

SAP Biller Consolidator **CUX-Services**



Release 1.5 SP3



Copyright

© Copyright 2004 SAP AG. Alle Rechte vorbehalten.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Die von SAP AG oder deren Vertriebsfirmen angebotenen Softwareprodukte können Softwarekomponenten auch anderer Softwarehersteller enthalten.

Microsoft, Windows, Outlook, und PowerPoint sind eingetragene Marken der Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, und Informix sind Marken oder eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern.

Oracle ist eine eingetragene Marke der Oracle Corporation.

UNIX, X/Open, OSF/1, und Motif sind eingetragene Marken der Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, und MultiWin sind Marken oder eingetragene Marken von Citrix Systems, Inc.

HTML, XML, XHTML und W3C sind Marken oder eingetragene Marken des W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java ist eine eingetragene Marke von Sun Microsystems, Inc.

JavaScript ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.






MaxDB ist eine Marke von MySQL AB, Schweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver und weitere im Text erwähnte SAP-Produkte und

-Dienstleistungen sowie die entsprechenden Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und anderen Ländern weltweit. Alle anderen Namen von Produkten und Dienstleistungen sind Marken der jeweiligen Firmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länderspezifische Unterschiede aufweisen.

In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden. Die vorliegenden Angaben werden von SAP AG und ihren Konzernunternehmen („SAP-Konzern“) bereitgestellt und dienen ausschließlich Informationszwecken. Der SAP-Konzern übernimmt keinerlei Haftung oder Garantie für Fehler oder Unvollständigkeiten in dieser Publikation. Der SAP-Konzern übernimmt lediglich diejenige Garantie für Produkte und Dienstleistungen, die in den gegebenenfalls mit den jeweiligen Produkten und Dienstleistungen gelieferten ausdrücklichen Garantiererklärungen festgelegt sind. Aus den in dieser Publikation enthaltenen Informationen ergibt sich keine weiterführende Garantie.

Symbole im Text

Symbol	Bedeutung
	Achtung
	Beispiel
	Hinweis
	Empfehlung
	Syntax

In der SAP-Dokumentation werden weitere Symbole verwendet, die verdeutlichen, welche Art von Informationen ein Text enthält. Weitere Informationen finden Sie auf der Startseite jeder Version der SAP-Bibliothek unter *Hilfe zur Hilfe* → *Allgemeine Informationsklassen* und *Informationsklassen für das Business Information Warehouse*.

Typografische Konventionen

Format	Beschreibung
<i>Beispieltext</i>	Wörter oder Zeichen, die vom Bildschirmbild zitiert werden. Dazu gehören Feldbezeichner, Bildtitel, Drucktastenbezeichner sowie Menünamen, Menüpfade und Menüeinträge. Querverweise auf andere Dokumentationen
Beispieltext	hervorgehobene Wörter oder Ausdrücke im Fließtext, Titel von Grafiken und Tabellen
BEISPIELTEXT	Namen von Systemobjekten. Dazu gehören Reportnamen, Programmnamen, Transaktionscodes, Tabellennamen und einzelne Schlüsselbegriffe einer Programmiersprache, die von Fließtext umrahmt sind, wie z.B. SELECT und INCLUDE
<code>Beispieltext</code>	Ausgabe auf der Oberfläche. Dazu gehören Datei- und Verzeichnisnamen und ihre Pfade, Meldungen, Quelltext, Namen von Variablen und Parametern sowie Namen von Installations-, Upgrade- und Datenbankwerkzeugen.
Beispieltext	exakte Benutzereingabe. Dazu gehören Wörter oder Zeichen, die Sie genau so in das System eingeben, wie es in der Dokumentation angegeben ist.
< Beispieltext >	variable Benutzereingabe. Die Wörter und Zeichen in spitzen Klammern müssen Sie durch entsprechende Eingaben ersetzen, bevor Sie sie in das System eingeben.
BEISPIELTEXT	Tasten auf der Tastatur, wie z.B. die Funktionstaste F2 oder die ENTER-Taste

CUX-Services	5
Aufbau der Request und –Response-Meldungen.....	6
SignOn Aggregat (CUX).....	11
Transaction Aggregat	13
EDIActivation.....	15
Schema – EDIActivation und EDIDeactivation.....	18
DTD – EDIActivation and EDIDeactivation	24
EDIDeactivation	25
Schema – EDIActivation und EDIDeactivation.....	27
DTD – EDIActivation and EDIDeactivation	33



Einsatzmöglichkeiten

Die CUX-Services bieten dem Customer die Möglichkeit, über XML-Meldungen mit dem iCSP zu kommunizieren.

Einführungshinweise

Die Meldungen sind als Request-Response-Messages definiert. Sämtliche BBX- und CUX-Services sowie die CCX-Meldungen sind in der gleichen Art aufgebaut. In jedem Request/Response werden neben dem eigentlichen Aggregat die Aggregate `SignOn` und `Trn` mitgegeben, die allgemeine Informationen zum Request/Response enthalten.

Integration

Der Customer benötigt für die Verarbeitung der strukturierten Daten eine Software. Dieses Programm ist nicht Bestandteil der Lösung *SAP Biller Consolidator*.

Funktionsumfang

SAP Biller Consolidator bietet zwei CUX-Services an:

- [EDIActivation \[Seite 15\]](#)
- [EDIDeactivation \[Seite 25\]](#)

Die Meldungen für die EDI-Aktivierung und –Deaktivierung können für Customer genutzt werden, die sich über eine Banken-Applikation bei dem Betreiber von *SAP Biller Consolidator* angemeldet haben, aber dennoch für ihre Buchhaltung eine Offline-Finanz-Software verwenden wollen. Für die Verarbeitung der Rechnung benötigt diese Software Rechnungsdaten in einem strukturierten Format. Die Daten können durch *SAP Biller Consolidator* im Format UN/EDIFACT, XML oder IDoc zur Verfügung gestellt werden. Damit der Betreiber des *SAP Biller Consolidator* erfahren kann, ob ein Customer EDI-Daten benötigt, kann sich dieser mit der Meldung `EDIActivation` bei dem Consolidator direkt anmelden. Die Meldung sollte in der entsprechenden Offline-Finanz-Software integriert sein.



Aufbau der Request und –Response-Meldungen

Definition

Für die Lösung *SAP Biller Consolidator* standardisierte XML-Requests und –Responses, die für mehrere Schnittstellen verwendet werden können.

Verwendung

Grundsätzlich sind die Request- und Response-Meldungen, die für den Austausch von XML-Meldungen zwischen den Teilnehmern verwendet werden, gleich aufgebaut. Die Requests und Responses gelten für folgende Schnittstellen:

- **CCX**: Meldungsaustausch zwischen Consolidator und externem CSP
- **CPX**: Meldungsaustausch zwischen Consolidator und dem CSP-Portal von SAP
- **BBX**: Meldungsaustausch zwischen Biller und BSP ([BBX-Services \[Extern\]](#))
- **CUX**: Meldungsaustausch zwischen Consolidator und einem User Interface beim Customer ([CUX-Services \[Seite 1\]](#)).

Struktur

Der Aufbau der Meldungen ist für CCX, CPX, BBX und CUX grundsätzlich gleich. Das umfassende Tag wird je Meldungstyp verschieden bezeichnet. Der Namespace bezieht sich dann auf die jeweilige Schnittstelle.

Die Aggregate `SignOn` und `Trn` werden in jeder Meldung als `Rq` oder `Rs` mitgegeben. Die eigentliche Meldung wird unter das `Trn`-Aggregat eingefügt.

Jede Funktion die als CCX, BBX- oder CUX-Service zur Verfügung steht, verlangt als Eingabeparameter eine XML-Message und gibt als Returnwert auch wieder eine XML-Message zurück.

Falls bei der Verarbeitung eines Requests ein Fehler aufgetreten ist, wird die entsprechende Fehlermeldung in der Response im Status-Aggregat mitgeteilt und die eigentliche Response weggelassen.

Für die Integration der XML-Meldungen werden DTDs und Schemata geliefert. Des weiteren sind in den folgenden Abschnitten die Meldungen in Tabellenform mit zusätzlichen Informationen hinterlegt.

Aufbau der Tabellen

Die Requests und Responses sind in Tabellen in den folgenden Unterkapiteln aufgegliedert. Die Tabellen enthalten folgende Rubriken:

- XML-Tag
- Format
- Typ
- Beschreibung

Unter XML-Tag werden die Aggregate, Elemente und Attribute aufgelistet. Aggregate werden immer mit Tag und End-Tag gekennzeichnet. Elemente werden einzeln gekennzeichnet. Attribute werden ohne Tags beschrieben. In der Spalte *Typ* wird die Unterscheidung nochmals verdeutlicht:

- A = Aggregat

- E = Element
- Attr. = Attribut

Ebenfalls im Typ wird die zulässige Menge des Tags erwähnt:

- 1 = nur ein Mal
- 0..1 = max. ein Mal (XML-Syntax: ?)
- 0..n = unbeschränkt (XML-Syntax: *)
- 1..n = beliebig viele, aber eines muss mind. kommen (XML-Syntax: +)

Fett gekennzeichnete Elemente oder Aggregate müssen mind. ein Mal angegeben werden.

Unter Format wird die Ausprägung der Meldungsinhalte dargestellt:

- an = alphanumerisch
- n = numerisch

Falls zwischen dem Format und der Anzahl der Stellen zwei Punkte aufgeführt sind, kann zwischen 0 und der angegebenen Anzahl die Länge frei gewählt werden. Ist das Format und die Anzahl der Stellen zusammengeschrieben, darf nur exakt diese Länge verwendet werden.

Datenelemente

Request Messages

XML-Tag	Format	Typ	Beschreibung
<xxx>	-	-	Name des umfassenden Aggregats. Folgende Aggregate werden unterstützt: <u>Für CCX-Meldungen:</u> Name des Input-Parameters <u>Für CPX-Meldungen:</u> Name des Input-Parameters <u>Für BBX-Services:</u> CustomerRegistrationMessage RequestPDFRequestMessage <u>Für CUX-Services:</u> EDIActivationMessageRequest EDIDeactivationMessageRequest
urn:xxxInterfaces:DataTypes	Siehe definierte Werte	xmlns, 1	XML-Namespace, für den die Elemente gelten. Folgende Werte sind als Ersatz für xxx möglich: CCX = Consolidator CSP Exchange CPX = Consolidator Portal Exchange BBX = Biller BSP Exchange CUX = Customer Userdata Exchange
<SignOnRq> </SignOnRq>	-	A, 1	SignOnRq-Aggregat, das den Sender der Meldung identifiziert (siehe SignOn Aggregat [Extern])
<TrnRq>	-	A, 1	TrnRq-Aggregat; ermöglicht die

</TrnRq>			eindeutige Zuordnung von einem Response zu einem Request (siehe Transaction Aggregat [Seite 13])
<xxxRequest> </xxxRequest>	-	A, 1	Request-Aggregat der eigentlichen Meldung, die versendet werden soll. z. B. bei einer CCX-Meldung für die Ablehnung einer Rechnung: <RejectBillRq>
</xxx>	-	-	

Response Messages

XML-Tag	Format	Typ	Beschreibung
<xxx>	-	-	Name des umfassenden Aggregats. Folgende Aggregate werden unterstützt: <u>Für CCX-Meldungen:</u> Name des Output-Parameters <u>Für CPX-Meldungen:</u> Name des Output-Parameters <u>Für BBX-Services:</u> CustomerRegistrationMessageResponse RequestPDFMessageResponse <u>Für CUX-Services:</u> EDIActivationMessageResponse EDIDeactivationMessageResponse
urn:xxXInterfaces:DataTypes	Siehe definierte Werte	xmlns, 1	XML-Namespace, für den die Elemente gelten. Folgende Werte sind als Ersatz für xxX möglich: CCX = Consolidator CSP Exchange CPX = Consolidator Portal Exchange BBX = Biller BSP Exchange CUX = Customer Userdata Exchange
<SignOnRs> </SignOnRs>	-	A, 1	SignOnRs-Aggregat, das den Sender der Meldung identifiziert (siehe SignOn Aggregat [Extern])
<TrnRs> </TrnRs>	-	A, 1	TrnRs-Aggregat; ermöglicht die eindeutige Zuordnung von einem Response zu einem Request (siehe Transaction Aggregat [Seite 13])
<xxxResponse> </xxxResponse>	-	A, 0..1	Response-Aggregat der eigentlichen Meldung, die versendet werden soll. Wenn in dem Request ein Fehler aufgetreten ist, kann das Aggregat weggelassen werden. z. B. bei einer CCX-Meldung für die

			Ablehnung einer Rechnung: <RejectBillRs>
</xxx>	-	-	-

Beispiel



Beispiel 1 für eine eingehende CCX-Meldung:

```
<ccx:EnrollMsg xmlns:ccx="urn:CCXInterfaces:DataTypes">
  <ccx:SignOnRq>
    .
    .
  </ccx:SignOnRq>
  <ccx:TrnRq>
    <ccx:TrnUID>12345</ccx:TrnUID>
  </ccx:TrnRq>
  <ccx:EnrollRq>
    .
    .
  </ccx:EnrollRq>
</ccx:EnrollMsg>
```



Beispiel 2 für eine eingehende CCX-Meldung:

```
<EnrollMsg xmlns="urn:CCXInterfaces:DataTypes">
  <SignOnRq>
    .
    .
  </SignOnRq>
  <TrnRq>
    <TrnUID>12345</TrnUID>
  </TrnRq>
  <EnrollRq>
    .
    .
  </EnrollRq>
</EnrollMsg>
```

**Beispiel für eine ausgehende CCX-Meldung (Reply):**

```
<ccx:EnrollMsgReply
xmlns:ccx="urn:CCXInterfaces:DataTypes">
  <ccx:SignOnRs>
    <ccx:Status>
      <ccx:Code>0</Code>
      <ccx:Severity>INFO</Severity>
    </ccx:Status>
  </ccx:SignOnRs>
  <ccx:TrnRs>
    <ccx:TrnUID>12345</ccx:TrnUID>
    <ccx:Status>
      <ccx:Code>0</ccx:Code>
      <ccx:Severity>INFO</ccx:Severity>
    </ccx:Status>
  </ccx:TrnRs>
  <ccx:EnrollRs>
    .
    .
  </ccx:EnrollRs>
</ccx:EnrollMsgReply>
```



SignOn Aggregat (CUX)

Definition

Aggregat innerhalb der CUX-Services, das ein Identifizieren des Senders ermöglicht.

Verwendung

Der erste Request innerhalb einer Meldung muss zwingend ein SignOn sein.

Kann der SignOn nicht durchgeführt werden, gibt der nachfolgende Request den Error Code 5000 (kein gültiger SignOn) zurück.

Struktur

1. Der Sender schickt seine PID und die gewünschte Sprache im SignOn Request an den BSP oder Consolidator.
2. Der BSP oder Consolidator prüft, ob der DN des HTTPS-Clients und der für diese PID ermittelte DN zusammen gehören. Wenn eine nicht unterstützte Sprache angegeben wurde, wird als Default im Response GER für *german* zurückgegeben.

Datenelemente

SignOn Request

XML-Tag	Format	Typ	Beschreibung
<SignOnRq>			SignOnRq Aggregat
<DtClient>	Datetime	E, 1	Datum und Zeit des Requests
<ClientPID>	n 17	E, 1	Eindeutige Teilnehmer-Identifikation für den Kunden
<Language>	an 3	E, 1	Sprache für Text Responses nach ISO-639-2. Folgende Werte sind möglich: GER, ger = deutsch ENG, eng = englisch FRA, fra = französisch FRE, fre = französisch ITA, ita = italienisch
</SignOnRq>			

SignOn Response

XML-Tag	Format	Typ	Beschreibung
<SignOnRs>	-		SignOnRs-Aggregat
<Status> </Status>	-	A, 1	Status-Aggregat
<DtClient>	Datetime	E, 1	Datum und Zeit des Requests
<DtServer>	Datetime	E, 1	Datum und Zeit des Response

<Language>	an 3	E, 1	Sprache für Text Responses nach ISO-639-2. Folgende Werte sind möglich: GER, ger = deutsch ENG, eng = englisch FRA, fra = französisch FRE, fre = französisch ITA, ita = italienisch
</SignOnRs>	-	-	

Status Aggregat

XML-Tag	Format	Typ	Beschreibung
<Status>	-	-	
<Code>	n6	E, 1	Status-Code (siehe Tabelle unten)
<Severity>	Siehe definierte Werte	E, 1	Enum für Schweregrad; Folgende Werte sind möglich: INFO = Information WARNING = Warnung ERROR = Fehler Es kann nur ein Wert angegeben werden.
<Message>	an..255	E, 0..1	Beschreibung des Fehlers
</Status>	-	-	

SignOn Status Codes

Code	Beschreibung	Severity
0	OK	INFO
1100	Customer nicht bekannt	ERROR
1120	Sender nicht bekannt	ERROR
1150	Sender nicht vom Customer für Einlieferungen ermächtigt	ERROR
1400	Syntaxfehler	ERROR



Transaction Aggregat

Definition

Aggregat innerhalb der CCX-Meldungen und der BBX- und CUX-Services, das eine eindeutige Zuordnung einer Response zu einem Request ermöglicht.

Verwendung

Das Transaction Aggregat ermöglicht die eindeutige Zuordnung einer Response zu seinem Request. Jede Meldung hat ein Transaction Request, der eine eindeutige Transaction-ID enthält. Die dazugehörige MessageReply hat eine Transaction Response, die dieselbe Transaction-ID und zusätzlich ein Status Aggregat enthält.

Wenn das Status Aggregat einen Fehler beinhaltet (Severity = ERROR), wird die Response weggelassen.

Struktur

In jeder Meldung muss im Transaction Request vom Client eine für ihn eindeutige Transaction-ID <TrnUID> vergeben werden. Im Transaction Response wird diese <TrnUID> und zusätzlich ein Status-Aggregat wieder zurückgegeben.

Datenelemente

Request Transaction Aggregat

XML-Tag	Format	Typ	Beschreibung
<TrnRq>	-	-	Request Transaction Aggregat
<TrnUID>	an..35	E, 1	Vom Client, für diese Transaktion erzeugte eindeutige Identifikation
<ClientCookie>	an..256	E, 0..1	Daten, die der Client mitgeben kann und die im Transaction Response wieder zurückgegeben werden.
</TrnRq>	-	-	

Response Transaction Aggregat

XML-Tag	Format	Typ	Beschreibung
<TrnRs>	-	-	Response Transaction Aggregat
<TrnUID>	an..35	E, 1	Vom Client, für diese Transaktion erzeugte eindeutige Identifikation
<ClientCookie>	an..256	E, 0..1	Daten, die vom Client dem Request Transaction Aggregat mitgegeben wurden.
<Status> </Status>	-	A, 1	Status-Aggregat (siehe unten)
</TrnRs>	-	-	

Status Aggregat

XML-Tag	Format	Typ	Beschreibung
<Status>	-	-	
<Code>	n6	E, 1	Status-Code; die Status-Code sind pro Meldung verschieden und sind bei den jeweiligen Meldungen beschrieben.
<Severity>	Siehe definierte Werte	E, 1	Enum für Schweregrad; Folgende Werte sind möglich: INFO = Information WARNING = Warnung ERROR = Fehler Es kann nur ein Wert angegeben werden.
<Message>	an..255	E, 0..1	Beschreibung des Fehlers
</Status>	-	-	



EDIActivation

Definition

CUX-Meldung an den Consolidator für die Aktivierung eines Customers für EDI-Meldungen.

Verwendung

Die Meldung EDIActivation wird verwendet, damit sich Customer mit einer einfachen Meldung für die Zustellung von Rechnungen im Format EDI anmelden können. Diese Meldung wird nicht direkt vom Customer selbst erzeugt und versendet, sondern kann in Finanz-Software für den privaten Gebrauch bzw. für kleinere Firmen integriert werden. Mit Hilfe der Schnittstelle kann der Customer EDI-Rechnungen erhalten, die in seine Finanz-Software abgespeichert werden können. Bei dem Empfang von den relevanten Rechnungsdaten kann die Meldung für mehrwertsteuerliche Aspekte geltend gemacht werden.

Die Meldung unterstützt folgende Funktionen:

- Erstmalige EDI-Aktivierung
- Änderung der Daten einer vorherigen EDI-Aktivierung

Struktur

Das Aggregat EDIActivationRequest wird in das folgende Aggregat eingefügt (siehe [Aufbau der Request- und Response-Meldungen \[Seite 6\]](#)):

```
<EDIActivationMessage xmlns="urn:CUXInterfaces:DataTypes">
```

Das Aggregat EDIActivationResponse existiert nicht. Ein Response beinhaltet nur die Aggregat `SignOn` und `Trn` (siehe [Aufbau der Request- und Response-Meldungen \[Seite 6\]](#)). Die Response ist in folgendem Aggregat definiert.

```
<EDIActivationMessageResponse  
  xmlns="urn:CUXInterfaces:DataTypes">
```

In jedem Request/Response werden neben dem eigentlichen Aggregat die Aggregate `SignOn` und `Trn` mitgegeben, die allgemeine Informationen zum Request/Response enthalten.

Auf alle Requests wird ein Response geliefert. In der Response können für diese Meldung Fehlercodes in den Status-Aggregaten von `SignOnRs` und `TrnRs` zurückgegeben werden (siehe Tabelle unten).

Des weiteren wird beim BSP geprüft, ob die CPID in allen Requests zum gleichen Partner gehört wie der Benutzername und das Aktivierungspasswort vom Login.

Datenelemente

EDIActivationRequest

XML-Tag	Format	Typ	Beschreibung
<EDIActivationRequest>	-	-	Aktivierung eines Customers für die Zustellung von EDI-Meldungen

<CustomerPID>	n17	E, 1	Eindeutige Teilnehmer-Identifikation des Customers
<CommunicationEDI>	-	A, 0..1	Aggregat mit Informationen für die Zustellung der Rechnung in einem EDI-Format
<InvoiceFormat>	Siehe definierte Werte	E, 1	Rechnungsformat; Folgende Werte sind möglich: EIXML = XML-FSCM-Rechnung EIXML_LIGHT = XML-FSCM-Rechnung ohne Line-Items IDoc = Idoc-XML-Rechnung EDIFACT = UN/EDIFACT-Rechnung
<ID>	an	E, 1	Identifikation des Customers. Mit diesem Wert werden die Kommunikationseinträge erstellt.
IDType	Siehe definierte Werte	Attr. 1	Attribut zu dem Element ID. Folgende Werte sind möglich: DN = Distinguished Name [Extern] USERNAME = Benutzername (bei Verwendung mit Passwort)
<Password>	an	E, 1	Passwort wird nur bei benötigt, wenn der IDType USERNAME verwendet wurde.
</CommunicationEDI>	-	-	-
<CommunicationPDF>	-	A, 0..1	Aggregat mit Informationen für die Zustellung der Rechnung als PDF
<ID>	an	E, 1	Identifikation des Customers. Mit diesem Wert werden die Kommunikationseinträge erstellt.
IDType	Siehe definierte Werte	Attr. 1	Attribut zu dem Element ID. Folgende Werte sind möglich: DN = Distinguished Name [Extern] USERNAME = Benutzername (bei Verwendung mit Passwort)
<Password>	an	E, 0..1	Passwort wird nur benötigt, wenn der IDType USERNAME verwendet wurde.
</CommunicationPDF>	-	-	-
</EDIActivationRequest>	-	-	-

Fehlercodes

Status Code	Beschreibung	Severity
0	Ok	INFO
5000	SignOn Fehler (siehe SignOnRs Status Code)	ERROR



Schema – EDIActivation und EDIDeactivation

```

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

<xs:schema targetNamespace="urn:CUXInterfaces:DataTypes"
  xmlns:cux="urn:CUXInterfaces:DataTypes"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:simpleType name="PID">
    <xs:restriction base="xs:string">
      <xs:minLength value="17"/>
      <xs:maxLength value="17"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="ID">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="IDType" type="cux:IDType"
            use="required"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:element name="CustomerPID" type="cux:PID"/>
  <xs:element name="TrnRs">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cux:TrnUID"/>
        <xs:element ref="cux:ClientCookie"
          minOccurs="0"/>
        <xs:element ref="cux:Status"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="TrnUID" type="xs:ID"/>
  <xs:element name="ClientCookie" type="xs:string"/>
  <xs:element name="InvoiceFormat">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="IDoc"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

```

```
        <xs:enumeration value="EDIFACT"/>
        <xs:enumeration value="EIXML"/>
        <xs:enumeration value="EIXML_LIGHT"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Status">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Code">
          <xs:simpleType>
            <xs:restriction base="xs:int">
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="999999"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Severity">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="INFO"/>
              <xs:enumeration value="WARNING"/>
              <xs:enumeration value="ERROR"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Message" type="xs:string"
minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SignOnRs">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cux:Status"/>
        <xs:element name="DtClient" type="xs:dateTime"/>
        <xs:element name="DtServer" type="xs:dateTime"/>
        <xs:element ref="cux:Language"/>
      </xs:sequence>
    </xs:complexType>
```

```
</xs:element>
<xs:element name="SignOnRq">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="DtClient" type="xs:dateTime"/>
      <xs:element name="ClientPID" type="cux:PID"/>
      <xs:element ref="cux:Language"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TrnRq">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:TrnUID"/>
      <xs:element ref="cux:ClientCookie"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="EDIActivationRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:CustomerPID"/>
      <xs:sequence>
        <xs:element ref="cux:CommunicationEDI"
minOccurs="0"/>
        <xs:element ref="cux:CommunicationPDF"
minOccurs="0"/>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="EDIDeactivationRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:CustomerPID"/>
      <xs:sequence>
        <xs:element ref="cux:DeactivationEDI"
minOccurs="0"/>
        <xs:element ref="cux:DeactivationPDF"
minOccurs="0"/>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="CommunicationEDI">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="cux:InvoiceFormat"/>
          <xs:element ref="cux:ID"/>
          <xs:element name="Password" type="xs:string"
minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="CommunicationPDF">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="cux:ID"/>
          <xs:element name="Password" type="xs:string"
minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Language">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="3"/>
          <xs:maxLength value="3"/>
          <xs:enumeration value="GER"/>
          <xs:enumeration value="FRA"/>
          <xs:enumeration value="ITA"/>
          <xs:enumeration value="ENG"/>
          <xs:enumeration value="FRE"/>
          <xs:enumeration value="ger"/>
          <xs:enumeration value="fra"/>
          <xs:enumeration value="fre"/>
          <xs:enumeration value="ita"/>
          <xs:enumeration value="eng"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="EDIActivationMessage">
```

```
<xs:complexType>
  <xs:sequence>
    <xs:element ref="cux:SignOnRq"/>
    <xs:element ref="cux:TrnRq"/>
    <xs:element ref="cux:EDIActivationRequest"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="DeactivationEDI">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:InvoiceFormat"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="DeactivationPDF"/>
<xs:element name="EDIDeactivationMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:SignOnRq"/>
      <xs:element ref="cux:TrnRq"/>
      <xs:element ref="cux:EDIDeactivationRequest"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:simpleType name="IDType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DN"/>
    <xs:enumeration value="USERNAME"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="EDIActivationMessageResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:SignOnRs"/>
      <xs:element ref="cux:TrnRs"/>
      <xs:element ref="cux:EDIActivationResponse"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="EDIDeactivationMessageResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:SignOnRs"/>
      <xs:element ref="cux:TrnRs"/>
      <xs:element ref="cux:EDIDeactivationResponse"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="EDIDeactivationResponse">
  <xs:complexType/>
</xs:element>
<xs:element name="EDIActivationResponse">
  <xs:complexType/>
</xs:element>
</xs:schema>
```



DTD – EDIActivation and EDIDeactivation

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT ClientCookie (#PCDATA)>
<!ELEMENT CommunicationEDI (InvoiceFormat, ID, Password?)>
<!ELEMENT CommunicationPDF (ID, Password?)>
<!ELEMENT CustomerPID (#PCDATA)>
<!ELEMENT DeactivationEDI (InvoiceFormat)>
<!ELEMENT DeactivationPDF ANY>
<!ELEMENT EDIActivationMessage (SignOnRq, TrnRq,
EDIActivationRequest)>
<!ELEMENT EDIActivationMessageResponse (SignOnRs, TrnRs,
EDIActivationResponse)>
<!ELEMENT EDIActivationRequest (CustomerPID,
(CommunicationEDI?, CommunicationPDF?))>
<!ELEMENT EDIActivationResponse EMPTY>
<!ELEMENT EDIDeactivationMessage (SignOnRq, TrnRq,
EDIDeactivationRequest)>
<!ELEMENT EDIDeactivationMessageResponse (SignOnRs, TrnRs,
EDIDeactivationResponse)>
<!ELEMENT EDIDeactivationRequest (CustomerPID,
(DeactivationEDI?, DeactivationPDF?))>
<!ELEMENT EDIDeactivationResponse EMPTY>
<!ELEMENT ID (#PCDATA)>
<!ATTLIST ID
  IDType (DN | USERNAME) #REQUIRED
>
<!ELEMENT InvoiceFormat (#PCDATA)>
<!ELEMENT Language (#PCDATA)>
<!ELEMENT SignOnRq (DtClient, ClientPID, Language)>
<!ELEMENT SignOnRs (Status, DtClient, DtServer, Language)>
<!ELEMENT Status (Code, Severity, Message?)>
<!ELEMENT TrnRq (TrnUID, ClientCookie?)>
<!ELEMENT TrnRs (TrnUID, ClientCookie?, Status)>
<!ELEMENT TrnUID (#PCDATA)>
<!ELEMENT Password (#PCDATA)>
<!ELEMENT DtClient (#PCDATA)>
<!ELEMENT ClientPID (#PCDATA)>
<!ELEMENT DtServer (#PCDATA)>
<!ELEMENT Code (#PCDATA)>
<!ELEMENT Severity (#PCDATA)>
<!ELEMENT Message (#PCDATA)>
```




EDIDeactivation

Definition

CUX-Meldung an den Consolidator für die Deaktivierung eines Customers für EDI-Meldungen.

Verwendung

Die Meldung EDIDeactivation wird verwendet, damit sich Customer mit einer einfachen Meldung für die Zustellung von Rechnungen im Format EDI abmelden können. Diese Meldung wird nicht direkt vom Customer selbst erzeugt und versendet, sondern kann in Finanz-Software für den privaten Gebrauch bzw. für kleinere Firmen integriert werden. Mit Hilfe der Schnittstelle kann der Customer EDI-Rechnungen erhalten, die in seine Finanz-Software abgespeichert werden können. Bei dem Empfang von den relevanten Rechnungsdaten kann die Meldung für mehrwertsteuerliche Aspekte geltend gemacht werden.

Die Meldung unterstützt folgende Funktionen:

- EDI-Deaktivierung

Struktur

Das Aggregat EDIDeactivationRequest wird in das folgende Aggregat eingefügt (siehe [Aufbau von Request- und Response-Meldungen \[Seite 6\]](#)):

```
<EDIDeactivationMessage  
  xmlns="urn:CUXInterfaces:DataTypes">
```

Das Aggregat EDIDeactivationResponse existiert nicht. Ein Response beinhaltet nur die Aggregat SignOn und Trn (siehe [Aufbau von Request- und Response-Meldungen \[Seite 6\]](#)). Die Response ist in folgendem Aggregat definiert.

```
<EDIDeactivationMessageResponse  
  xmlns="urn:CUXInterfaces:DataTypes">
```

In jedem Request/Response werden neben dem eigentlichen Aggregat die Aggregate SignOn und Trn mitgegeben, die allgemeine Informationen zum Request/Response enthalten.

Auf alle Requests wird ein Response geliefert. In der Response können für diese Meldung Fehlercodes in den Status-Aggregaten von SignOnRs und TrnRs zurückgegeben werden (siehe Tabelle unten).

Des weiteren wird beim BSP geprüft, ob die CPID in allen Requests zum gleichen Partner gehört wie der Benutzername und das Aktivierungspasswort vom Login.

Datenelemente

EDIDeactivationRequest

XML-Tag	Format	Typ	Beschreibung
<EDIDeactivationRequest>	-	-	Aktivierung eines Customers für die

			Zustellung von EDI-Meldungen
<CustomerPID>	n17	E, 1	Eindeutige Teilnehmer-Identifikation des Customers
<DeactivationEDI>	-	A, 0..1	Aggregat mit Informationen, welches EDI-Format deaktiviert werden soll. Damit werden im System die betreffenden Kommunikationseinträge gelöscht.
<InvoiceFormat>	Siehe definierte Werte	E, 1	Rechnungsformat; Folgende Werte sind möglich: EIXML = XML-FSCM-Rechnung EIXML_LIGHT = XML-FSCM-Rechnung ohne Line-Items IDoc = Idoc-XML-Rechnung EDIFACT = UN/EDIFACT-Rechnung
</DeactivationEDI>	-	-	-
<DeactivationPDF>	empty	E, 0..1	Leeres Element. Wenn vorhanden, dann soll die Zustellung von PDFs deaktiviert werden.
</EDIDeactivationRequest>	-	-	-

Fehlercodes

Status Code	Beschreibung	Severity
0	Ok	INFO
5000	SignOn Fehler (siehe SignOnRs Status Code)	ERROR



Schema – EDIActivation und EDIDeactivation

```

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

<xs:schema targetNamespace="urn:CUXInterfaces:DataTypes"
  xmlns:cux="urn:CUXInterfaces:DataTypes"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:simpleType name="PID">
    <xs:restriction base="xs:string">
      <xs:minLength value="17"/>
      <xs:maxLength value="17"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="ID">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="IDType" type="cux:IDType"
            use="required"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:element name="CustomerPID" type="cux:PID"/>
  <xs:element name="TrnRs">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cux:TrnUID"/>
        <xs:element ref="cux:ClientCookie"
          minOccurs="0"/>
        <xs:element ref="cux:Status"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="TrnUID" type="xs:ID"/>
  <xs:element name="ClientCookie" type="xs:string"/>
  <xs:element name="InvoiceFormat">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="IDoc"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

```

```
        <xs:enumeration value="EDIFACT"/>
        <xs:enumeration value="EIXML"/>
        <xs:enumeration value="EIXML_LIGHT"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Status">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Code">
          <xs:simpleType>
            <xs:restriction base="xs:int">
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="999999"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Severity">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="INFO"/>
              <xs:enumeration value="WARNING"/>
              <xs:enumeration value="ERROR"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Message" type="xs:string"
minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SignOnRs">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cux:Status"/>
        <xs:element name="DtClient" type="xs:dateTime"/>
        <xs:element name="DtServer" type="xs:dateTime"/>
        <xs:element ref="cux:Language"/>
      </xs:sequence>
    </xs:complexType>
```

```
</xs:element>
<xs:element name="SignOnRq">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="DtClient" type="xs:dateTime"/>
      <xs:element name="ClientPID" type="cux:PID"/>
      <xs:element ref="cux:Language"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TrnRq">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:TrnUID"/>
      <xs:element ref="cux:ClientCookie"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="EDIActivationRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:CustomerPID"/>
      <xs:sequence>
        <xs:element ref="cux:CommunicationEDI"
minOccurs="0"/>
        <xs:element ref="cux:CommunicationPDF"
minOccurs="0"/>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="EDIDeactivationRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:CustomerPID"/>
      <xs:sequence>
        <xs:element ref="cux:DeactivationEDI"
minOccurs="0"/>
        <xs:element ref="cux:DeactivationPDF"
minOccurs="0"/>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="CommunicationEDI">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="cux:InvoiceFormat"/>
          <xs:element ref="cux:ID"/>
          <xs:element name="Password" type="xs:string"
minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="CommunicationPDF">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="cux:ID"/>
          <xs:element name="Password" type="xs:string"
minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Language">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="3"/>
          <xs:maxLength value="3"/>
          <xs:enumeration value="GER"/>
          <xs:enumeration value="FRA"/>
          <xs:enumeration value="ITA"/>
          <xs:enumeration value="ENG"/>
          <xs:enumeration value="FRE"/>
          <xs:enumeration value="ger"/>
          <xs:enumeration value="fra"/>
          <xs:enumeration value="fre"/>
          <xs:enumeration value="ita"/>
          <xs:enumeration value="eng"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  <xs:element name="EDIActivationMessage">
```

```
<xs:complexType>
  <xs:sequence>
    <xs:element ref="cux:SignOnRq"/>
    <xs:element ref="cux:TrnRq"/>
    <xs:element ref="cux:EDIActivationRequest"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="DeactivationEDI">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:InvoiceFormat"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="DeactivationPDF"/>
<xs:element name="EDIDeactivationMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:SignOnRq"/>
      <xs:element ref="cux:TrnRq"/>
      <xs:element ref="cux:EDIDeactivationRequest"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:simpleType name="IDType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DN"/>
    <xs:enumeration value="USERNAME"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="EDIActivationMessageResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:SignOnRs"/>
      <xs:element ref="cux:TrnRs"/>
      <xs:element ref="cux:EDIActivationResponse"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="EDIDeactivationMessageResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="cux:SignOnRs"/>
      <xs:element ref="cux:TrnRs"/>
      <xs:element ref="cux:EDIDeactivationResponse"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="EDIDeactivationResponse">
  <xs:complexType/>
</xs:element>
<xs:element name="EDIActivationResponse">
  <xs:complexType/>
</xs:element>
</xs:schema>
```




DTD – EDIActivation and EDIDeactivation

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT ClientCookie (#PCDATA)>
<!ELEMENT CommunicationEDI (InvoiceFormat, ID, Password?)>
<!ELEMENT CommunicationPDF (ID, Password?)>
<!ELEMENT CustomerPID (#PCDATA)>
<!ELEMENT DeactivationEDI (InvoiceFormat)>
<!ELEMENT DeactivationPDF ANY>
<!ELEMENT EDIActivationMessage (SignOnRq, TrnRq,
EDIActivationRequest)>
<!ELEMENT EDIActivationMessageResponse (SignOnRs, TrnRs,
EDIActivationResponse)>
<!ELEMENT EDIActivationRequest (CustomerPID,
(CommunicationEDI?, CommunicationPDF?))>
<!ELEMENT EDIActivationResponse EMPTY>
<!ELEMENT EDIDeactivationMessage (SignOnRq, TrnRq,
EDIDeactivationRequest)>
<!ELEMENT EDIDeactivationMessageResponse (SignOnRs, TrnRs,
EDIDeactivationResponse)>
<!ELEMENT EDIDeactivationRequest (CustomerPID,
(DeactivationEDI?, DeactivationPDF?))>
<!ELEMENT EDIDeactivationResponse EMPTY>
<!ELEMENT ID (#PCDATA)>
<!ATTLIST ID
  IDType (DN | USERNAME) #REQUIRED
>
<!ELEMENT InvoiceFormat (#PCDATA)>
<!ELEMENT Language (#PCDATA)>
<!ELEMENT SignOnRq (DtClient, ClientPID, Language)>
<!ELEMENT SignOnRs (Status, DtClient, DtServer, Language)>
<!ELEMENT Status (Code, Severity, Message?)>
<!ELEMENT TrnRq (TrnUID, ClientCookie?)>
<!ELEMENT TrnRs (TrnUID, ClientCookie?, Status)>
<!ELEMENT TrnUID (#PCDATA)>
<!ELEMENT Password (#PCDATA)>
<!ELEMENT DtClient (#PCDATA)>
<!ELEMENT ClientPID (#PCDATA)>
<!ELEMENT DtServer (#PCDATA)>
<!ELEMENT Code (#PCDATA)>
<!ELEMENT Severity (#PCDATA)>
<!ELEMENT Message (#PCDATA)>
```