



*Der SAP-Conversion-Agent
von Itemfield*

Conversion-Agent- Bibliotheken

Version 4

Conversion-Agent-Bibliotheken

© 2006 Itemfield Inc. Alle Rechte vorbehalten.

Itemfield hat möglicherweise Patente, angemeldete Patente, Marken, Urheberrechte oder sonstige Rechte an geistigem Eigentum inne, die Inhalte dieses Dokumentes abdecken. Sofern nichts anderes ausdrücklich in einem schriftlichen Lizenzvertrag mit Itemfield vereinbart wurde, erhalten Sie durch die Bereitstellung dieses Dokumentes keinerlei Anspruch auf diese Patente, Marken, Urheberrechte oder auf sonstiges geistiges Eigentum.

Die in diesem Dokument enthaltenen Informationen können jederzeit ohne vorherige Ankündigung geändert werden. Der Anwender trägt die Verantwortung für die Einhaltung aller anwendbaren Urheberrechte. Ohne ausdrückliche schriftliche Genehmigung durch Itemfield Inc. darf kein Teil dieses Dokumentes in irgendeiner Form oder auf irgendeine elektronische oder mechanische Weise zu irgendeinem Zweck vervielfältigt oder weitergegeben werden.

SAP AG
<http://www.sap.com>

Dokumentdaten:

Version: 4
Datum: September 2006

Inhalt

1. Übersicht.....	1
Was sind Bibliotheken?	1
2. Installieren von Bibliotheken	3
Systemvoraussetzungen	3
Versionskompatibilität der Bibliotheken	3
Installationsort	4
Installationsverfahren	4
3. Verwenden der Bibliotheken.....	5
Erstellen eines Bibliothekprojektes	5
Hinzufügen von Bibliotheksdateien zu einem Projekt.....	6
Praktischer Einsatz von Bibliotheksprojekten.....	6
Identifizieren der Startkomponente	7
Testen von Parsern	7
Testen von Serializern.....	9
Verstehen des IntelliScript.....	10
Bearbeiten von Bibliothekskomponenten	10
Aktualisieren von Bibliotheksversionen	12
Bereitstellen von Projekten als Conversion-Agent-Dienst.....	12
4. Beschreibung der Bibliotheken	14
AL3-Bibliothek	14
Struktur von AL3-Nachrichten	14
XML einer AL3-Nachricht	15
EDI-Bibliotheken	15
EID-X12-Validierung.....	16
Struktur von EDI-Nachrichten.....	17
XML einer EDI-Nachricht.....	18
HIPAA-Bibliothek	18
Struktur von HIPAA-Nachrichten.....	19
XML einer HIPAA-Nachricht.....	20
HL7-Bibliothek	20
Struktur einer HL7-Nachricht	21
XML einer HL7-Nachricht.....	21

SWIFT	22
Struktur von SWIFT-Nachrichten	22
XML einer SWIFT-Nachricht	23
Index.....	25

1 Übersicht

Conversion-Agent-Bibliotheken enthalten einsatzbereite Umwandlungskomponenten, die auf verschiedene Branchenstandards wie etwa HL7, EDI und AL3 zugeschnitten sind.

Jede Bibliothek enthält eine Vielzahl vordefinierter Komponenten (Parser, Serializer und XSD-Schemata) für die Verwendung mit Nachrichten des jeweiligen Branchenstandards und bestimmter Anwendungen. Diese Komponenten erleichtern Ihnen die Datenumwandlung von Branchenstandards in XML und umgekehrt.

Die Bibliothekskomponenten können wertvolle Hilfsmittel sein, die Ihnen die Arbeit mit Conversion Agent erleichtern. Sie bieten die folgenden Vorteile:

- Sie sparen Zeit, da Sie die Komponenten nicht mehr selbst entwerfen und implementieren müssen, sondern stattdessen einfach vordefinierte Bibliothekskomponenten verwenden können.
- Das einheitliche Implementierungsverfahren erleichtert die anwendungsübergreifende Konsistenz.
- Der robuste Aufbau unterstützt alle Optionen und syntaktischen Feinheiten der einzelnen Branchenstandards.

Dieses Buch beschreibt die Verwendung der Bibliotheken und ihrer Komponenten.

Was sind Bibliotheken?

Eine Bibliothek ist ein Ordner, der eine Gruppe vollständig konfigurierter Conversion-Agent-Komponenten enthält. In der Regel umfasst eine Bibliothek Parser, Serializer und XSD-Schemata. Wenn Sie ein neues Projekt anlegen, können Sie eine Bibliothekskomponente (zum Beispiel einen Parser) importieren, statt sie selbst von Grund auf neu zu schreiben. Außerdem können Sie die Bibliothekskomponenten ganz an Ihren Bedarf anpassen.

Jede Bibliothek ist auf einen bestimmten Branchenstandard zugeschnitten. Die Bibliothek HL7 etwa enthält Projektkomponenten für die einzelnen Nachrichtenarten und Datenstrukturen des in medizinischen Informationssystemen eingesetzten Kommunikationsstandards HL7.

Die Unterordner des Bibliotheksordners enthalten Conversion-Agent-Projekte, die die Parser, Serializer usw. definieren. Die Verteilung auf die einzelnen

Unterordner erfolgt anhand der Projektkategorie. Die Bibliothek HL7 enthält zum Beispiel die folgenden Ordner:

Admission Discharge and Transfer Messages
Ancillary Data Reporting
Control Section

und so weiter. Der Ordner Admission Discharge and Transfer Messages wiederum enthält Unterordner für die verschiedenen Arten von HL7-Nachrichten, also zum Beispiel:

ADT_A01_Admit_a_Patient
ADT_A02_Transfer_a_Patient

Die Unterordner auf der niedrigsten Ebene enthalten in der Regel zwei CMW-Projektdateien von Conversion Agent (jeweils eine für den Parser und den Serializer) sowie die zugehörigen TGP-Skriptdateien und XSD-Schemata. Versuchen Sie nicht, die Komponenten direkt von den Bibliotheksordnern aus auszuführen. Erstellen Sie stattdessen mit Conversion Agent Studio in Ihrem Workspace Projekte und importieren Sie die Bibliothekskomponenten in diese Projekte.

2 Installieren von Bibliotheken

Die Bibliotheken werden mit dem Setup von Conversion Agent bereitgestellt. Dieser Abschnitt beschreibt die Installation.



Eventuell sind weitere Bibliotheken oder Bibliothekskomponenten vorhanden, die einzeln installiert werden können. Weitere Informationen dazu erhalten Sie beim SAP-Support.

Systemvoraussetzungen

Die Bibliotheken benötigen den folgenden Festplattenspeicherplatz:

Bibliothek	Ungefäher Speicherbedarf (MB)
AL3	9
EDI-AAR	8
EDIFACT	1780
EDI-UCS_and_WINS	7
EDI-VCS	11
EDI-X12	84
HIPAA	4
HL7	96
SWIFT	61

Versionskompatibilität der Bibliotheken

Wir empfehlen die Verwendung der mit Ihrer aktuellen Version von Conversion Agent bereitgestellten Bibliotheken. Dadurch ist sichergestellt, dass Ihnen die

neuesten Bibliothekskomponenten zur Verfügung stehen, die auf der aktuellen Version der Conversion Agent Engine getestet wurden.

Eventuell möchten Sie aus irgendeinem Grund eine ältere Bibliotheksversion verwenden. Zum Beispiel, um die Kompatibilität mit vorhandenen Anwendungen zu gewährleisten, die Komponenten einer älteren Bibliothek verwenden. Conversion Agent Studio unterstützt das Importieren von Komponenten aus älteren Bibliotheksversionen. Bei Bedarf aktualisiert das Studio den internen Code der Komponenten, um die Kompatibilität mit der aktuellen Version von Conversion Agent sicherzustellen und gleichzeitig die Datenumwandlungslogik der älteren Bibliotheksversion beizubehalten. Der Aktualisierungsprozess und seine Grenzen werden im Kapitel *Aktualisieren von Conversion-Agent-Projekten* des *Conversion-Agent-Administratorhandbuchs* besprochen.

Installationsort

Sie sollten die Bibliotheken immer in Ihrer Entwicklungsumgebung auf dem Windows-Rechner installieren, auf dem Sie auch Conversion Agent Studio ausführen. Denn dort benötigen Sie die Bibliotheken, wenn Sie Projekte erstellen, die Bibliothekskomponenten verwenden. Nach der Konfiguration können Sie das betreffende Projekt als Dienst in einer Produktionsumgebung bereitstellen, die die Conversion Agent Engine ausführt.

Auf Computern, die nur die Conversion Agent Engine ohne das Studio ausführen, brauchen Sie die Bibliotheken nicht zu installieren. Die bereitgestellten Dienste, die auf der Engine ausgeführt werden, greifen zur Laufzeit nicht auf die Bibliotheken zu, sondern lediglich auf die Kopien der Bibliothekskomponenten, die Sie in Ihre Projekte importiert haben.

Installationsverfahren

Um die Bibliotheken zu installieren, führen Sie das Setup von Conversion Agent aus. Dabei werden Sie aufgefordert, die zu installierenden Bibliotheken und Komponenten von Conversion Agent anzugeben. Eine ausführliche Anleitung dazu finden Sie im *Conversion-Agent-Administratorhandbuch*.

In der Voreinstellung werden die Bibliotheken im Unterordner `Libraries` des Installationsordners von Conversion Agent installiert:

`c:\Programme\SAP\ConversionAgent\Libraries`

3 Verwenden der Bibliotheken

Sie können die in den Bibliotheken bereitgestellten Komponenten in Ihren Conversion-Agent-Projekten verwenden. Dieses Kapitel beschreibt die entsprechenden Verfahren.

Erstellen eines Bibliothekprojektes

Sie können ein neues Projekt anlegen, das eine Bibliothekskomponente wie etwa einen Parser oder Serializer verwendet. Dabei gehen Sie folgendermaßen vor:

1. Wählen Sie in Conversion Agent Studio „Datei > Neu > Projekt“ aus. Daraufhin wird der Assistent „Neues Projekt“ geöffnet.
2. Wählen Sie im linken Bereich des Assistenten „Conversion Agent“ aus. Im rechten Bereich markieren Sie „Bibliothekprojekt“.
3. Auf der nächsten Seite des Assistenten geben Sie einen Projektnamen und den Speicherort ein. Der Standardspeicherort ist der Workspace-Ordner von Conversion Agent Studio.
4. Erweitern Sie auf der Seite „Nachricht auswählen“ den Baum so, dass der Name und die Version der Bibliothek sowie die gewünschte Nachricht oder Datenstruktur angezeigt werden.

Wenn Ihr Projekt zum Beispiel die Nachrichten ADT_A01 von HL7, Version 2.3, parst, erweitern Sie den Baum, bis Folgendes angezeigt wird:

```
HL7\2_3\Admission Discharge and Transfer Messages\  
ADT_A01_Admin_t_a_Patient
```

5. Wählen Sie die Parser- oder Serializer-Komponente aus, die Sie im Projekt verwenden möchten. Beispiel:

```
ADT_A01_Admin_t_a_Patient_Parser
```
6. Vergewissern Sie sich, dass Sie die richtige Komponente ausgewählt haben, und klicken Sie auf „Fertigstellen“.

Conversion Agent Studio importiert die Bibliothekskomponenten, also zum Beispiel die CMW-, TGP und XSD-Dateien, in ein neues Projekt. Der Conversion Agent Explorer zeigt eine Liste aller vorhandenen Projekte an.

Hinzufügen von Bibliotheksdateien zu einem Projekt

Sie können nicht nur neue Projekte erstellen, sondern die Bibliothekskomponenten auch in ein vorhandenes Conversion-Agent-Projekt einbinden. Dafür empfehlen wir das folgende Verfahren:

1. Erstellen Sie ein neues Bibliotheksprojekt, das die gewünschte Komponente enthält (siehe *Erstellen eines Bibliotheksprojektes* weiter oben).
2. Im Conversion Agent Explorer:
 - a. Erweitern Sie das vorhandene Projekt, dem Sie die Bibliothekskomponente hinzufügen möchten.
 - b. Um einen Parser oder Serializer hinzuzufügen, klicken Sie den Knoten „Scripts“ mit der rechten Maustaste an und wählen „Datei hinzufügen“ aus. Um ein XSD-Schema hinzuzufügen, klicken Sie stattdessen den Knoten „XSD“ mit der rechten Maustaste an.
 - c. Navigieren Sie zum gerade erstellten Bibliotheksprojekt, wählen Sie die hinzuzufügende TGP-Skriptdatei oder XSD-Datei aus und klicken Sie auf „Öffnen“. Conversion Agent Studio kopiert die Datei in das vorhandene Projekt.
3. Wiederholen Sie Schritt 2 für alle weiteren TGP- und XSD-Dateien, die Sie hinzufügen möchten.

Beachten Sie dabei, dass zwischen den Dateien Abhängigkeiten bestehen können. Zum Beispiel erfordert eine TGP-Datei, die einen Hauptparser definiert, in der Regel eine XSD-Datei, die das XML-Schema der Parserausgabe definiert. Außerdem benötigt sie eventuell eine TGP-Datei, die sekundäre Parser definiert.

Wenn Sie nicht sicher sind, welche TGP- und XSD-Dateien für eine bestimmte Komponente erforderlich sind, fügen Sie am besten alle hinzu. Nicht verwendete TGP- und XSD-Dateien wirken sich nicht auf Ihr Projekt aus.

4. Anschließend können Sie das neue Bibliotheksprojekt löschen, sofern Sie es nicht für andere Zwecke benötigen.

Praktischer Einsatz von Bibliotheksprojekten

Sie können Bibliotheksprojekte in Conversion Agent Studio genauso wie alle anderen Conversion-Agent-Projekte anzeigen, bearbeiten und testen. Eine Anleitung dazu finden Sie in *Erste Schritte mit Conversion Agent* und im *Benutzerhandbuch für Conversion Agent Studio*.

Der folgende Abschnitt enthält einige Tipps zu häufig ausgeführten Operationen.

Identifizieren der Startkomponente

Bibliotheksprojekte haben in aller Regel eine komplexe Struktur mit zahlreichen Komponenten auf oberster Ebene und geschachtelten IntelliScript-Ebenen. Diese Struktur ist notwendig, da die von den Komponenten verarbeiteten Branchenstandard-Nachrichten höchst komplex sein können.

Jedes Projekt enthält eine *Startkomponente*, also die Hauptkomponente, die Conversion Agent aktiviert, wenn Sie das Projekt ausführen. Welche Komponente die Startkomponente ist, finden Sie heraus, indem Sie den Befehl „Ausführen > Ausführen“ aufrufen. Das Dialogfenster „Startkomponente auswählen“ zeigt den Namen der Startkomponente.

Wenn Sie die Startkomponente aus irgendeinem Grund ändern möchten, gehen Sie folgendermaßen vor:

1. Öffnen Sie die TGP-Skriptdatei im IntelliScript-Editor.
2. Klicken Sie die gewünschte Komponente mit der rechten Maustaste an und wählen Sie „Als Startkomponente festlegen“ aus.

Testen von Parsern

Es gibt mehrere Möglichkeiten, einen Parser auf einem Quelldokument zu testen.

Möglichkeit 1: Ausführen des Parsers auf einem Beispielquelldokument

1. Öffnen Sie im IntelliScript-Editor eine TGP-Skriptdatei des Projekts.
2. Finden Sie die Startkomponente, wie es oben beschrieben wurde (siehe *Identifizieren der Startkomponente* weiter oben).
3. Suchen Sie die Definition der Startkomponente im IntelliScript. In den meisten Bibliotheksprojekten ist dies die erste Komponente in der TGP-Datei.
4. Erweitern Sie den IntelliScript-Baum und bearbeiten Sie die Eigenschaft `example_source` der Komponente. Weisen Sie ihr den Wert `Local File` zu.
5. Navigieren Sie in der Komponente `Local File` zum gewünschten Quelldokument. Um zum Beispiel den HL7-Parser für ADT_A01 zu testen, navigieren Sie zu einer Datei, die eine ADT_A01-Nachricht enthält.
6. Das Dokument wird im Beispielbereich des IntelliScript-Editors aufgeführt.
Wenn der Beispielbereich nicht sichtbar ist, wählen Sie im Menü die Option „IntelliScript > Beide“ aus. Wenn das Dokument nicht angezeigt wird, klicken Sie den Namen des Parsers im IntelliScript an und wählen „Beispielquelle öffnen“ aus.
7. Optional können Sie mit dem Befehl „IntelliScript > Beispiel markieren“ die Daten markieren, die der Parser im Dokument findet.

Wenn das Dokument der Nachrichtenstruktur des Branchenstandards entspricht, sollte der Bibliotheksparser alle Daten im Dokument finden. Der Befehl „Beispiel markieren“ sollte demnach das gesamte Dokument markieren.

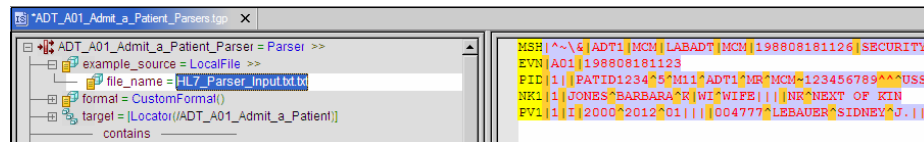
Wenn die Dokumentstruktur angepasst wurde, werden einzelne Teile eventuell nicht hervorgehoben. In diesem Fall kann es sinnvoll sein, den Parser so zu bearbeiten, dass die geänderte Struktur verarbeitet wird (siehe *Bearbeiten von Bibliothekskomponenten* weiter unten).

Hinweis: Die Farbcodierung wird im Buch *Conversion Agent Studio in Eclipse* erklärt.

8. Aktivieren Sie den Parser mit dem Befehl „Ausführen > Ausführen“.
9. Die Ereignisansicht zeigt ein Protokoll des Vorgangs an. Wie Sie das Protokoll interpretieren, wird im Kapitel *Projekte ausführen und testen im Benutzerhandbuch für Conversion Agent Studio* erklärt.
10. Um die Parserausgabe einzusehen, klicken Sie die Datei `Results/Output.xml` im Conversion Agent Explorer doppelt an.



Die Bibliotheksparser sind so konfiguriert, dass kein Beispielquelldokument erforderlich ist. Nach Abschluss des Tests können Sie die Beispielquelle entfernen. Das Beibehalten der Beispielquelle hat zur Laufzeit keinerlei Auswirkungen auf die Bibliotheksdatenumwandlung.



Zuweisen der Eigenschaft `example_source` eines Parsers

```

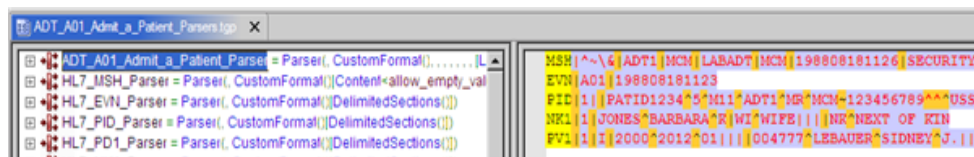
- <ADT_A01_Admit_a_Patient>
- <MSH>
  <Field_Separator>|</Field_Separator>
  <Encoding_Characters>^~\&</Encoding_Characters>
- <Sending_Application>
  <namespace_ID>ADT1</namespace_ID>
</Sending_Application>
- <Sending_Facility>
  <namespace_ID>MCM</namespace_ID>
</Sending_Facility>
- <Receiving_Application>
  <namespace_ID>LABADT</namespace_ID>
</Receiving_Application>
- <Receiving_Facility>
  <namespace_ID>MCM</namespace_ID>
</Receiving_Facility>
- <Date_Time_of_Message>
  <free_text>198808181126</free_text>
</Date_Time_of_Message>
<Security>SECURITY</Security>
- <Message_Type>
  <identifier>ADT</identifier>
  <text>A01</text>
</Message_Type>
  <Message_Control_ID>MSG00001</Message_Control_ID>
- <Processing_ID>
  <processing_ID>P</processing_ID>
</Processing_ID>

```

Ausgabe eines Parsers

Möglichkeit 2: Graphische Darstellung des Parserergebnisses

1. Öffnen Sie im IntelliScript-Editor eine TGP-Skriptdatei des Projekts.
2. Klicken Sie auf „IntelliScript > Dokument testen“ und navigieren Sie zu einem Quelldokument. Um zum Beispiel den HL7-Parser für ADT_A01 zu testen, navigieren Sie zu einer Datei, die eine ADT_A01-Nachricht enthält.
3. Das Dokument wird im Beispielbereich des IntelliScript-Editors aufgeführt. Markieren Sie mit dem Befehl „IntelliScript > Beispiel markieren“ die Anker, die der Parser im Dokument findet.



Parserstestdokument mit markierten Anker

Testen von Serializern

Um einen Serializer zu testen, führen Sie das Projekt aus. Wählen Sie auf Aufforderung das gewünschte XML-Eingabedokument aus.

Natürlich muss das XML dem im Projekt definierten XSD-Schema entsprechen. Eine einfache Möglichkeit, ein dem Schema entsprechendes XML-Dokument zu erzeugen, besteht darin, ein Bibliotheksparserprojekt zu erstellen und es auf einem Quelldokument auszuführen. Die Parserausgabe ist ein XML-Dokument, das dann als Testeingabe des Serializers dienen kann.

Verstehen des IntelliScript

Wenn Sie das IntelliScript-Skript verstehen oder Fehler beheben müssen, erleichtern Ihnen die folgenden Tipps das Verstehen der IntelliScript-Konfiguration.

Parser:

1. Öffnen Sie den Parser im IntelliScript-Editor.
2. Öffnen Sie im Beispielbereich des IntelliScript-Editors ein Beispielquelldokument und führen Sie den Befehl „IntelliScript > Beispiel markieren“ aus (siehe *Testen von Parsern* weiter oben).
3. Klicken Sie einen der markierten Anker mit der rechten Maustaste an und wählen Sie „Instanz anzeigen“ aus.

Conversion Agent Studio markiert den Teil des IntelliScript, der den Anker definiert.

4. Klicken Sie mit der rechten Maustaste einen der Anker im IntelliScript an und wählen Sie „Markierung anzeigen“ aus.

Conversion Agent Studio hebt den Anker im Testdokument hervor.

Parser oder Serializer:

1. Öffnen Sie den Parser oder Serializer im IntelliScript-Editor.
2. Führen Sie den Parser oder Serializer auf einem Quelldokument aus (siehe *Testen von Parsern* oder *Testen von Serializern*).
3. Erweitern Sie in der Ereignisansicht den Knoten „Ausführung“ des Ereignisprotokolls.
4. Klicken Sie eines der Ereignisse, zum Beispiel ein Marker-Ereignis, doppelt an.
5. Conversion Agent Studio markiert den Anker oder die Komponente, der oder die das Ereignis ausgelöst hat, im IntelliScript-Bereich und im Beispielbereich.

Bearbeiten von Bibliothekskomponenten

Wenn Sie eine Bibliothekskomponente importiert haben, können Sie sie direkt verwenden oder an Ihre Bedürfnisse anpassen. Die Bearbeitung der in ein Projekt importierten Kopie einer Komponente hat keinerlei Auswirkungen auf die ursprüngliche Komponente in der Bibliothek.

Im Folgenden werden einige Beispieloperationen beschrieben, die Sie auf Bibliothekskomponenten ausführen können.

Ändern eines Feldes von „optional“ auf „erforderlich“

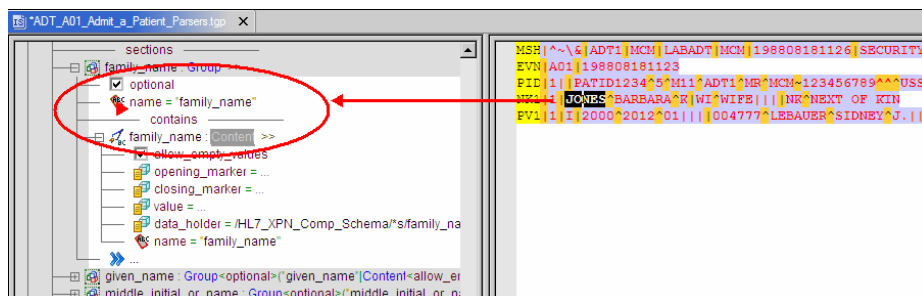
Die Ausgangssituation: Ein Kommunikationsstandard definiert ein bestimmtes Feld als optional, während es in Ihren Anwendungen unbedingt erforderlich ist.

In diesem Fall können Sie die Bibliothekskomponenten bearbeiten, indem Sie die Kennzeichnung des entsprechenden Ankers von „optional“ auf „erforderlich“ setzen. Wenn dann keine Eingabe für den Anker vorhanden ist, gibt Conversion Agent einen Fehler aus. (Dadurch wird die Verarbeitung der restlichen Eingabe nicht unbedingt angehalten. Eine Erklärung dazu finden Sie unter *Projekte ausführen und testen* im *Benutzerhandbuch für Conversion Agent Studio*.)

Umgekehrt definiert ein Standard ein Feld möglicherweise als Pflichtfeld, während es in Ihren Anwendungen nicht unbedingt vorhanden sein muss. In den Bibliothekskomponenten können Sie den entsprechenden Anker von „erforderlich“ auf „optional“ setzen. Wenn dann für den Anker keine Eingabe vorhanden ist, gibt Conversion Agent keinen Fehler aus.

So setzen Sie einen Anker von „optional“ auf „erforderlich“ oder umgekehrt:

1. Öffnen Sie die Bibliothekskomponente im IntelliScript-Editor.
2. Suchen Sie den Anker, der dem Nachrichtefeld entspricht (siehe *Verstehen des IntelliScript* weiter oben).
3. Um die erweiterten Eigenschaften der Komponente anzuzeigen, klicken Sie auf das Symbol >> neben ihrem Namen.
4. Bearbeiten Sie die Eigenschaft `optional` der Komponente.



Bearbeiten der Eigenschaft `optional` eines Ankers

Bearbeiten eines Felddelimiters

Bei einem Parser können die Eingabedatenfelder durch Delimiter wie etwa Leerstellen, Kommata oder Tabulatoren oder durch andere Hilfsmittel wie zum Beispiel absolute Abstände definiert werden. Um die Datenfelder oder die dazwischen liegenden Delimiter zu ändern, können Sie die folgenden Stellen in der Parserkonfiguration bearbeiten:

- die Eigenschaft `format` des Parsers (siehe *Formate* im *Benutzerhandbuch für Conversion Agent Studio*)

- die Anker, die auf bestimmte Positionen im Eingabedokument verweisen (siehe *Anker* im *Benutzerhandbuch für Conversion Agent Studio*)

Hinzufügen eines Segments

Einige Kommunikationsstandards, zum Beispiel HL7 oder EDI, fassen Datenfelder zu Einheiten wie etwa *Segmenten* zusammen. Möglicherweise möchten Sie eine Bibliothekskomponente so ändern, dass sie ein benutzerdefiniertes Segment unterstützt.

Dafür können Sie einen Teil des IntelliScript suchen, der ein ähnliches Segment verarbeitet. Kopieren Sie die IntelliScript-Komponenten, fügen Sie sie ein und bearbeiten Sie sie nach Bedarf. Außerdem sollten Sie das XSD-Schema so ändern, dass es Ihre neue Definition unterstützt.

Verwenden eines proprietären XML-Schemas

Eventuell müssen Sie eine Nachricht aus einem Branchenstandard in ein proprietäres XML-Schema umwandeln, das vom Bibliotheksschema abweicht. Dafür wandeln Sie die Nachricht mit einem Bibliotheksparser in ein Bibliotheksschema um. Anschließend konfigurieren Sie einen Mapper, der das XML in das proprietäre Schema umwandelt (siehe *Mapper* im *Benutzerhandbuch für Conversion Agent Studio*).

Dokumentieren Ihrer Bearbeitungen

Wir empfehlen Ihnen dringend, alle Bearbeitungen von Bibliothekskomponenten zu dokumentieren. Die Dokumentation kann sehr wichtig sein, wenn Sie die Bibliothekskomponente später aktualisieren und Ihre Bearbeitung reproduzieren müssen.

Aktualisieren von Bibliotheksversionen

Nach dem Importieren einer Bibliothekskomponente in ein Projekt besteht keine Verknüpfung zwischen dem Projekt und der Bibliothek. Wenn Sie eine neue Version der Bibliothek installieren, wird das Projekt nicht automatisch aktualisiert.

Wenn Sie das Projekt aktualisieren möchten, müssen Sie die Komponente erneut importieren. Anschließend sollten Sie das Projekt auf Unterschiede im Verhalten testen.

Bereitstellen von Projekten als Conversion-Agent-Dienst

Nach dem Testen können Sie das Projekt als Conversion-Agent-Dienst bereitstellen, der auf der Conversion Agent Engine ausgeführt wird.

Eine Anleitung dazu finden Sie im Abschnitt *Bereitstellen von Conversion-Agent-Diensten* im *Benutzerhandbuch für Conversion Agent Studio*. Weitere Informationen zur Ausführung der Dienste finden Sie im *Conversion-Agent-Entwicklerhandbuch*.

4 Beschreibung der Bibliotheken

Dieses Kapitel beschreibt die für Conversion Agent verfügbaren Bibliotheken kurz.

AL3-Bibliothek

AL3 ist ein Kommunikationsstandard, der im Versicherungswesen eingesetzt wird. AL3 wurde von der Association for Cooperative Operations Research and Development (ACORD) entwickelt, einem gemeinnützigen Versicherungsverband. Im Allgemeinen wickeln AL3-Nachrichten die Übertragung versicherungsbezogener Elemente (Policen, Forderungen usw.) ab.

AL3 beruht auf dem Standard EDI. Weitere Informationen finden Sie unter

<http://www.acord.org/standards/al3.aspx>

Struktur von AL3-Nachrichten

Das folgende Beispiel zeigt ein Segment einer typischen AL3-Nachricht, die sich auf eine Wasserfahrzeugversicherung bezieht.

```

1MHG176  IBM716ESTEDL  ?????????????????231225          AFW
50270000010406111100051          70
2004061111000512TRG212  70          3P PBOAT FMG
0001040611          040511 PCH
20040611200405112TCG135
2TAG212  00003278          P06260822
WTRP  PRODUCER
2ACI200  Stephen F. Newman, Inc.
1600 Delaware Avenue, Suite 999 Seneca NY1499916837168563387
5BIS172  B10001          P          Ester W.          Powell
????????????????????????????????00003278
????????????????????????????
9BIS168  B10001          3672 Almond drive
Hopatka NY14888  7166090166
5ISI183  B200015BISB10001          ??  ?????????? ??????????????????
5SNG117  B200015BISB10001  01DECJane W. Fieldstone
??1????????????????????????
5BPI285  F100015BISB10001  P06260822          ??????????
FISH 040511050511

```

Das positionsgestützte Format von AL3 erfordert, dass alle Daten, auch wenn sie nicht vorhanden sind, abgebildet werden. Das obige Beispiel zeigt, wie Fragezeichen als Platzhalter für fehlende Daten eingesetzt werden.

XML einer AL3-Nachricht

Das folgende Beispiel zeigt den Anfang der XML-Struktur der oben gezeigten Nachricht nach dem Parsen mit einer Bibliothekskomponente von Conversion Agent.

```
- <AL3_301_WaterCraft>
- <Message_Header_Group>
  <Header_00>1MHG176</Header_00>
  <Message_Address_Origination_01>IBM716ESTEDL</Message_Address_Origination_01>
  <Message_Address_Destination_02>????????????????</Message_Address_Destination_02>
  <Contract_Number_03>231225</Contract_Number_03>
  <Password_User_04 />
  <System_Type_Code_05>AFW</System_Type_Code_05>
  <Interface_Software_Revision_Level_06>5027</Interface_Software_Revision_Level_06>
  <Message_Sequence_Number_07>000001</Message_Sequence_Number_07>
  <Message_Transmission_Date_Time_08>0406111100051</Message_Transmission_Date_Time_08>
  <Count_Unit_Code_09 />
  <Special_Handling_10 />
  <Message_Standard_Revision_Level_11>70</Message_Standard_Revision_Level_11>
  <Transmission_Status_Flag_12 />
  <Message_Response_Code_13 />
  <Error_Code_14 />
  <Network_Reference_Number_15 />
  <Network_Reserved_for_Future_Use_16 />
  <Message_Transmission_Date_Time_17>200406111100051</Message_Transmission_Date_Time_17>
</Message_Header_Group>
```

EDI-Bibliotheken

EDI (Electronic Data Interchange) ist ein Standardformat für den Austausch von Geschäftsdaten zwischen Handelspartnern, Firmen, die geschäftlich miteinander zu tun haben und über Netzwerke wie das Internet Informationen austauschen.

Eine EDI-Nachricht enthält eine Kette von Datenelemente, die jeweils ein einmaliges Datum (Preis, Produktmodellnummer usw.) abbilden und durch Kommata voneinander abgetrennt sind. Die gesamte Zeichenkette wird als Datensegment bezeichnet. Ein oder mehr Datensegmente ergeben zusammen mit einem Header und einem Trailer ein Transaktionsset, die Übertragungseinheit von EDI (entspricht einer Nachricht). Ein Transaktionsset besteht häufig aus den Elementen, die man auch in gängigen Geschäftsformularen findet. Beispiel:

- Bestellung
- Rechnung
- Kontoauszug des Kunden

Conversion Agent bietet Bibliotheken wie die folgende, die EDI-gestützte Nachrichtenstandards unterstützen. Die größte Bibliothek ist EDIFACT, ein EDI-

Standard, der von den Vereinten Nationen gepflegt wird. Einige der anderen Bibliotheken werden für EDI-gestützte Standards bestimmter Branchen verwendet.

EDI-AAR

Eine Bibliothek, die den EDI-Standard der American Association of Railroads unterstützt. Weitere Informationen finden Sie unter

<http://www.ismrif.org/infostnd/ediguide.html>

EDIFACT

Eine Bibliothek, die den internationalen EDI-Standard der Vereinten Nationen unterstützt. Weitere Informationen finden Sie unter

<http://www.unece.org/trade/untdid/welcome.htm>

EDI-UCS_and_WINS

Eine Bibliothek, die den UCS (Uniform Communication Standard) und den WINS (Warehouse Information Network Standard) unterstützt, die in der Lebensmittelbranche und der Lagerbranche verwendet werden. Weitere Informationen finden Sie unter

http://www.uc-council.org/ean_ucc_system/stnds_and_tech/ucs.html

EDI-VICS

Eine Bibliothek, die den Standard VICS (Voluntary Interindustry Commerce Solutions) der Einzelhandelsbranche unterstützt. Weitere Informationen finden Sie unter

<http://www.vics.org/home>

EDI-X12

Eine Bibliothek, die X12 EDI unterstützt, den US-Standard für Geschäftstransaktionen. Weitere Informationen finden Sie unter

<http://www.x12.org>



Die an derer Stelle in diesem Kapitel beschriebenen Standards AL3 und HIP AA beruhen ebenfalls auf EDI. Da ihre Namen nicht mit „EDI“ beginnen, werden sie der besseren Lesbarkeit halber gesondert aufgeführt..

EID-X12-Validierung

Die Bibliothek EDI-X12 enthält zwei Versionen der Bibliotheksverzeichnisse, die jeweils eine andere Ebene der Eingabepfung unterstützen.

Bibliotheksverzeichnisse mit Validierungen

Die Bibliotheksverzeichnisse, deren Namen mit dem Suffix `_with_validation` enden, enthalten Schemata und Parser, die ihre Eingaben auf den X12 EDI-Standard hin validieren.

Validierungsfehler führen nicht dazu, dass ein Parser scheitert. Stattdessen versieht der Parser seine XML-Ausgabe mit einem Fehlercode-Attribut, das den vom Standard festgelegten Wert enthält. Wenn beispielsweise ein zwingend

vorgeschriebenes Feld fehlt, werden sowohl das Feld als auch das Segment, das es enthält, mit Fehlercodes versehen. Die Ausgabe kann wie folgt aussehen:

```
<LX segErrorCd="8">
  <R01 errorCd="6">01.1</R01>
</LX>
```

EDI-X12 Bestätigungen

Der X12 EDI-Standard sieht Bestätigungen vor (als EDI 997-Bestätigungen bezeichnet), die den Empfang und die Gültigkeit einer EDI-Transaktion bestätigen.

Wenn Sie die `_withvalidations`-Bibliothekskomponenten verwenden, können Sie eine Datenumwandlung so konfigurieren, dass sie die Bestätigungen automatisch erzeugt. Dazu kann die Umwandlung einen vordefinierten Parser mit der XML-Ausgabe der Bibliotheksumwandlung ausführen. Die Ausgabe dieses Parsers ist die Bestätigung, die auch eine Zusammenfassung der Validierungsfehler enthält.

Um weitere Informationen zu erhalten, kontaktieren Sie bitte SAP-Support.

Sonstige Bibliotheksverzeichnisse

Die anderen EDI-X12-Bibliotheksverzeichnisse, ohne das Suffix `_withvalidations`, bieten eine geringfügigere Validierung.

Zum Beispiel definieren die Schemata alle Felder als optional. Dennoch prüfen die Parser auf obligatorische Daten.

Die Parser implementieren dies mit Hilfe der Ankereigenschaft `optional`. Wenn ein Feld gemäß des X12 EDI-Standards zwingend vorgeschrieben ist, ist die Eigenschaft `optional` des entsprechenden Ankers deaktiviert. Wenn ein zwingend vorgeschriebenes Feld fehlt, führt dies dazu, dass der Parser scheitert und keine XML-Ausgabe erzeugt. Der Fehler wird im Ereignisprotokoll angegeben.

Struktur von EDI-Nachrichten

Das folgende Beispiel zeigt eine Transaktion mit Details zum Behälterinhalt, also eine typische X12-EDI-Transaktion. Die Struktur umfasst Segmente und Felder auf verschiedenen Schachtelungsebenen. Der besseren Lesbarkeit wegen wurden Zeilenumbrüche eingefügt.

```
ST*109*0001~B4*15*0*2*19960515*1424*Statu*Equi *Equi pment*D*XXXX*
Location Identifier*A*3~N9*01*Reference Identification*
Free-form Description*19960515*1424*01*01^Reference Identification^01^
Reference Identification^01^Reference Identification-Q2*X*XX*19960515*
19960515*19960515*10045*21392*A*Flight/Voy*01*Reference Identification*
B*Vessel Name*10691*B*E~V9*AAD*Event*19960515*1424*City Name*XX*XX*001*
XXXXXX*25946*Tr*Free-Form Message*01*25878*XXXXXX*717*437*272*2457*
12935~R4*1*A*Location Identifier*Port Name*XX*Terminal Name*Pier*XX~DTM*
001*19960515*1424*01*CC*Date Time Period~V9*AAD*Event*19960515*1424*
```



*City Name*XX*XX*001*XXXXXX*4685*Tr*Free-Form Message*01*13647*XXXXXX*813**

*605*52*20035*12104~N9*01*Reference Identification*Free-form Description*
19960515*1424*01*01^Reference Identification^01^Reference Identification^
01^Reference Identification-SG*2*001*XX*19960515*1424*01-SE*11*0001~*

Die Nachrichten einiger anderer EDI-gestützter Standards sehen zwar anders aus, alle weisen jedoch dieselbe Struktur mit Segment-Feld-Delimiter auf.

XML einer EDI-Nachricht

Das folgende Beispiel zeigt den Anfang der XML-Struktur der oben gezeigten Nachricht nach dem Parsen mit einer Bibliothekskomponente von Conversion Agent.

```
- <Vessel_Content_Details_109>
- <Interchange_envelope_Loop>
  - <Functional_Loop>
    - <Transaction_Set_Header_1>
      - <Transaction_Set_Header>
        <Transaction_Set_Identifier_Code_01>109</Transaction_Set_Identifier_Code_01>
        <Transaction_Set_Control_Number_02>0001</Transaction_Set_Control_Number_02>
      </Transaction_Set_Header>
    </Transaction_Set_Header_1>
  - <Beginning_Segment_for_Inquiry_or_Reply_2>
    - <Beginning_Segment_for_Inquiry_or_Reply>
      <Special_Handling_Code_01>15</Special_Handling_Code_01>
      <Inquiry_Request_Number_02>0</Inquiry_Request_Number_02>
      <Shipment_Status_Code_03>2</Shipment_Status_Code_03>
      <Date_04>19960515</Date_04>
      <Time_05>1424</Time_05>
      <Status_Location_06>Statu</Status_Location_06>
      <Equipment_Initial_07>Equi</Equipment_Initial_07>
      <Equipment_Number_08>Equipment</Equipment_Number_08>
      <Equipment_Status_Code_09>D</Equipment_Status_Code_09>
      <Equipment_Type_10>XXXX</Equipment_Type_10>
      <Location_Identifier_11>Location Identifier</Location_Identifier_11>
      <Location_Qualifier_12>A</Location_Qualifier_12>
      <Equipment_Number_Check_Digit_13>3</Equipment_Number_Check_Digit_13>
    </Beginning_Segment_for_Inquiry_or_Reply>
  </Beginning_Segment_for_Inquiry_or_Reply_2>
```

HIPAA-Bibliothek

HIPAA ist ein Kommunikationsstandard für Verwaltungs- und Finanztransaktionen im Gesundheitswesen. Traditionell gab es im Gesundheitswesen Hunderte von Nachrichtenformaten, die HIPAA durch einen einheitlichen Standard ersetzt hat.

HIPAA beruht auf dem Standard X12 EDI. Weitere Informationen finden Sie unter

<http://aspe.hhs.gov/admnsimp/faqtx.htm>

<http://www.hipaa.org>

Struktur von HIPAA-Nachrichten

Das folgende Beispiel zeigt die Transaktion „Leistungsantrag- und -pflege“, eine typische HIPAA-Transaktion. Sie wird vom Geldgeber der Versicherung dafür verwendet, den Beitragszahlern erste Informationen zum Leistungsantrag zu geben und die Informationen nach Bedarf zu aktualisieren.

```
ISA*00**00**ZZ*EDI0021*ZZ*54828*032403*1253*U*00405*000000905*1*T*:
GS*BE*EDI0021*54828*20030324*1253*5*X*004050X125~
ST*834*26608~
BGN*00*26608*20021226*1200*ES*2~
N1*P5*UES CORP*FI*111222333~
N1*IN*UNITED HEALTHCARE*FI*777888999~
INS*Y*18*021*28*A***FT~
REF*OF*851027461~
REF*1L*199301~
DTP*356*D8*20021219~
NM1*IL*1*CANOE*NANCY*A*MRS. *I*99*99999999~
PER*IP**HP*4104038790*WP*4104035500*EX*5684~
N3*9999 ROSE VILLAGE*FIRST FLOOR~
N4*FLOWERS*ND*999995*CY*WHEELING~
DMG*D8*19740801*F*M~
HD*021**PPO*PLAN COVERAGE DESCRIPTION*ESP~
DTP*348*D8*20021219~
SE*16*26608~
ST*834*26608~
BGN*00*26608*20021226*1200*ES*2~
N1*P5*UES CORP*FI*111222333~
N1*IN*UNITED HEALTHCARE*FI*777888999~
INS*N*01*021*28*A***FT~
REF*OF*948047150~
REF*1L*199301~
```

HIPAA-Meldungen bestehen aus mehreren Segmenten, die jeweils mit einer Kennung beginnen. Die folgende Tabelle nennt einige der in dieser Nachricht enthaltenen Kennungen:

Kennung	Beschreibung
ISA	Austauschkontrollheader
GS	Headersegment der Funktionsgruppe
ST	Header-Segment
BGN	Anfangssegment
N1	Name des Geldgebers
SE	Trailer-Segment

XML einer HIPAA-Nachricht

Das folgende Beispiel zeigt den Anfang der XML-Struktur der oben gezeigten Nachricht nach dem Parsen mit einer Bibliothekskomponente von Conversion Agent.

```
- <Benefit_Enrollment_and_Maintenance>
- <Interchange_Envelope_Header>
  - <Interchange_Control_Header>
    <Authorization_Information_Qualifier_ISA01>00</Authorization_Information_Qualifier_ISA01>
    <Authorization_Information_ISA02 />
    <Security_Information_Qualifier_ISA03>00</Security_Information_Qualifier_ISA03>
    <Security_Information_ISA04 />
    <Interchange_ID_Qualifier_ISA05>ZZ</Interchange_ID_Qualifier_ISA05>
    <Interchange_Sender_ID_ISA06>EDI0021</Interchange_Sender_ID_ISA06>
    <Interchange_ID_Qualifier_ISA07>ZZ</Interchange_ID_Qualifier_ISA07>
    <Interchange_Receiver_ID_ISA08>54828</Interchange_Receiver_ID_ISA08>
    <Interchange_Date_ISA09>032403</Interchange_Date_ISA09>
    <Interchange_Time_ISA10>1253</Interchange_Time_ISA10>
    <Interchange_Control_Standards_Identifier_ISA11>U</Interchange_Control_Standards_Identifier_ISA11>
    <Interchange_Control_Version_Number_ISA12>00405</Interchange_Control_Version_Number_ISA12>
    <Interchange_Control_Number_ISA13>00000905</Interchange_Control_Number_ISA13>
    <Acknowledgment_Requested_ISA14>1</Acknowledgment_Requested_ISA14>
    <Usage_Indicator_ISA15>T</Usage_Indicator_ISA15>
    <Element_ISA16>:</Element_ISA16>
  </Interchange_Control_Header>
</Interchange_Envelope_Header>
```

HL7-Bibliothek

HL7 ist ein Kommunikationsstandard für das Gesundheitswesen. Er wird weltweit in Krankenhäusern und medizinischen Informationssystemen eingesetzt.

Conversion Agent unterstützt derzeit HL7, Version 2.3.

Eine kurze Einführung in den Standard HL7 finden Sie im Kapitel *Definieren eines HL7-Parsers* des Handbuchs *Erste Schritte mit Conversion Agent*. Ausführliche Informationen finden Sie auf der Website zu Health Level 7 unter <http://www.hl7.org>.

Struktur einer HL7-Nachricht

Das folgende Beispiel zeigt eine typische HL7-Nachricht, die Aufnahme eines Patienten. Der besseren Lesbarkeit halber wurden Zeilenumbrüche und Leerzeilen eingefügt.

```
MSH|^-\&|ADT1|MCM|FINGER|MCM|198808181126|SECURITY|ADT^A01|MSG00001|P|
2.3.1
```

```
EVN|A01|198808181123
```

```
PID|1||PATID1234^5^M11^ADT1^MR^MCM-123456789^^^USSA^SS| |
SMITH^WILLIAM^A^I||19610615|M||C|1200 N ELM STREET^^JERUSALEM^TN^99999?
1020|GL|(999)999?1212|(999)999?3333||S||PATID12345001^2^M10^ADT1^AN^A|
123456789|987654^NC
```

```
NK1|1|SMITH^OREGANO^K|WIFE|||NK^NEXT OF KIN
```

```
PV1|1||2000^2012^01||004777^CASTRO^FRANK^J.||SUR||ADM|A0
```

Die Nachricht besteht aus Segmenten, die durch Zeilenschaltungen voneinander getrennt sind. Jedes Segment ist mit einer dreistelligen Bezeichnung wie etwa MSH (Nachrichtenheader) oder PID (Patientennummer) beschriftet. Jedes Segment enthält eine vordefinierte Hierarchie von Feldern und Unterfeldern, die durch die Zeichen direkt nach der Beschriftung MSH begrenzt werden (|^-\&).

So folgt zum Beispiel der Patient name (Smith^William^A) nach der Beschriftung PID und fünf Delimitern |. Nach- und Vorname (Smith und William) sind durch einen Delimiter ^ voneinander getrennt.

Der Nachrichtentyp wird durch in Feld im Segment MSH angegeben. Im obigen Beispiel ist der Nachrichtentyp AD, der Untertyp A01; es handelt sich also um die Meldung Admit a Patient.

XML einer HL7-Nachricht

Das folgende Beispiel zeigt den Anfang der XML-Struktur der oben gezeigten Nachricht nach dem Parsen mit einer Bibliothekskomponente von Conversion Agent.

```

- <ADT_A01_Admit_a_Patient>
- <MSH>
  <Field_Separator>|</Field_Separator>
  <Encoding_Characters>^~\&</Encoding_Characters>
- <Sending_Application>
  <namespace_ID>ADT1</namespace_ID>
</Sending_Application>
- <Sending_Facility>
  <namespace_ID>MCM</namespace_ID>
</Sending_Facility>
- <Receiving_Application>
  <namespace_ID>LABADT</namespace_ID>
</Receiving_Application>
- <Receiving_Facility>
  <namespace_ID>MCM</namespace_ID>
</Receiving_Facility>
- <Date_Time_of_Message>
  <free_text>198808181126</free_text>
</Date_Time_of_Message>
<Security>SECURITY</Security>
- <Message_Type>
  <identifier>ADT</identifier>
  <text>A01</text>
</Message_Type>
<Message_Control_ID>MSG00001</Message_Control_ID>
- <Processing_ID>
  <processing_ID>P</processing_ID>
</Processing_ID>

```

SWIFT

SWIFT ist ein Kommunikationsstandard für die Finanzbranche.

Einige Beispiele für SWIFT-Nachrichten:

- Überweisungsantrag
- Antrag auf Stornierung eines Schecks
- Lastschrift und Antrag auf Bankeinzug
- mehrfache Interbank-Überweisung

Weitere Informationen finden Sie unter

<http://www.swift.com/>

Struktur von SWIFT-Nachrichten

Das folgende Beispiel zeigt eine Überweisungsantrag-Nachricht, eine typische SWIFT-Transaktion.

```
{1: F01DEUTDEFFEDI X0317000005}{2: I 101ABCDEFGHI JKAU1020}{4:  
: 20: Sender Ref Dom01  
: 21R: Customer Ref  
: 28D: 1/1  
: 50L: Instructing Party  
: 50H: /00190020974010046074  
DB SPAIN  
: 30: 030120  
: 21: REFF21-1  
: 32B: EUR1,00  
: 57D: /ES999999999  
Madrid  
: 59: /64930056505  
DB Spain  
Barcelona  
: 70: First payment11  
: 71A: OUR  
: 21: REFF21-2  
: 32B: EUR2,00  
: 57D: /E999999999  
Madrid  
: 59: /64930056506  
DB Spain  
Barcelona  
: 70: Second payment2  
: 71A: OUR
```

XML einer SWIFT-Nachricht

Das folgende Beispiel zeigt den Anfang der XML-Struktur der oben gezeigten Nachricht nach dem Parsen mit einer Bibliothekskomponente von Conversion Agent.

```
- <MT101>
- <BasicHeaderBlock>
  <ApplicationIdentifier>F</ApplicationIdentifier>
  <ServiceIdentifier>01</ServiceIdentifier>
- <LTAddress>
  <BankCode>DEUT</BankCode>
  <CountryCode>DE</CountryCode>
  <LocationCode>FF</LocationCode>
  <LogicalTerminalCode>E</LogicalTerminalCode>
  <BranchCode>DIX</BranchCode>
</LTAddress>
  <SessionNumber>0317</SessionNumber>
  <SequenceNumber>000005</SequenceNumber>
</BasicHeaderBlock>
- <ApplicationHeaderInputBlock>
  <Indicator>I</Indicator>
  <MessageType>101</MessageType>
- <DestinationAddress>
  <BankCode>ABCD</BankCode>
  <CountryCode>EF</CountryCode>
  <LocationCode>GH</LocationCode>
  <LogicalTerminalCode>I</LogicalTerminalCode>
  <BranchCode>JKA</BranchCode>
</DestinationAddress>
  <MessagePriority>U</MessagePriority>
  <DeliveryMonitoring>1</DeliveryMonitoring>
  <ObsolescencePeriod>020</ObsolescencePeriod>
</ApplicationHeaderInputBlock>
```

Index

A

- AAR
 - Bibliothek, 16
- AL3
 - Bibliothek, 14

B

- Bibliothek
 - Conversion Agent, 1
 - Installation, 3
 - Komponenten bearbeiten, 10
 - neue Version, 12
 - Projekten Komponenten hinzufügen, 6
 - Systemvoraussetzungen, 3
 - Versionskompatibilität, 4
- Bibliotheksprojekt
 - erstellen, 5

C

- Conversion-Agent-Dienst
 - Bibliotheksprojekte, 12

D

- Delimiter
 - in Bibliothekskomponenten, 11

E

- EDI
 - Bibliotheken, 15
- EDI-AAR
 - Bibliothek, 16
- EDIFACT
 - Bibliothek, 16
- EDI-JCS_and_WINS
 - Bibliothek, 16
- EDI-VICS
 - Bibliothek, 16
- EDI-X12
 - Bibliothek, 16

- Validierung, 16

H

- HL7
 - Bibliothek, 20

O

- optionales Feld
 - in Bibliothekskomponente, 11

P

- Parser
 - testen, 7

S

- Segment
 - in Bibliothekskomponenten, 12
- Serializer
 - testen, 9
- SWIFT
 - Bibliothek, 22
 - Systemvoraussetzungen
 - Bibliotheken, 3

U

- UCS
 - Bibliothek, 16

V

- VICS
 - Bibliothek, 16

W

- WINS
 - Bibliothek, 16

X

- X12 EDI
 - Bibliothek, 16