:~$ statistics
Total TAX: 2
Total TAX productive: 0
Total TAS: 1
Total TAS productive: 1
sdArT:~$
```

9. Probleme und Lösungen

Meldung

```
Caused by: org.apache.ws.security.components.crypto.CredentialException: Failed to load credentials.
   Inner Exception: [Keystore was tampered with, or password was incorrect]
```

Falsches *Keystore* Passwort in swissdec-Adapter Receiver swissdecAdapter.properties (Property 'org.apache.ws.security.crypto.merlin.keystore.password').

Meldung

```
Caused by: java.io.FileNotFoundException: abTst_server.jks (No such file or directory)
```

Der *Keystore* referenziert in swissdec-Adapter Receiver swissdecAdapter.properties existiert nicht (Property 'org.apache.ws.security.crypto.merlin.file')

Meldung

```
Caused by: java.security.UnrecoverableKeyException: Cannot recover key
```

Falsches Private Key Passwort in swissdec-Adapter Receiver swissdecAdapter.properties (Property 'wss4j.privateKeyPassword')

Meldung

```
Caused by: java.security.InvalidKeyException: Illegal key size
```

Die Java Crypto Extension sind nicht installiert.

Meldung

```
Caused by: javax.xml.ws.soap.SOAPFaultException: Marshalling Error: Connection refused
 at org.apache.cxf.jaxws.JaxWsClientProxy.invoke(JaxWsClientProxy.java:156)
 at $Proxy205.saveSalaryDeclaration(Unknown Source)
```

Die Receiver Applikation hat keine Verbindung zur Integration. Bitte kontrollieren Sie folgende Konfiguration im Receiver swissdecAdapter.properties:

```
salarydeclaration.service.protocol = http://
salarydeclaration.service.host = localhost
salarydeclaration.service.port = 9090
salarydeclaration.service.context = integration/ws
salarydeclaration.service.name = SalaryDeclarationServiceFacade
salarydeclaration.service.interface =
ch.itserve.lohnstandard.receiver.shared.bi.ISalaryDeclarationFacade
```

Stellen Sie sicher, dass die Konfiguration an Ihre Infrastruktur angepasst wurde und dass eine HTTP Verbindug von der Receiver Applikation auf die konfigurierte URL geöffnet werden kann.

A. Anhang

A.1. Referenzierte Dokumente

[ENDRECREQ] *Richtlinien für Lohndatenübermittlung*. Endreceiver Requirements. swissdec. Version 4 (20130514).

[DETAILSPEZ] *swissdec-Adapter*. Detailspezifikation. itServe AG. V1.0 (20130430).

A.2. Glossar

DMZ	ein Computernetz mit sicherheitstechnisch kontrollierten Zugriffsmöglichkeiten auf die daran angeschlossenen Server
Keystore	Ein Keystore ist eine Datenbank von Schlüsseln und Zertifikaten in einer Datei.
NTP	Standard zur Synchronisierung von Uhren in Computersystemen.
sM-Client	Der sM-Client bietet Funktionalität zum Versenden und Empfangen von Meldungen in diversen Formaten
STEP	Produkt der Firma itServe AG, *swissdec* Empfänger
swissdec	Qualitätslabel für Lohnbuchhaltungssysteme. XML-Standards für die Einheitliche Lohn-Meldung (ELM).

B. Beispiele

B.1. Konfiguration

```
                #
# Copyright 1996-2013 itServe AG. All rights reserved.
#
# This software is the proprietary information of itServe AG
# Bern Switzerland. Use is subject to license terms.
#

# RECEIVER INDENTIFICATION
userAgent.institutionName=KSTV Bern
institution.tas.id=BE
institution.tax.id=BE
tax.accept.ex=false

# INTERNAL COMMUNICATION
salarydeclaration.service.protocol=http://
salarydeclaration.service.host=localhost
salarydeclaration.service.port=9090
salarydeclaration.service.context=integration/ws
salarydeclaration.service.name=SalaryDeclarationServiceFacade

# MONITORING
monitoring.enabled=false
monitoring.user=admin
monitoring.pass=admin

# WS-SECURITY CONFIGURATION
wss4j.signatureUser=abtst-server
wss4j.privateKeyPassword=abtst
org.apache.ws.security.crypto.merlin.file=security/abTst_server.jks
org.apache.ws.security.crypto.merlin.keystore.password=htabern
```

Beispiel B.1. Standardkonfiguration Receiver

```
                #
# Copyright 1996-2013 itServe AG. All rights reserved.
#
# This software is the proprietary information of itServe AG
# Bern Switzerland. Use is subject to license terms.
#


# RECEIVER INDENTIFICATION
institution.tax.id=BE
institution.tas.id=BE

# DIRECTORY CONFIGURATION
released.filesystem.parent.xml=data/received
failed.filesystem.parent.xml=data/failed
result.filesystem.parent.xml=data/result
sent.filesystem.parent.xml=data/sent
undeliverable.filesystem.parent.xml=data/undeliverable
maintenance.filesystem.parent.xml=data/maintenance

# MONITORING
monitoring.enabled=false
monitoring.user=admin
monitoring.pass=admin

# BEHAVIOUR CONFIGURATION
commune.splitting=false
la.splitting=false
write.original.xml=false
await.result=false
processing.default.hours=168

housekeeping.days=180
```

Beispiel B.2. Standardkonfiguration Integration

```
                <?xml version='1.0' encoding='utf-8'?>
<!--

   Copyright 1996-2013 itServe AG. All rights reserved.

   This software is the proprietary information of itServe AG
   Bern Switzerland. Use is subject to license terms.

-->
<Server port="8005" shutdown="SHUTDOWN">

   <!--APR library loader. Documentation at /docs/apr.html -->
   <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on"/>
   <!--Initialize Jasper prior to webapps are loaded. Documentation at /docs/jasper-howto.html -->
   <Listener className="org.apache.catalina.core.JasperListener"/>
   <!-- Prevent memory leaks due to use of particular java/javax APIs-->
   <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>
   <!-- JMX Support for the Tomcat server. Documentation at /docs/non-existent.html -->
   <Listener className="org.apache.catalina.mbeans.ServerLifecycleListener"/>
   <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>

   <GlobalNamingResources>
      <Resource name="UserDatabase" auth="Container"
            type="org.apache.catalina.UserDatabase"
            description="User database that can be updated and saved"
            factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
            pathname="conf/tomcat-users.xml"/>
   </GlobalNamingResources>

   <Service name="Catalina">
      <Connector port="8080" protocol="HTTP/1.1"
            connectionTimeout="20000"/>

      <!-- Uncomment for activate SSL/TLS termination and mutual authentication -->
      <!-- <Connector port="8443" protocol="HTTP/1.1"
            connectionTimeout="20000"
            scheme="https" secure="true" SSLEnabled="true"
            keystoreFile="[path]/server.jks" keystorePass="keystore pass"
            truststoreFile="[path]/truststore.jks" truststorePass="truststore pass"
            clientAuth="true" sslProtocol="TLS"/> -->

      <Engine name="Catalina" defaultHost="localhost">
         <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase"/>
         <Host name="localhost" appBase="webapps" deployOnStartup="false"
            unpackWARs="false" autoDeploy="false"
            xmlValidation="false" xmlNamespaceAware="false">

            <Context path="/webservice" docBase="swissdecAdapter-webservice-2.0_5.war" reloadable="false"/>
         </Host>
      </Engine>
   </Service>
</Server>
```

Beispiel B.3. Standardkonfiguration Tomcat Receiver

```
                <?xml version='1.0' encoding='utf-8'?>
<!--

   Copyright 1996-2013 itServe AG. All rights reserved.

   This software is the proprietary information of itServe AG
   Bern Switzerland. Use is subject to license terms.

-->
<Server port="8006" shutdown="SHUTDOWN">

   <!--APR library loader. Documentation at /docs/apr.html -->
   <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on"/>
   <!--Initialize Jasper prior to webapps are loaded. Documentation at /docs/jasper-howto.html -->
   <Listener className="org.apache.catalina.core.JasperListener"/>
   <!-- Prevent memory leaks due to use of particular java/javax APIs-->
   <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>
   <!-- JMX Support for the Tomcat server. Documentation at /docs/non-existent.html -->
   <Listener className="org.apache.catalina.mbeans.ServerLifecycleListener"/>
   <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>

   <GlobalNamingResources>
      <Resource name="UserDatabase" auth="Container"
            type="org.apache.catalina.UserDatabase"
            description="User database that can be updated and saved"
            factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
            pathname="conf/tomcat-users.xml"/>
   </GlobalNamingResources>

   <Service name="Catalina">
      <Connector port="9090" protocol="HTTP/1.1"
               connectionTimeout="20000"/>

      <Engine name="Catalina" defaultHost="localhost">
         <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase"/>
         <Host name="localhost" appBase="webapps" deployOnStartup="false"
            unpackWARs="false" autoDeploy="false"
            xmlValidation="false" xmlNamespaceAware="false">

            <Context path="/integration" override="true" docBase="swissdecAdapter-integration-2.0_5.war"
                  reloadable="false"/>
         </Host>
      </Engine>
   </Service>
</Server>
```

Beispiel B.4. Standardkonfiguration Tomcat Integration

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--

   Copyright 1996-2013 itServe AG. All rights reserved.

   This software is the proprietary information of itServe AG
   Bern Switzerland. Use is subject to license terms.

-->
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration debug="false" xmlns:log4j="http://jakarta.apache.org/log4j/">
   <appender name="CATALINA" class="org.apache.log4j.DailyRollingFileAppender">
      <param name="File" value="${catalina.base}/logs/catalina.out"/>
      <param name="DatePattern" value="yyyy-MM-dd"/>
      <layout class="org.apache.log4j.PatternLayout">
         <param name="ConversionPattern" value="%d [%t] %-5p %c - %m%n"/>
      </layout>
   </appender>
   <appender name="FILE" class="org.apache.log4j.DailyRollingFileAppender">
      <param name="File" value="${catalina.base}/logs/swissdecAdapter.log"/>
      <param name="DatePattern" value="yyyy-MM-dd"/>
      <layout class="org.apache.log4j.PatternLayout">
         <param name="ConversionPattern" value="%d [%t] %-5p %c - %m%n"/>
      </layout>
   </appender>
   <appender name="ACCESS" class="org.apache.log4j.FileAppender">
      <param name="File" value="${catalina.base}/logs/access.log"/>
      <param name="ImmediateFlush" value="true"/>
      <layout class="org.apache.log4j.PatternLayout">
         <param name="ConversionPattern" value="%d [%t] %-5p %c - %m%n"/>
      </layout>
   </appender>

   <logger name="ch.itserve.lohnstandard">
      <level value="INFO"/>
   </logger>

   <logger name="ch.itserve.lohnstandard.receiver.webservice.SwissdecAdapterStartupListener">
      <level value="DEBUG"/>
   </logger>

   <logger name="ch.itserve.lohnstandard.receiver.integration.impl.SwissdecAdapterStartupListener">
      <level value="DEBUG"/>
   </logger>

   <logger name="org.apache.catalina">
      <level value="INFO"/>
      <appender-ref ref="CATALINA"/>
   </logger>

   <logger name="org.apache.ws">
      <level value="ERROR"/>
      <appender-ref ref="FILE"/>
   </logger>
   <logger name="org.hibernate">
      <level value="ERROR"/>
   </logger>
```

Beispiel B.5. Standardkonfiguration Logging

```
   <logger name="access">
      <level value="INFO"/>
      <appender-ref ref="ACCESS"/>
   </logger>

   <root>
      <level value="INFO"/>
```