

Technische Referenz



für

XML mit Local Server

XML Version 2.0

Stand: 05.12.2002

Zu diesem Dokument

Diese Technische Referenz beschreibt die verschiedenen XML-Aufrufe, die Bedeutung der Elemente innerhalb der XML-Aufrufe und die Verwendung dieser Elemente. Die technischen Grundlagen der XML-Schnittstelle werden in der allgemeinen XML-Dokumentation von Pago beschrieben.

Diese Technische Referenz ist folgendermaßen aufgebaut:

- Die Kapitel 1 bis 5 und 10 sind allgemeingültig.
- Kapitel 6 ist für die Anwender von Modus 0,
- Kapitel 7 ist für die Anwender von Modus 1,
- Kapitel 8 ist für die Anwender von Modus 2,
- Kapitel 9 ist für die Anwender von Modus Gui.

So können Sie sich gezielt auf die Kapitel konzentrieren, die für Ihre Belange wichtig sind.

Erläuterung von Begriffen

In der Anbieter-Kunden-Kette gibt es zwei Kategorien von Kunden. Um diese klar voneinander zu unterscheiden, verwendet Pago konsequent die Begriffe *Mandant* und *Kunde*:

- Der *Mandant* ist der direkte Kunde von Pago, der Pago-Dienstleistungen in Anspruch nimmt.
- Der *Kunde* ist der Kunde des Mandanten.

Erläuterung von Hinweisen

In diesem Dokument gibt es zwei Arten von Hinweisen:

- Hinweise, die Ihnen nützliche Informationen geben, jedoch keine hohe Dringlichkeit haben. Sie sind mit dem Pago Auge gekennzeichnet:



Beachten Sie bitte ... ?

- Hinweise, die Sie unbedingt befolgen müssen, da sonst Funktionsmängel auftreten können. Sie sind mit einem Ausrufungszeichen gekennzeichnet und kursiv geschrieben:



Beachten Sie unbedingt ...

Copyright © 2002 Pago eTransaction Services GmbH

Dieses Dokument ist Eigentum von Pago eTransaction Services GmbH. Es darf weder als Ganzes noch als Teil ohne schriftliche Genehmigung von Pago eTransaction Services veröffentlicht oder Dritten zugänglich gemacht werden. Alle Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Bildentnahme, der Funksendung, der Wiedergabe auf fotomechanischem oder ähnlichem Wege, der Speicherung und Auswertung in Datenverarbeitungsanlagen, bleiben, auch bei Verwendung von Teilen des Werkes, dem Urheber vorbehalten.

Inhaltsverzeichnis

1	Pago Multipay	9
1.1	Generelle Anforderungen an Zahlungsfunktionen.....	9
1.2	Implementierungshinweis kaufmännisch/technisch	9
1.3	Einsatz von Kreditkartenprüfnummern (CVV2, CVC2)	10
1.3.1	Position der KPN bei VISA, Europay International und American Express (AMEX)	10
1.3.2	Anforderungen an Mandanten.....	11
1.3.3	Feldlänge.....	11
1.3.4	Bedienerführung	11
2	Entscheidungshilfen für die angemessene Payment-Methode	12
2.1	Die Modi in Pago Multipay	12
2.1.1	Modus 0.....	12
2.1.2	Modus 1	13
2.1.3	Modus 2	13
2.1.4	Modus GUI.....	14
3	Methoden von Pago Multipay.....	15
3.1	Bankverbindung prüfen (MultipayCheckBankRequest)	16
3.2	Kreditkarte prüfen (MultipayCheckCreditCardRequest).....	16
3.3	Bezahlvorgang initiieren (MultipayInitPaymentRequest).....	17
3.4	Bezahlvorgang initiieren im Modus GUI (MultipayInitGui- PaymentRequest)	17
3.5	Forderungen einstellen (MultipayInsertDebitRequest)	18
3.6	Forderungen einziehen (MultipaySettleAccountRequest)	19
3.7	Eine Autorisierung zurücknehmen (MultipayReduceAmount- Request)	19
3.8	Rückmeldungen von Pago an den Mandanten.....	19
4	Einsatz von Pago XML	20
4.1	Voraussetzungen für XML	20

4.2	Legende	21
4.2.1	Beispiel.....	22
5	Pago-XML-Struktur	24
5.1	Nachrichtenaustausch	26
5.2	Header-Block	27
5.3	Request-Block	29
5.4	Response-Block	31
5.4.1	Fehlerbenachrichtigung	33
5.5	Fehlermanagement	33
6	Multipay mit Modus 0	37
6.1	Anwendungsregeln für den Einsatz von Multipay im Modus 0	37
6.2	Umgang mit Retouren und Gutschriften.....	38
6.3	Transaktionssicherheit	39
6.3.1	Aufrufwiederholung bei Verbindungsabbruch.....	39
6.3.2	Handhabung der Sequenznummer	40
6.3.3	Verhalten der Methoden bei Wiederholungsaufrufen.....	40
6.4	Anwendungsbeispiele für Multipay mit Modus 0.....	41
6.4.1	Autorisierung für einen Vorgang	42
6.4.2	Einstellen einer Forderung mit anschließendem Einzug.....	42
6.4.3	Rückmeldung mit dem Ergebnis der Autorisierung und dem veränderten Debitorenstatus.....	43
6.4.4	Einstellen mehrerer Forderungen mit anschließendem Einzug des Saldos	43
6.4.5	Meldung nach Chargeback	45
6.4.6	Gutschrift	46
6.5	Technische Schnittstellenbeschreibung	48
6.5.1	Prüfen einer Bankverbindung.....	49
6.5.2	Prüfen einer Kreditkarte	51
6.5.3	Initiieren von Zahlungen	56
6.5.4	Rückmeldung des Payment-Status	65
6.5.5	Einstellen von Forderungsbuchungen.....	68
6.5.6	Bereitstellen des aktuellen Debitorenstatus	73
6.5.7	Anstoßen des Einzugs	77

6.5.8	Reduzierung des reservierten Betrags	79
7	Multipay mit Modus 1	83
7.1	Anwendungsregeln für den Einsatz von Multipay im Modus 1	83
7.2	Umgang mit Retouren und Gutschriften	84
7.3	Transaktionssicherheit	84
7.3.1	Aufrufwiederholung bei Verbindungsabbruch	84
7.3.2	Handhabung der Sequenznummer	86
7.3.3	Verhalten der Methoden bei Wiederholungsaufrufen	86
7.4	Anwendungsbeispiele für Multipay mit Modus 1	87
7.4.1	Meldung nach Chargeback	89
7.4.2	Gutschrift	90
7.4.3	Forderungskonto ausgleichen	91
7.5	Technische Schnittstellenbeschreibung	92
7.5.1	Prüfen einer Bankverbindung	93
7.5.2	Prüfen einer Kreditkarte	95
7.5.3	Initiieren von Zahlungen	100
7.5.4	Rückmeldung des Payment-Status	108
7.5.5	Einstellen von Forderungsbuchungen	111
7.5.6	Empfangen des Debitorenstatus	116
7.5.7	Anstoßen des Einzugs	120
8	Multipay mit Modus 2	122
8.1	Anwendungsregeln für den Einsatz von Multipay im Modus 2	122
8.2	Umgang mit Retouren und Gutschriften	123
8.3	Transaktionssicherheit	124
8.3.1	Aufrufwiederholung bei Verbindungsabbruch	124
8.3.2	Handhabung der Sequenznummer	125
8.3.3	Verhalten der Methoden bei Wiederholungsaufrufen	126
8.4	Anwendungsbeispiele für Multipay mit Modus 2	126
8.4.1	Autorisierung und Zahlung sofort, reines Payment	126
8.4.2	Meldung nach Chargeback	128
8.4.3	Gutschrift	128
8.4.4	Forderungskonto ausgleichen	129

8.5	Technische Schnittstellenbeschreibung	130
8.5.1	Prüfung einer Bankverbindung	131
8.5.2	Prüfen einer Kreditkarte	134
8.5.3	Initiieren von Zahlungen	139
8.5.4	Einstellen von Gutschriften mit InsertDebit.....	147
8.5.5	Empfangen des Debitorenstatus.....	152
8.5.6	Auszahlung einer Gutschrift	156
9	Multipay mit Modus GUI	158
9.1	Anwendungsregeln für den Einsatz von Multipay im Modus GUI	158
9.2	Umgang mit Retouren und Gutschriften.....	159
9.3	Transaktionssicherheit	160
9.3.1	Aufrufwiederholung bei Verbindungsabbruch.....	160
9.3.2	Handhabung der Sequenznummer	161
9.3.3	Verhalten der Methoden bei Wiederholungsaufrufen.....	161
9.4	Pago Risk Management Services in Multipay GUI.....	162
9.4.1	Der Protokollstring aus Pago Basischeck	164
9.4.2	Beispielsergebnisse der Adressprüfung.....	165
9.4.3	Kurzbeschreibung der RMS-Typen.....	166
9.4.4	Ein- und Rückgabewerte der RMS-Typen.....	167
9.4.5	Verwendung des Scorewertes	168
9.5	Beispiele für den Einsatz von Multipay mit GUI	169
9.5.1	GUI-Payment mit Scoring und Zahlarten-Auswahl.....	169
9.5.2	GUI-Payment ohne Zahlarten-Auswahl	170
9.5.3	GUI-Payment mit oder ohne Zahlarten-Auswahl, mit SSL-Form für Zahlung mit Kreditkarte	171
9.5.4	GUI-Payment mit oder ohne Zahlarten-Auswahl, mit SSL-Form für Zahlung mit Kreditkarte und Teilnahme am 3-D-Secure-Verfahren.....	174
9.5.5	GUI-Payment mit oder ohne Zahlarten-Auswahl, Online-Überweisung mit Stackbox-Applet	177
9.5.6	GUI-Payment mit oder ohne Zahlarten-Auswahl, mit SSL-Form für Zahlung per Lastschrift (ELV)	180
9.5.7	GUI-Payment mit oder ohne Zahlarten-Auswahl, Einstellen einer Forderung, alles einziehen.....	182
9.5.8	GUI-Payment mit oder ohne Zahlarten-Auswahl, Einstellen mehrerer Forderun- gen bei mehreren Positionen, späterer Einzug	183
9.5.9	GUI-Payment mit oder ohne Zahlarten-Auswahl, Zahlungsrückläufe und Gut- schriften; Meldung nach Chargeback.....	185

9.5.10	GUI-Payment mit oder ohne Zahlarten-Auswahl, Zahlungsrückläufe und Gutschriften, Gutschrift	186
9.6	Einbindung des SSL-Form bzw. Applet in das Mandantensystem	187
9.7	Technische Schnittstellenbeschreibung	191
9.7.1	Initiieren von Zahlungen	192
9.7.2	Rückmeldung des Payment-Status.....	200
9.7.3	Einstellen von Forderungsbuchungen	204
9.7.4	Empfangen des Debitorenstatus	208
9.7.5	Anstoßen des Einzugs.....	212
9.7.6	Reduzierung des reservierten Betrags	214
10	Die wichtigsten Länderkennzeichen laut ISO 3166	218

1 Pago Multipay

Pago Multipay ist ein Transaktionssystem für die gesicherte Abwicklung von Online-Zahlungsvorgängen im Internet, kombiniert mit einer Anbindung an ein Debitorenmanagement-System zur Verwaltung offener Posten. Pago Multipay unterstützt durch die exklusive Server-Server-Kommunikation verschiedene Businessmodelle wie z. B. Silent-Commerce oder Anbindung an Call-Center-Systeme.

Pago Multipay ist zur Abwicklung der gesamten Zahlungsprozesse an die unterstützten Banken und Kreditkartengesellschaften angebunden.

1.1 Generelle Anforderungen an Zahlungsfunktionen

Je nach Geschäftsmodell benötigen Sie unterschiedliche Modelle der Zahlungsabwicklung. Auch die Entscheidung für oder gegen eine bestimmte Zahlungsart (Kreditkarte, Lastschrift, etc.) wird von Ihrem Geschäftsmodell beeinflusst.

Im stationären Handel sind vor allem die traditionellen Zahlungsarten üblich. Allerdings kennt man hier auch die Unterteilung der Zahlung in eine Autorisierung und Einlösung (authorization/capture), wie z. B. bei Autovermietungen.

Im Versandhandel sind die Anforderungen umfangreicher, da hier die Lieferung einer Dienstleistung oder Ware zeitlich versetzt zur Zahlung stattfindet und unter Umständen auf mehrere Lieferschritte verteilt ist.

Pago Multipay unterstützt diverse Zahlungsarten durch den Kunden, z. B.:

- Komplette Zahlung vor Auslieferung (z. B. bei Download-Produkten oder Online-Services)
- Komplette Zahlung nach Auslieferung mit oder ohne Autorisierung (z. B. im klassischen Versand)
- Zahlung je Teillieferung inkl. Nachautorisierung der Restbeträge

Je nach Zahlungsart gibt es Einschränkungen, da nicht in allen Fällen eine Autorisierung möglich ist (z. B. beim Lastschriftverfahren).

1.2 Implementierungshinweis kaufmännisch/technisch

Zu jeder Zahlungsart gehören bestimmte Handlungsschritte und Möglichkeiten (Autorisierung, Capture, Rücklastschrift, Chargeback, Auszahlungen, etc.) mit ihren jeweiligen Gesetzmäßigkeiten und Grenzen. Diese müssen Sie genau kennen, damit Sie die Zahlungsarten sauber in Ihre Geschäftsabläufe integrieren können.

Die grundlegenden Fragen zur monetären Abwicklung und die Regeln eines Geschäftsprozesses werden in diesem Dokument nicht erläutert. Pago stellt Ihnen jedoch gerne Gesprächspartner aus dem Pago Consulting zur Verfügung, die Sie in Detailfragen beraten und in Ihrer Prozessbildung unterstützen.

Die vorliegende Technische Referenz weist auf die technischen Regeln zwischen allen beteiligten Systemen hin und verdeutlicht sie durch Beispiele.

1.3 Einsatz von Kreditkartenprüfnummern (CVV2, CVC2)

Als zusätzliches Sicherheitsfeature bietet Pago die Möglichkeit, die Kartenprüfnummern abzufragen. Kartenprüfnummern sind ein zusätzliches Sicherheitsmerkmal, mit dessen Hilfe Betrugsfälle reduziert werden können. Die Eingabe der Kreditkartennummer und der richtigen Kartenprüfnummer wird dabei als Nachweis angesehen, dass die Kreditkarte vom rechtmäßigen Inhaber eingesetzt wird. Die Nummern werden von VISA, EUROCARD/MasterCard und AMERICAN EXPRESS auf die Kreditkarten aufgedruckt. Bei VISA werden sie mit „CVV2“ bezeichnet, bei EUROCARD/MasterCard mit „CVC2“ und bei AMERICAN EXPRESS mit „KPN“. Während der Autorisierung oder der direkten Buchung wird geprüft, ob der Kartenbenutzer die richtige Kartenprüfnummer eingegeben hat. Je nach Ergebnis der Prüfung und der Autorisierung wird die Kreditkartentransaktion entweder fortgeführt oder abgebrochen.

Wenn die Kreditkartenprüfnummer des Kunden abgefragt werden soll, lässt sich dies mit Hilfe des Parameters **CVV_Mandatory** einrichten. Wenn der Parameter **CVV-Mandatory** gesetzt ist, wird eine Transaktion ohne Kartenprüfnummer in jedem Fall abgelehnt. Wenn der Parameter nicht gesetzt ist, wird die Kartenprüfnummer in den Fällen geprüft, in denen sie übermittelt wird.

1.3.1 Position der KPN bei VISA, Europay International und American Express (AMEX)

VISA nennt die KPN „CVV2“ (Card Verification Value). Sie ist Teil einer Nummer auf dem Unterschriftenfeld der Kreditkarte. Die letzten drei Ziffern dieser Nummer sind der „CVV2“.

Europay International verwendet die Bezeichnung „CVC2“ (Card Validation Code). Er besteht ebenfalls aus drei Ziffern und befindet sich an der gleichen Stelle wie der CVV2 von VISA.

AMEX bezeichnet die KPN als „Four Digit Batch Code“. Die KPN besteht aus vier Ziffern und befindet sich auf der Vorderseite der Kreditkarte, rechts oberhalb der Kartennummer.

Nicht betrachtet werden die Kartenbrands Diners und JCB. Diners wird in den Bereichen E-Commerce und MOTO nur in sehr geringem Umfang eingesetzt; JCB hat in Deutschland noch keine Karten herausgegeben.



Die Kundenkarten „Galeria“ und „Douglas“ sind keine Kreditkarten und enthalten keine KPN.

1.3.2 Anforderungen an Mandanten

Mandanten müssen in der Lage sein, neben den bekannten Kreditkartendaten (Kartenbrand, Name des Karteninhabers, Kartennummer und Verfalldatum) auch den CVV2/CVC2 des Kunden aufzunehmen. Dazu muss ein Feld existieren, in das der Kunde seine KPN eingibt. Anschließend wird die KPN zusammen mit den übrigen Kreditkartendaten an Pago geleitet. Die KPN darf *nicht* zur späteren Verwendung (z. B. in den Stammdaten) gespeichert werden.

1.3.3 Feldlänge

Das Feld für die Eingabe der KPN muss eine vierstellige Zahl aufnehmen können. Zwar stammen 95 % aller im Internet eingesetzten Kreditkarten von VISA und EUROCARD/MasterCard, deren KPN nur dreistellig sind. Dennoch werden immer mehr Kreditkarten von AMEX eingesetzt, die über eine vierstellige KPN verfügen.

1.3.4 Bedienerführung

Den meisten Kunden ist im Gegensatz zur Kreditkartennummer die KPN nicht bekannt. Aus diesem Grund sollte der Mandant einen Text für die Bedienerführung mitliefern, der die Position der KPN auf der jeweiligen Karte beschreibt.

Der Text lautet in Deutsch:

Bitte geben Sie hier die Kartenprüfnummer ein. Sie finden die Kartenprüfnummer an folgender Stelle Ihrer Kreditkarte:

VISA-Kreditkarte (CVV):

Die letzten drei Ziffern auf dem Unterschriftenfeld. Das Unterschriftenfeld befindet sich auf der Rückseite Ihrer VISA-Kreditkarte.

EUROCARD/MasterCard (CVC):

Die letzten drei Ziffern auf dem Unterschriftenfeld. Das Unterschriftenfeld befindet sich auf der Rückseite Ihrer EUROCARD/MasterCard.

AMEX (American Express) (FDBC):

Die vierstellige Kartenprüfnummer befindet sich auf der Vorderseite der American-Express-Kreditkarte, rechts oberhalb der Kreditkartennummer.

Die englische Version lautet:

Please enter the card verification code number. You will find the code number on your credit card at the following places:

VISA (CVV):

The last three digits of the credit card number that is printed on the signature panel (the stripe with the holder's signature) on the back side of your card.

EUROCARD/MasterCard (CVC):

The last three digits of the credit card number that is printed on the signature panel (the stripe with the holder's signature) on the back side of your card.

AMEX (American Express) (FDBC):

The four-digit batch code that is printed on the front side of your card, just above the card number on the right.

2 Entscheidungshilfen für die angemessene Payment-Methode

Im Folgenden werden die Modi von Pago Multipay beschrieben. Die Beschreibung soll Ihnen helfen, zu entscheiden, welcher Modus für die Anwendung in Ihrem Geschäftsmodell am besten geeignet ist.

Den gewünschten Modus geben Sie anhand eines Multipay-Request an. Hierzu wählen Sie:

- für den Aufruf von Pago Multipay ohne GUI: **MultipayInitPaymentRequest**
- für den Aufruf von Pago Multipay mit GUI: **MultipayInitGuiPaymentRequest**.

Bitte schauen Sie sich die nachfolgenden Fallstudien sorgfältig an, damit Sie die erforderlichen Buchungen richtig abbilden können.

In den Abschnitten 6.4, 7.4, 8.4 und 9.5 werden Anwendungsbeispiele zu den einzelnen Modi aufgeführt.

2.1 Die Modi in Pago Multipay

Pago Multipay bietet vier verschiedene Modi. Jeder Modus wurde für ein bestimmtes Anwendungsprofil konzipiert. Bevor Sie mit der Programmierung beginnen, prüfen Sie bitte, welcher Modus für Ihren Bedarf am besten geeignet ist. Sie können auch mehrere Modi miteinander kombinieren, sollten dies jedoch in jedem Fall mit Ihrem Pago Consultant absprechen.

Die Modi steuern das Maß Ihres Einflusses auf das Forderungs-Management, die Menge an internen Teilschritten im Zahlungsprozess und die Abfolge Ihrer Datenübermittlung an Pago.

Wenn sich grundlegende Änderungen an Ihrem organisatorischen Layout ergeben (z. B. in der Zahlungs- oder Lieferabwicklung), beraten Sie sich bitte mit Ihrem Pago-Ansprechpartner darüber, ob ein Wechsel zu einem anderen Modus für Sie sinnvoll oder notwendig ist.

2.1.1 Modus 0

Dieser Modus eignet sich für Sie, wenn Sie zuerst online autorisieren und zu einem späteren Zeitpunkt die eigentliche Zahlungstransaktion durchführen wollen (z. B. wenn Sie die bestellte Ware nicht sofort zum Versand bringen können). Im Modus 0 hat das Fälligkeitsdatum eine Wirkung auf spätere Mahnungen.

Beschreibung:

- Autorisierung und späterer Einzug des Betrags.
- Alle von Pago unterstützten Zahlarten können genutzt werden.
- Alle Sätze zu Online-Zahlarten werden der Zahlungsabwicklung übergeben.



Für das Paybox-Verfahren ist der Modus 0 notwendig, denn hierbei wird zunächst eine Autorisierung durchgeführt und zu einem späteren Zeitpunkt mit der Forderung die Zahlung ausgelöst.



Zu jedem Vorgang muss mindestens eine Forderung mit **Multipay-InsertDebitRequest** eingestellt werden.

2.1.2 Modus 1

Dieser Modus eignet sich für die Zahlarten „Kreditkarte“ und „Lastschrift“ (z. B. wenn Sie die Kreditkarte des Kunden sofort belasten wollen, weil Sie Ihr Produkt sofort zur Verfügung stellen können). Im Modus 1 wird der Betrag aufgrund des **MultipayInit-PaymentRequest** sofort autorisiert und vollständig eingezogen. Das Fälligkeitsdatum hat eine Wirkung auf spätere Mahnungen.

Beschreibung:

- Durchführung der Zahlung sofort und vollständig bei Lastschrift (**ELV**) und Kreditkarten (**KKE**). Durch die nachträgliche Einstellung von Forderungen in die Buchhaltung via **MultipayInsertDebitRequest** wird der Vorgang geschlossen.



Zu jedem Vorgang muss mindestens eine Forderung mit **Multipay-InsertDebitRequest** eingestellt werden.

- Forderungsbuchungen werden nicht an die Zahlungsabwicklung übergeben, da die Zahlung schon durchgeführt wurde.

2.1.3 Modus 2

Modus 2 eignet sich ebenfalls für die Zahlarten „Kreditkarte“ und „Lastschrift“. Im Vergleich zu Modus 1 müssen Sie jedoch in diesem Modus dem Einzug keine entsprechenden Rechnungsposten gegenüberstellen (z. B. Transaktionsarten „Forderung“, „Versandkosten“, „Positives Soll), da diese von Pago aus den Daten des Einzugs generiert werden. Der Betrag wird aufgrund der bei Pago vorhandenen Daten sofort autorisiert und mit dem Einstellen der Forderung vollständig eingezogen. Die nachgelagerten buchhalterischen Systeme von Pago werden nicht genutzt (kein Einsatz von Mahnwesen o.ä.). Außerdem brauchen Sie in diesem Fall eine geringere Zahl an internen Teilschritten.

Beschreibung:

- Sofortige Zahlung, reines Payment ohne debitorische Anbindung. Mögliche Zahlarten sind Lastschrift (**ELV**) und Kreditkarten (**KKE**).

- Forderungen über den Gesamtbetrag werden im Modus 2 mit **MultipayInitPaymentRequest** generiert. **MultipayInsertDebitRequest** ist in diesem Modus nur für nachgelagerte Gutschriften nötig.
- Es dürfen keine Forderungsbuchungen auftreten, da diese automatisch durch die Pago-Systeme erzeugt werden.

2.1.4 Modus GUI

Dieser Modus ermöglicht eine Zahlungsabwicklung, bei der die Zahldaten direkt zwischen dem Kunden und Pago ausgetauscht werden (GUI = Graphical User Interface). Dies geschieht mit Hilfe eines Java-Applet, Wallet oder SSL-Form, das im Browser des Kunden gestartet wird. Auch die Art der Kunden-Authentifizierung ist durch den Kunden wählbar.

Im Modus GUI werden für die Initialisierung von Multipay andere Parameter benötigt als in den Modi 0, 1 und 2. Auch im Response werden andere Werte zurückgegeben. Deshalb wird für die Initialisierung von Multipay mit GUI ein separater Request, der **MultipayInitGuiPaymentRequest** benötigt.

Das GUI-Payment hat für Sie den Vorteil, dass die sensiblen Zahldaten der Kunden nicht bei Ihnen gespeichert werden müssen. Allerdings muss der Kunde seine Daten bei jedem Kauf neu eingeben; es ist kein „One-Click-Shopping“ möglich.

Beschreibung:

- Der Modus GUI ist eine Eigenschaft der Zahlart und kann nicht getrennt übergeben werden. Daher ist die Anzahl der verschiedenen Zahlungswege im Modus GUI erheblich größer als bei den anderen Modi. Im Anschluss an ein Scoring (eine Risikobewertung) des Kunden können Sie ihm einige der von Pago realisierten Zahlarten anbieten. Dazu gehören auch komplexere Zahlarten (z. B. SET).
- Wahlweise für die verzögerte Zahlung (= Modus 0) oder die sofortige Zahlung (= Modus 1) möglich.

Mögliche Zahlarten sind:

- Lastschrift (**ELV**; sofort oder verzögert)
- Kreditkarten (**SKK**) mit und ohne „Verified by VISA“; sofort oder verzögert)
- Paybox (sofort oder verzögert)
- Online-Überweisung (Stackbox)
- Vorkasse
- Nachnahme.

3 Methoden von Pago Multipay

Pago Multipay bietet eine Infrastruktur an Methoden, die sich an den neuesten Anforderungen des e-Commerce orientiert.

Die Autorisierung eines Betrags geschieht an der „Kasse“ des Online-Shops. Die Rückmeldung darüber wird als Funktionsergebnis geliefert, so dass die Autorisierung vor dem Versenden der Waren ausgewertet werden kann. Im nächsten Schritt kann der Betrag eingelöst werden.

Anschaulich dargestellt sind die Vorgänge in Anwendungsbeispielen zu den einzelnen Modi:

- für Modus 0 in Abschnitt 6.4 „Anwendungsbeispiele für Multipay mit Modus 0“
- für Modus 1 in Abschnitt 7.4 „Anwendungsbeispiele für Multipay mit Modus 1“
- für Modus 2 in Abschnitt 8.4 „Anwendungsbeispiele für Multipay mit Modus 2“
- für Modus GUI in Abschnitt 9.5 „Beispiele für den Einsatz von Multipay mit GUI“).

Die Funktionalität von Pago Multipay wird durch die folgenden Methoden angesprochen.

In Modus 0

MultipayCheckBankRequest	für die Plausibilitätsprüfung von Bankdaten für das elektronische Lastschriftverfahren vor der Durchführung des Payment
MultipayCheckCreditCardRequest	für die Plausibilitätsprüfung von Kreditkartendaten vor der Durchführung des Payment
MultipayInitPaymentRequest	für die Initialisierung der Zahlungsvorgänge ohne Applet oder SSL-Form
MultipayInsertDebitRequest	zum Einstellen der Forderungen und Gutschriften
MultipaySettleAccountRequest	zum Auslösen des Einzugs / der Auszahlung
MultipayReduceAmountRequest	zur Reduzierung des autorisierten Betrags.

In Modus 1

MultipayCheckBankRequest	für die Plausibilitätsprüfung von Bankdaten für das elektronische Lastschriftverfahren vor der Durchführung des Payment
MultipayCheckCreditCardRequest	für die Plausibilitätsprüfung von Kreditkartendaten vor der Durchführung des Payment
MultipayInitPaymentRequest	für die Initialisierung der Zahlungsvorgänge ohne Applet oder SSL-Form

MultipayInsertDebitRequest	zum Einstellen einer Forderung und einer Gutschrift
MultipaySettleAccountRequest	zum Auslösen von Gutschriften.

In Modus 2

MultipayCheckBankRequest	für die Plausibilitätsprüfung von Bankdaten vor der Durchführung des Payment
MultipayCheckCreditCardRequest	für die Plausibilitätsprüfung von Kreditkartendaten vor der Durchführung des Payment
MultipayInitPaymentRequest	für die Initialisierung der Zahlungsvorgänge ohne Applet oder SSL-Form
MultipayInsertDebitRequest	zum Einstellen einer Gutschrift (negative Forderung)
MultipaySettleAccountRequest	zum Auslösen einer Gutschrift (negative Forderung).

In Modus GUI

MultipayInitGuiPaymentRequest	für die Initialisierung der Zahlungsvorgänge mit SSL-Form
MultipayInsertDebitRequest	zum Einstellen der Forderungen und Gutschriften
MultipaySettleAccountRequest	zum Auslösen des Einzugs / der Auszahlung
MultipayReduceAmountRequest	zur Reduzierung des autorisierten Betrags.

3.1 Bankverbindung prüfen (MultipayCheckBankRequest)

Mit der Methode **MultipayCheckBankRequest** wird die Bankverbindung überprüft. Dies geschieht mit Hilfe der Bankleitzahl, der Kontonummer und des entsprechenden Prüfziffernverfahrens. Die Prüfung dient u. a. dazu, grobe Fehler (z. B. Zahlendreher) auszuschließen. Sie ist vor allem bei der Registrierung eines Kunden sinnvoll. Die Methode sollte dann verwendet werden, wenn die Prüfung der Bankverbindung ohne andere Payment-bezogene Methoden aus dem Paket Pago Multipay benötigt wird. Diese Prüfung wird auch in Abhängigkeit zur Zahlart mit **MultipayInitPayment** durchgeführt.

3.2 Kreditkarte prüfen (MultipayCheckCreditCardRequest)

Mit der Methode **MultipayCheckCreditCardRequest** wird die Kreditkarte überprüft. Dies geschieht mit Hilfe der Kreditkarten-Anfangsziffern, der Kreditkartengesellschaft, der Prüfziffer und des Verfalldatums der Kreditkarte. Die Prüfung dient u. a. dazu, grobe Fehler (z. B. Zahlendreher) auszuschließen. Sie ist vor allem bei der Registrierung eines

Kunden sinnvoll. Die Methode sollte dann verwendet werden, wenn die Prüfung von Kreditkarten ohne andere Payment-bezogene Methoden aus dem Paket Pago Multipay benötigt wird. Diese Prüfung wird mit jedem **MultipayInitPayment** automatisch für Kreditkarten durchgeführt.

3.3 Bezahlvorgang initiieren (MultipayInitPaymentRequest)

Mit der Methode **MultipayInitPaymentRequest** wird ein Bezahlvorgang initiiert. Je nach Zahlart werden dabei bestimmte Funktionen ausgeführt:

- Bei Kreditkarte: Online-Autorisierung mit Betragsreservierung,
- Bei Paybox: Online-Autorisierung mit Betragsreservierung,
- Bei Lastschrift: Prüfung der Bankverbindung nach dem Prüfzifferalgorithmus bzw. HWD-Abfrage (je nach Konfiguration).



Bei der Kreditkarte wird das Kreditkartenkonto im Modus 1 und 2 sofort belastet. Beim elektronischen Lastschriftverfahren wird die Buchung sofort ausgelöst.

Der Status der Autorisierung bzw. der Status der Sofortbelastung wird mittels **PaymentNotifyRequest** von Pago übermittelt.

3.4 Bezahlvorgang initiieren im Modus GUI (MultipayInitGuiPaymentRequest)

Mit der Methode **MultipayInitGuiPaymentRequest** wird ein Bezahlvorgang initiiert. Je nach Zahlart und Parameter werden dabei bestimmte Funktionen ausgeführt:

- Prüfung und Korrektur der Adresse des Kunden
- Bildung eines Scorewertes durch die Pago Risk Management Services und Definition der Zahlarten für den Kunden
- Bei Kreditkarte: Online-Autorisierung mit Betragsreservierung
- Bei Paybox: Online-Autorisierung mit Betragsreservierung
- Bei Lastschrift: Prüfung der Bankverbindung nach dem Prüfzifferalgorithmus bzw. HWD-Abfrage (je nach Konfiguration).



Kreditkarten können bei Einsatz von GUI auf sofortigen oder verzögerten Einzug voreingestellt werden.

Der Status der Zahlung wird mittels **PaymentNotifyRequest** von Pago übermittelt.

3.5 Forderungen einstellen (MultipayInsertDebitRequest)

Der Funktionsumfang der Methode **MultipayInsertDebitRequest** ist je nach Modus unterschiedlich.



Alle Funktionen stehen für Modus 0 und Modus GUI zur Verfügung. Im Modus 1 stehen nur die Zahlarten Bankeinzug und Kreditkarte zur Verfügung. Im Modus 2 werden keine Forderungen eingestellt. Hier wird **MultipayInsertDebitRequest** nur für Gutschriften verwendet.

Mit **MultipayInsertDebitRequest** können Sie je nach Lieferstatus und verwendeter Zahlart in Abhängigkeit vom zuvor aufgerufenen **MultipayInitPaymentRequest** oder **MultipayInitGuiPaymentRequest** Forderungen einstellen. Bei Online-Bezahlverfahren werden dann jeweils Teilbeträge eingezogen. Bei Rechnung (**REC**) und Nachnahme (**NN**) wartet das Debitorensystem auf den Eingang der Zahlung. Bei Vorkasse (**VOR**) wird die Forderung in der Regel erst eingestellt, nachdem der Zahlungseingang von Pago gemeldet wurde.

Mit **MultipayInsertDebitRequest** können Sie auch Retouren und Gutschriften einstellen. Schon im Forderungssatz kann der Einzug angestoßen werden. Verwenden Sie für den Einzug den Parameter **CollectNow = true**¹.

Alternativ können Sie Forderungen mit verschiedenen Buchhaltungsparametern (z. B. „Transaktionsart“, „MwSt-Satz“) einstellen und dann gemeinsam einziehen. Verwenden Sie hierzu die Methode **MultipaySettleAccountRequest**, nachdem die zugehörigen Forderungen eingestellt wurden.



Bei Ihrer Einrichtung in der Pago-Buchhaltung wird nach Ihren Wünschen einmalig festgelegt, ob grundsätzlich bei allen Forderungen

- ein MwSt.-Satz oder
- verschiedene MwSt.-Sätze oder
- ein MwSt.-Satz

eingestellt werden. Ein Wechsel zwischen diesen Einstellungen bei jeder einzelnen Forderung ist nicht möglich.

Für Paybox ist immer Modus 0 anzuwenden. Die Durchführung ist immer verzögert („deferred“). Zuerst wird der Betrag reserviert, anschließend sind Forderungsbeträge mit **MultipayInsertDebitRequest** einzustellen. Optional können Sie bei den Zahlarten Kreditkarte, Bankeinzug und Paybox zu der Forderung auch den Einzug des Betrags veranlassen. Dies stellen Sie über den Parameter **CollectNow** in **MultipayInsertDebitRequest** ein. Der Status des Debtors wird mittels **DebtorNotifyRequest** von Pago übermittelt.

1. Beispiel: Durch den Parameter „CollectNow = true“ bei der Forderung werden alle vorherigen Vorgänge verarbeitet, die ursprünglich wegen „CollectNow = false“ nicht verarbeitet wurden.

3.6 Forderungen einziehen (MultipaySettleAccountRequest)

Die Methode **MultipaySettleAccountRequest** weist Pago an, die eingestellten Forderungen zu einem bestimmten Vorgang einzuziehen bzw. Gutschriften aus dem Umsatz auszuzahlen. Dabei wird Bezug auf die Reservierung genommen, aber auch Teilbeträge können eingezogen werden.

3.7 Eine Autorisierung zurücknehmen (MultipayReduceAmountRequest)

Eine bestehende Autorisierung können Sie mit der Methode **MultipayReduceAmountRequest** teilweise oder ganz zurücknehmen, wenn z. B. der Kunde den Auftrag zurückzieht oder die Ware nicht geliefert bzw. die Dienstleistung nicht erbracht werden kann. Der Vorgang kann auf diese Weise für weitere Forderungen geschlossen werden.

3.8 Rückmeldungen von Pago an den Mandanten

Es gibt zwei Rückmeldungsinstrumente:

- Der Debitorenstatus informiert über den Status von Forderungen.
- Der Payment-Status informiert über den Status einer Zahlung nach dem GUI-Payment.

Der Debitorenstatus wird bei Pago nach einem Statuswechsel neu erstellt, und zwar höchstens zweimal täglich. Außerdem werden über den Debitorenstatus auch Rücklastschriften und Chargebacks kommuniziert. Die Rückmeldung über den Debitorenstatus erfolgt durch die Methode **DebtorNotifyRequest**, die von Pago über den LocalServer zugestellt wird.

Der Payment-Status wird bei Pago unmittelbar nach einer Zahlung mittels SSL-Form (erfolgreich oder nicht erfolgreich) erstellt. Die Rückmeldung über den Payment-Status erfolgt durch die Methode **PaymentNotifyRequest**, die von Pago über den LocalServer zugestellt wird. Der Status einer Zahlung wird verändert durch:

- Autorisierung (Betragsreservierung) im Modus 0
- Capture (späteres Einlösen eines reservierten Betrags) im Modus 0
- Sales (sofortige Belastung des initiierten Betrags) im Modus 1
- das Ergebnis der Autorisierung und des Capture für verzögerte Zahlung („deferred“) bzw. der Sales für sofortige Zahlung („immediate“) im Modus GUI.

4 Einsatz von Pago XML

Pago XML dient der direkten Kommunikation zwischen Pago und Ihrem Backoffice über den LocalServer. Die Syntax wurde speziell auf die Pago-Online- und Offline-Dienste zugeschnitten. So konnte ein übermäßiger Ballast sowohl in der Sprachdefinition als auch in den übermittelten XML-Dokumenten vermieden werden. Gleichzeitig ähnelt Pago XML von der Struktur bis zu den Elementtypen anderen XML-Sprachen aus dem B2B-Bereich, wie etwa cXML² von Ariba Inc. oder xCBL³ von CommerceOne.

Grundlage für die Nutzung der Pago-Dienste sind Transaktionen. Die Transaktionen werden eingeleitet, indem der Client ein XML-Dokument an den Pago-Server sendet. Das XML-Dokument enthält transport- bzw. dienstspezifische Daten. Sie werden von der Pago-Prozessplattform benötigt, um den Client zu identifizieren und den angeforderten Dienst auszuführen. Anschließend sendet der Pago-Server eine Antwort an den Client und informiert ihn so über das Ergebnis seiner Anfrage. Damit kann die Transaktion beendet sein, oder sie kann weitere Anfragen und Antworten nach sich ziehen. Im letzteren Fall wird mit Transaktionsnummern auf die vorausgegangenen Anfragen und Antworten Bezug genommen.

Bei folgenden Pago-Services gibt es für asynchrone Rückmeldungen Anfragen von Pago, die sich an den Client (beim Mandanten) richten:

- Pago Multipay im Anschluss an das GUI Payment
- Pago Multipay für den Debitorenstatus
- Pago Bill.

4.1 Voraussetzungen für XML

Mit dem Release 3.0 stellt Pago seine Dienste erstmalig mit XML-Schnittstelle bereit. Dadurch ist es möglich, die Produktpalette von Pago mit Standard-APIs wie DOM oder SAX zu nutzen. Entsprechende Standard-Parser sind für alle gängigen Plattformen und Programmiersprachen erhältlich.

Pago XML ist eine Implementierung des Standards XML 1.0, der vom W3C-Konsortium (World Wide Web Consortium) spezifiziert wurde.

Bei den folgenden Beschreibungen wird vorausgesetzt, dass Sie mit der XML-Spezifikation des W3C-Konsortiums vertraut sind. Ferner müssen Sie in der Lage sein, per XML mit dem Pago LocalServer zu kommunizieren. Dies schließt folgende Fähigkeiten ein:

- XML zu erzeugen
- XML per HTTP 1.0 (oder einem anderen vom Pago LocalServer unterstützten Protokoll) an den LocalServer zu senden
- XML (als Response) vom LocalServer zu empfangen und auszuwerten.

2. <http://www.cxml.org>

3. <http://www.xcbl.org>

- XML (als Response) vom LocalServer zu empfangen und auszuwerten.

Um XML-Dokumente bei Pago richtig zuordnen zu können, muss jedes Pago-XML-Dokument eine eindeutige ID und den Sendezeitpunkt enthalten. Das Client-System muss deshalb in der Lage sein, die folgenden Angaben im richtigen Format zu erzeugen:

- die servicespezifischen Daten
- eine global eindeutige ID
- den Zeitstempel.

! *Es dürfen keine IDs doppelt generiert werden, da der Pago-Server sonst einen Fehler meldet.*

Näheres zum ID- und Zeitstempelformat finden Sie in 5.2 „Header-Block“.

4.2 Legende

Elemente

Pago XML setzt sich aus Elementen zusammen. Die Zusammensetzung der Elemente wird in diesem Handbuch anhand von Diagrammen dargestellt. Die Diagramme enthalten Symbole, die im Folgenden erklärt werden.



Ein Diagramm enthält lediglich Elemente. Attribute werden nicht dargestellt und sind nur als Text in der jeweiligen Tabelle aufgeführt.

Es gibt Parent-Elemente und Child-Elemente. Die Child-Elemente sind den Parent-Elementen untergeordnet. Jedes Child-Element kann gleichzeitig Parent-Element für weitere ihm untergeordnete Child-Elemente sein.

Ein Element wird immer als Kasten dargestellt, z. B.

Optionale Elemente und Gruppen sind durch gestrichelte Linien gekennzeichnet, z. B.:



In den Tabellen sind manche Tags als Pflichtfeld bezeichnet, obwohl sie in der zugehörigen Abbildung als optionales Element dargestellt sind. In solchen Fällen wird das ursprünglich optionale Element durch den Kontext des jeweiligen Pago-Services zu einem Pflicht-Element.



Die kleinen Pfeile links unten und die kurzen parallelen Linien links oben in manchen Kästen sind hier ohne Belang und können ignoriert werden.

Einige Elemente dürfen an der jeweiligen Stelle im XML-Dokument wiederholt werden, und zwar:

- beliebig oft: „0..∞“
- ein- oder mehrmals: „1..∞“

Sequenz- und Auswahlgruppe

Als Verbindungsglied zwischen Parent- und Child-Elementen fungieren Gruppen. In den Gruppen sind die Child-Elemente zusammengefasst. Es gibt:

- Sequenzgruppen
- Auswahlgruppen

In einer Sequenzgruppe müssen alle Child-Elemente eines Parent-Elements enthalten sein, und zwar in der angegebenen Reihenfolge (von oben nach unten gelesen). Eine Ausnahme sind die optionalen Elemente (siehe unten).

Bei einer Auswahlgruppe kann einem Parent-Element immer nur ein Element der Gruppe zugeordnet werden.

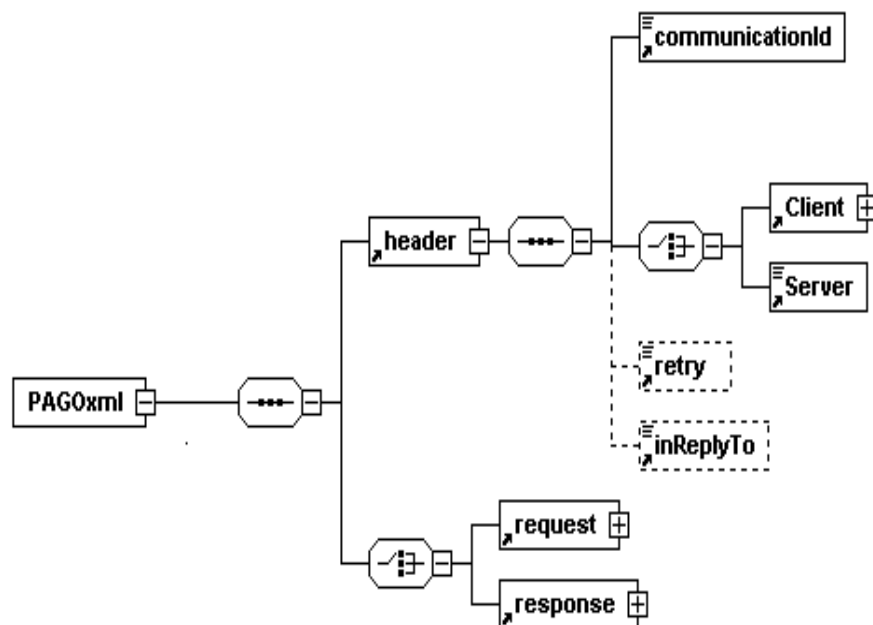
Blöcke

Ein Block setzt sich immer aus einem Parent-Element und den dazugehörigen Child-Elementen zusammen.

4.2.1 Beispiel

Das folgende Pago-XML-Element wird im unten abgebildeten Diagramm dargestellt:

```
<!ELEMENT PAGOxml (header, (request | response))>
<!ELEMENT header (communicationId, (Client | Server), retry?, inReplyTo?)>
```



5 Pago-XML-Struktur

Pago-XML-Dokumente bestehen aus

- einem XML-Vorspann
- einem Kopfteil (Element **header**)
- einem Rumpfteil (Element **request** bzw. **response**).

Der Kopf enthält transportspezifische Parameter wie IDs, Absender-Daten, Zeitstempel etc. Der Rumpf enthält servicespezifische Parameter.

Die Struktur der Pago-XML-Dokumente ist in der Document-Type-Definition **PAGOxml.dtd** festgelegt. Die DTD des Pago XML ist unter folgender URL abgelegt:

```
http://xml.pago.de/2.0/PAGOxml.dtd
```

Die folgenden Code-Ausschnitte sind aus der DTD kopiert.

Der XML-Vorspann ist unveränderlich und referenziert die Pago-XML-Sprachdefinition (DTD):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
```



Wenn die **DOCTYPE**-URL im Pago-XML-Dokument von der oben angegebenen URL abweicht, wird das Dokument vom Pago-Server zurückgewiesen.

Es folgt das Root-Element **PAGOxml** mit den beiden Attributen **version** (fixed) und **timeStamp**:

```
<PAGOxml version="2.0" timeStamp="Wed Apr 20 11:42:15 2000">
  <header>
    {hier die transportspezifische Information...}
  </header>
  <request>
    {hier die Service-spezifische Information...}
  </request>
</PAGOxml>
```

Das Pago-XML-Element enthält ein Header-Element, gefolgt entweder von einem Request-Element oder von einem Response-Element (siehe Abschnitt 5.2 „Header-Block“, Abschnitt 5.3 „Request-Block“ und Abschnitt 5.4 „Response-Block“).

Pago-XML-Tag-Namen sind case-sensitiv und bestehen aus einem Wort. Großbuchstaben werden verwendet, um mehrere Wörter zu einem Namen zusammenzusetzen. Tag-Namen, die eine direkte Entsprechung in einem Pago-Service haben, beginnen mit einem Großbuchstaben (z. B. **PaymentConnection**), die übrigen mit einem Kleinbuchstaben (z. B. **errorNotification**).

Elementwerte können alle Zeichen aus dem ISO-8859-1-Zeichensatz und dem UTF-8-Zeichensatz enthalten. Bestimmte Zeichenfolgen, die mit XML-Markup verwechselt werden können, müssen entweder in einer CDATA-Sektion übergeben werden, oder durch ihre Escape-Sequenz ersetzt werden:

Zeichen	Escape-Sequenz
&	&amp;
<	&lt;
>	&gt;

Die folgenden Beispiele sind äquivalent:

```
<LastName>Mustermann&Söhne</LastName>
```

```
<LastName><![CDATA[Mustermann&Söhne]]</LastName>
```

Näheres dazu finden Sie in der XML-Spezifikation des W3C⁴.



Das Pago-System entfernt einleitende und angehängte Leerzeichen aus dem Text. Deshalb sieht der Input des Mandantensystems anders aus als der Output des Pago-Systems. Dieser Unterschied ist in folgendem Beispiel veranschaulicht.

Input (durch das Mandantensystem erzeugt):

```
<InfoText>
    Dies ist ein
Beispiel-Text!
</InfoText>
```

Output (von Pago verarbeitet):

```
<InfoText>Dies ist ein
Beispiel-Text !</InfoText>
```

4. <http://www.w3org/TR/2000/REC-xml-20001006> - syntax

In den XML-Dokumenten, die von Pago erzeugt werden, wird immer der ISO-8859-1-Zeichensatz verwendet. UTF-8-Zeichen werden nach ISO-8859-1 konvertiert.

Rückgabewerte werden von Pago in CDATA-Sektionen gekapselt.

5.1 Nachrichtenaustausch

Pago XML unterstützt das Request-Response-Transaktionsmodell. Dieses Modell liegt auch dem HTTP-Protokoll zugrunde.

Bei der Transaktion handelt es sich um einen Nachrichtenaustausch zwischen Client und Server. Der Client sendet einen Request. Der Server nimmt ihn entgegen und schickt einen Response an den Client zurück. Diese Transaktion besteht – angewandt auf den Austausch von Pago-XML-Dokumenten – aus folgenden Schritten:

- 1) Der Client formatiert ein Pago-XML-Dokument (Request), um beim Server einen bestimmten Service ausführen zu lassen.
- 2) Der Client initiiert eine Verbindung mit dem Server auf einer vordefinierten URL, die die Adresse des Servers repräsentiert.
- 3) Der Client sendet das Pago-XML-Dokument (Request) an den Server. Anschließend wartet er und hält die Verbindung, bis die Antwort vom Server kommt.
- 4) Der Server leitet den Request an den Pago-XML-Handler weiter, der für die Verarbeitung des Request zuständig ist.
- 5) Der Handler des Servers führt den Service aus und formatiert ein Pago-XML-Antwortdokument (Response).
- 6) Der Server schickt den Response an den Client zurück.
- 7) Der Client liest die Pago-XML-Antwort und gibt sie an die Anwendung weiter, die die Anfrage ausgelöst hat.
- 8) Der Client schließt die Verbindung.

Dieser Prozess wird für weitere Request-Response-Zyklen wiederholt.

Pago liefert für die Clients den Pago LocalServer aus. Er fungiert auf Client-Seite als Kommunikationsschnittstelle und kümmert sich um Logging, Verschlüsselung und Authentifizierung. Außerdem bedient der LocalServer gleich mehrere Protokolle und vereinfacht somit den Zugriff auf die Pago-Services erheblich. Die Client-Anwendungen kommunizieren dabei lediglich mit dem LocalServer und nicht direkt mit dem Pago-Server. Weitere Details dazu enthält das Handbuch zum LocalServer.

5.2 Header-Block

Das Header-Element enthält Parameter zur Authentifizierung, zur Dokumentidentifizierung und weitere Daten, die den Transport betreffen. Es enthält keine Service-spezifischen Daten, ist also für jeden Request bzw. für jeden Response gleich. Die Definition in DTD-Schreibweise lautet:

```
<!ELEMENT header (communicationId,(Client|Server),retry?,inReplyTo?)>
```

communicationId

communicationId ist ein Dokumentschlüssel, der vom Absender der Nachricht generiert werden muss. Die **communicationId** setzt sich aus zwei Teilen zusammen.

Der erste Teil besteht aus einem Date-Time-Feld im eingeschränkten ISO-8601-Format und hat die folgende Form:

CCYY-MM-DDThh:mm:ss+hh:mm

Dabei sind:

CCYY = Jahr, vierstellig

MM = Monat, zweistellig (01 = Januar, usw.)

DD = Monatstag, zweistellig (01 bis 31)

hh = Stunde, zweistellig (00 bis 23); „am“ und „pm“ sind nicht erlaubt

mm = Minute, zweistellig (00 bis 59)

ss = Sekunde, zweistellig (00 bis 59)

+hh:mm = Zeitzonendifferenz zu UTC/GMT (+hh:mm oder -hh:mm).

Dieses Datumsformat wird vom W3C ausführlicher unter dem Titel „Date and Time Formats“⁵ beschrieben.

Der zweite Teil der **communicationId** besteht aus einer zehnstelligen numerischen Thread-ID und einer fünfstelligen Zufallszahl. Sie werden durch einen Bindestrich vom Date-Time-Feld getrennt. Abschließend folgt ein „@“-Zeichen und die IP-Adresse des verarbeitenden Client-Rechners, wahlweise im IPv4- oder IPv6-Format. Die gesamte **communicationId** hat also folgendes Format:

CCYY-MM-DDThh:mm:ss+hh:mm-ThreadID-Random@IP-Adresse

Beispiele:

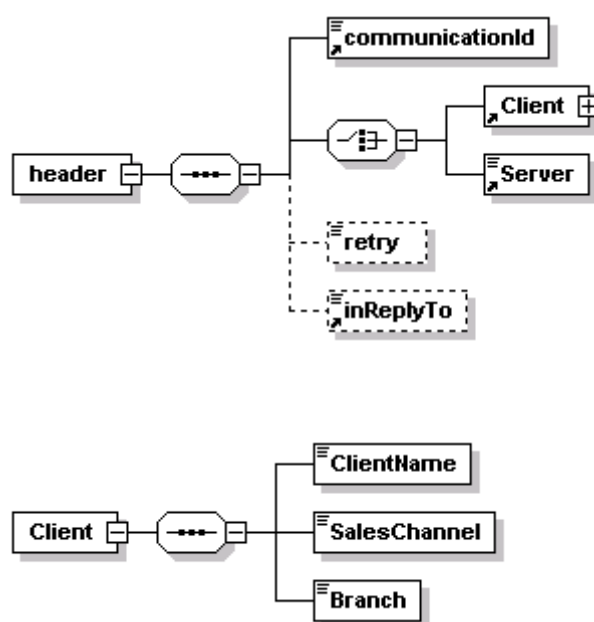
```
2000-11-16T11:38+01:00-1234567890-12345@192.168.90.2
```

5. <http://www.w3.org/TR/NOTE-datetime>

2000-04-22T10:50+00:00-137789-56478@AFFF:EEFF:07:FFEE:0304:1234:ABBA:BABA

Client / Server

Der **Client** bzw. **Server** identifiziert den Absender des Dokuments. Der **Server** spezifiziert den Namen des Pago-Servers und wird nur von Pago verwendet. Der **Client** spezifiziert Sie über die dreistufige Pago-Mandantenstruktur (**ClientName**, **SalesChannel**, **Branch**). Sie benutzen *nur* das Client-Element; das Server-Element wird *ausschließlich* von Pago verwendet.



Tag	Daten typ	Feld-länge	Pflicht-feld	Bemerkung
Client				
ClientName	an	20	Ja	Name des Mandanten Konstante in Absprache mit Pago
SalesChannel	an	20	Nein	Vertriebslinie Konstante in Absprache mit Pago
Branch	an	20	Nein	Filiale Konstante in Absprache mit Pago

retry

retry ist ein boolescher Wert, der angibt, ob es sich bei dem Dokument um eine Wiederholungsnachricht handelt. Fehlt dieser optionale Parameter, so wird **retry=false** angenommen.

Das Retry-Element zeigt der Gegenseite an, ob der Request mit identischem **<request>...</request>**-Element schon einmal übermittelt wurde. Das wiederholte Senden eines Request kann notwendig sein, wenn beim vorhergehenden Versuch ein Fehler aufgetreten ist. Beim ersten Sendeversuch eines Request sollte das Retry-Element auf jeden Fall den Wert **false** haben. Sollte das erste Senden zu einem Fehler führen, der einen erneuten Versuch sinnvoll erscheinen lässt (s. Abschnitt 5.4.1 „Fehlerbenachrichtigung“), wird im selben Request einfach das Retry-Element auf **true** gesetzt und der Request zu einem späteren Zeitpunkt (mit derselben **communicationId**) erneut versendet.

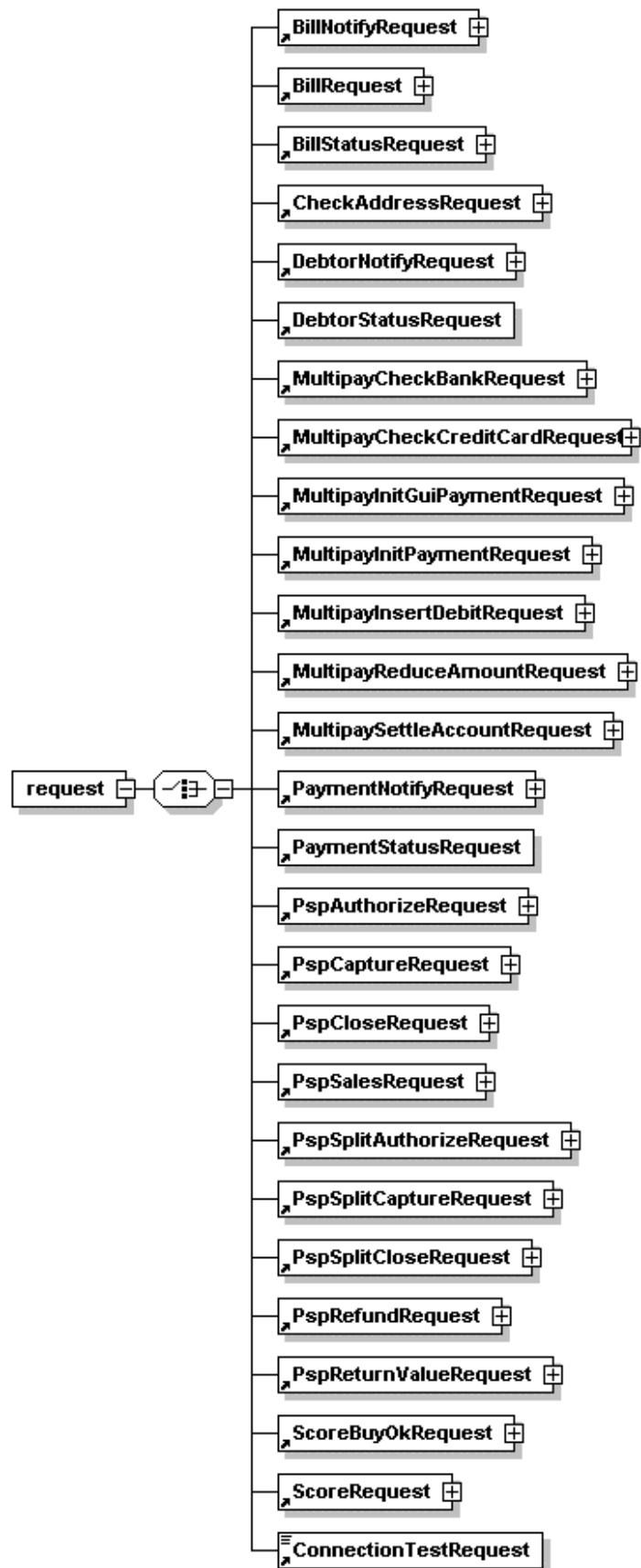
Bei Dokumenten mit gesetztem Retry-Flag und bekannter **communicationId** wird der Inhalt des Request-Blockes nicht mehr überprüft. Der Pago-Server geht aufgrund des Flags und der bekannten **communicationId** davon aus, dass sich die Daten im Request-Block *nicht* geändert haben können. Wurde der Request schon einmal verarbeitet, wird das zuvor ermittelte Ergebnis erneut versendet.

inReplyTo

inReplyTo ist ein optionales Feld, das genutzt werden kann, um einen Bezug auf ein zuvor verschicktes oder erhaltenes Pago-XML-Dokument anzugeben. Pago füllt dieses Feld in Antworten mit der **communicationId** der zugehörigen Anfrage.

5.3 Request-Block

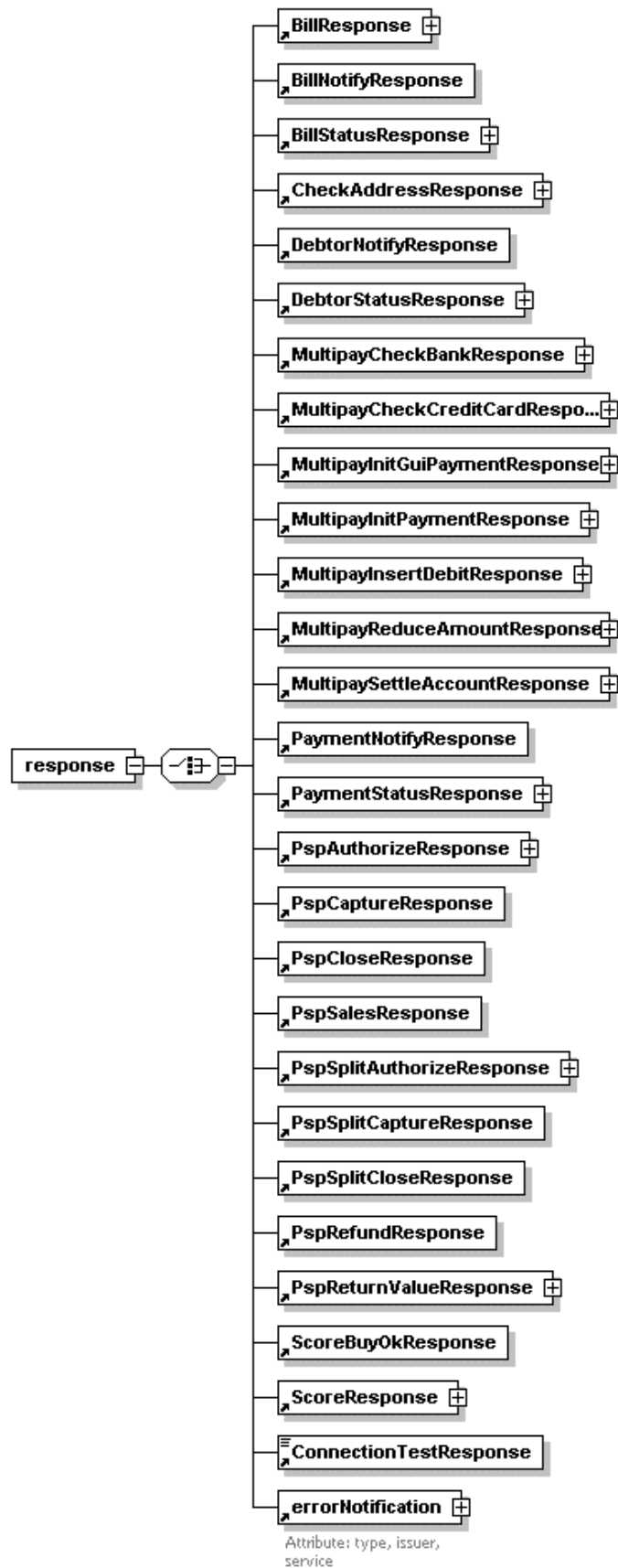
Der Request-Block folgt im Pago-XML-Wurzelement direkt auf den Header-Block. Sein einziges Element ist eines der Service-Request-Elemente, auf die später noch detailliert eingegangen wird.



Alle Request-relevanten Daten, die nicht schon im Header angegeben sind, sind im jeweiligen Service-Request-Element enthalten.

5.4 Response-Block

Der Response-Block ist ähnlich aufgebaut wie der Request-Block. Allerdings kann der Response-Block statt eines Service-Response-Elements auch ein **errorNotification**-Element enthalten, falls der Aufruf oder die Verarbeitung des vorausgegangenen Request fehlerhaft waren.



5.4.1 Fehlerbenachrichtigung

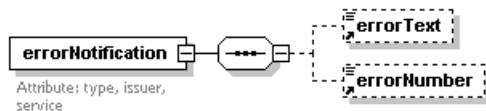
Wenn beim Transport oder bei der Verarbeitung des vorausgehenden Request ein Fehler aufgetreten ist, enthält der Response-Block eines Pago-XML-Dokuments einen **errorNotification**-Block. Der **errorNotification**-Block ist dann das einzige Element im Response-Block. Es tritt an Stelle eines Service-spezifischen Elements auf.

5.5 Fehlermanagement

Damit eine Transaktion korrekt verarbeitet werden kann, muss ein aufgetretener Fehler richtig identifiziert werden. Die XML-Anbindung der Pago-Services verfügt über ein zentrales Fehlermanagement. Es ermöglicht eine globale Fehlerbehandlung.

! *Unplausible Daten, wie eine ungültige Kreditkartennummer, sind kein Fehler. Sie werden im Response als negatives Ergebnis der Verarbeitung zurückgeliefert.*

Im Fehlerfall wird anstatt eines servicespezifischen Response-Blocks ein **errorNotification**-Block gesendet. Der **errorNotification**-Block hat folgenden Aufbau:



Tag	Daten typ	Feld-länge	Bemerkung
errorText	an	50	Beschreibung des Fehlers, Hinweise auf die Fehlerursache
errorNumber	n	6	Fehlernummer Mit Hilfe der Fehlernummer kann ein Fehler einem bestimmten Funktionsbereich zugeordnet werden.

```
<!ELEMENT errorNotification (errorText?,errorNumber?)>
<!--LIST errorNotification
      type CDATA "unknown"
      issuer CDATA #IMPLIED
      service CDATA #IMPLIED-->
<!--ELEMENT errorNumber (#PCDATA)>
<!--ELEMENT errorText (#PCDATA)>
```

Das **errorNotification**-Element enthält drei Attribute (**type**, **issuer** und **service**) und zwei optionale Elemente (**errorText** und **errorNumber**).

type

Mögliche Werte für das type-Attribut sind **recoverable**, **fatal** und **unknown**.

type=recoverable bedeutet:

Der Request konnte im Moment nicht bearbeitet werden. Mögliche Fehlerursachen sind Verbindungsprobleme oder Serverüberlastung. Der Fehler lässt sich jedoch möglicherweise beheben. Der Request sollte zu einem späteren Zeitpunkt noch einmal wiederholt werden. Als Kennzeichen, dass es sich um einen Wiederholungs-Request handelt, muss das Retry-Flag auf **true** gesetzt werden. In dem Fall muss dieselbe **communicationId** wie beim ersten Request verwendet werden.

type=fatal bedeutet:

Der Request konnte in der Form nicht bearbeitet werden. Das Request-Dokument muss geändert werden. Ein erneutes Senden des gleichen Request würde wieder denselben Fehler auslösen. Auch ein Wiederholungsrequest mit gesetztem Retry-Flag und ansonsten unveränderten Parametern ist aussichtslos. Mögliche Fehlerursachen sind ein ungültiges XML, Requests auf nichtauthorisierte Services, und ungültige IDs.

issuer

Das Element **issuer** bezeichnet den Erzeuger der Fehlermeldung und damit den Ort, an dem der Fehler aufgetreten ist. Mögliche Werte sind **PagoServer**, **LocalServer** oder **UserApplication**.

service

Das Element **service** enthält den Namen des Services, bei dem der Fehler auftrat.

Beispiel:

```
<errorNotification type="fatal" issuer="PagoServer" service="Order">
  <errorText>unknown CustomerId</errorText>
  <errorNumber>701</errorNumber>
</errorNotification>
```

Die Fehlerkreise

Fehlerkreis	Funktionsbereich, Service
0 - 15	LocalServer, siehe LocalServer Log
5000 - 99.999	XML-Backoffice,
100.000 - 100.999	Pago-Services allgemein

Fehlerkreis	Funktionsbereich, Service
101.000 - 101.999	Pago MultipayInitPayment
102.000 - 102.999	Pago MultipayInitGuiPayment
103.000 - 103.999	Pago MultipayInsertDebit
104.000 - 104.999	Pago MultipayReduceAmount
105.000 - 105.999	Pago MultipaySettleAccount
106.000 - 106.999	Pago MultipayCheckBank
107.000 - 107.999	Pago MultipayCheckCreditCard

Beispiele für Request-übergreifende Fehlercodes

Fehlercode	Grund	Abhilfe
20.001	aus DB: NO_SECURITY_FOUND	
20.002	aus DB: SECURITY_CORRUPTED	
30.001	HTTP Protokollfehler, Feld Content-Length fehlt	Feld hinzufügen
30.002	aus DB: MISSING_ACCEPT	
30.003	aus DB: MISSING_HOST	
30.004	aus DB: MISSING_FROM	
30.005	aus DB: MISSING_CONTENT	
30.006	aus DB: WRONG_SENDER	
30.007	aus DB: IN_PROCESS	Später Request wiederholen
40.001	aus DB: PARSING_FAILED	
40.003	aus DB: VALIDATION_FAILED	
40.004	aus DB: TRANSFORMATION_FAILED	
40.005	(Unter)knoten nicht gefunden oder leerer Pflichtknoten	Knoten füllen
40.006	aus DB: INVALID_FORMAT	
40.007	aus DB: INVALID_VALUE	

Fehlercode	Grund	Abhilfe
100.003	aus DB: CLIENTID_INVALID	Parameter mit Konfiguration prüfen; Parameter oder Konfiguration anpassen
100.005	aus DB: MISSING_XML_DATA	
100.010	aus DB: MISSING_DTD_ID_FIELD	
100.011	aus DB: MISSING_DTD_URL_FIELD	

6 Multipay mit Modus 0

In diesem Kapitel wird die Handhabung des Modus 0 beschrieben. Detaillierte Informationen für die Auswahl des benötigten Modus finden Sie in Kapitel 2 „Entscheidungshilfen für die angemessene Payment-Methode“.

Kennzeichnend für den Modus 0 sind folgende Merkmale:

- Der Betrag wird sofort autorisiert und später eingezogen. Alle von Pago unterstützten Zahlarten können genutzt werden.
- Im Modus 0 werden alle Buchungssätze zu Online-Zahlarten der Zahlungsabwicklung übergeben.
- Für Paybox müssen Sie den Modus 0 verwenden. (Paybox ist eine verzögerte Zahlart („deferred“), d. h. im ersten Schritt wird eine Autorisierung durchgeführt, und zu einem späteren Zeitpunkt mit der Forderung durch **CollectNow = true** oder **MultipaySettleAccountRequest** die Zahlung ausgelöst.)
- Auch Kreditkartentransaktionen müssen als verzögerte Zahlart und als Netzbetrieb über den Modus 0 initiiert werden.

6.1 Anwendungsregeln für den Einsatz von Multipay im Modus 0

Die gewünschten Zahlarten müssen Sie mit Pago absprechen. Gemäß dieser Absprache richtet Pago Bankkonten, Kreditkarten- und Paybox-Parameter ein und teilt sie, soweit notwendig, dem aufrufenden System mit.

Ein Auftrag wird mit dem Ziel eingestellt, einen Geldbetrag in einer vorher bekannten Maximalhöhe vom Kunden zu erhalten. Beim Anlegen des Auftrags wird dieser Gesamtbetrag an Pago übergeben. Für den Betrag gelten die drei folgenden Regeln:

- (§1) Die Summe aller Einzüge pro Auftrag darf den Gesamtbetrag nicht überschreiten.
- (§2) Der Gesamtbetrag darf nicht nachträglich erhöht werden.
- (§3) Die Summe aller Gutschriftenbeträge darf die Summe aller Forderungsbeträge nicht überschreiten.

Wenn die Forderung erst zu einem späteren Zeitpunkt entsteht, kann die Zahlung zum Zeitpunkt der Auftragseinstellung lediglich autorisiert werden (Modus 0). Bei Online-Zahlarten wird der Einzug durch den Aufruf von **MultipaySettleAccountRequest** zum Zeitpunkt der (Teil-)Lieferung initiiert, und zwar in Höhe der bis zu diesem Zeitpunkt verbuchten noch nicht eingezogenen Forderungen. Wenn für den Vorgang lediglich eine Forderung benötigt wird, kann der Aufruf **MultipaySettleAccountRequest** entfallen. In diesem Fall muss bei **MultipayInsertDebitRequest** der Parameter **CollectNow = true** verwendet werden.

Nicht nur die Forderungen, die aus der Bestellung des Kunden hervorgehen, können storniert werden, sondern auch Gutschriften, die z. B. bei Beschädigung der Ware oder Retoure ausgestellt werden.

Darüber hinaus wird zwischen Buchungen und Stornos unterschieden. Forderungsbuchungen und Gutschriftenstornos werden in positives Soll gebucht, Forderungsstornos und Gutschriftenbuchungen in negatives Soll. Dabei gelten folgende Regeln:

- (§4) Die Summe der Stornos darf pro Vorgang die Summe der Buchungen nicht überschreiten.
- (§5) Die Summe aller Forderungsbuchungen darf den Gesamtbetrag nicht überschreiten.

! *Ein einmal stornierter Betrag (Forderungsstorno, Retoure, Versandkostenstorno) kann nicht mehr gefordert werden.
Eine Gutschrift (aus Kulanz) kann nur auf existierende Zahlungseingänge gewährt werden. Eine Gutschrift kann auch wieder storniert werden, die Regeln in §1 bis §5 sind jedoch immer verbindlich.*

! *Je nach Zahlart gibt es weitere Einschränkungen bezüglich Gutschriften und Stornos. Deshalb sollten Sie Forderungen möglichst zu einem Zeitpunkt einstellen, an dem Sie prozessbedingte Rückabwicklungen abschließen können.*

6.2 Umgang mit Retouren und Gutschriften

Mit folgender Vorgehensweise bleiben die Warenbewegungen und Zahlungen je Vorgang bei allen Parteien übersichtlich und nachvollziehbar. Rückbelastungen aufgrund unklarer Abbuchungen des Kunden können reduziert werden.

- Situation 1: Dem Kunden wird nach einer Warenretoure Ersatz mit gleichem Wert geliefert:
Diese Situation ist für den Vorgang in Multipay nicht relevant.
- Situation 2: Dem Kunden wird nach einer Warenretoure Ersatz mit geringerem Wert geliefert. Zum Ausgleich soll er eine Gutschrift erhalten:
Die Gutschrift über den gewählten Betrag wird mittels **MultipayInsertDebitRequest** angelegt und, je nach Bedarf, entweder sofort oder mit **MultipaySettleAccount** zur Auszahlung gebracht.
- Situation 3: Dem Kunden wird nach einer Warenretoure der gezahlte Betrag erstattet. Zu einem späteren Zeitpunkt wird erneut eine Ware an ihn gesandt:
Mit der Warenretoure sollte der zugehörige Vorgang geschlossen werden. Dazu wird mit **MultipayInsertDebitRequest** eine Retoure eingestellt. Der Gutschriftsbetrag wird nach Bedarf sofort oder mit dem nächsten **MultipaySettleAccount** ausgezahlt.
Für die nächste Bestellung wird mit **MultipayInitPayment** ein neuer Vorgang angelegt.

6.3 Transaktionssicherheit

6.3.1 Aufrufwiederholung bei Verbindungsabbruch

In der Verbindung zwischen dem Mandantensystem und Pago kann es vorkommen, dass der Aufrufende kein Ergebnis erhält. Das kann u. a. aus zwei Gründen der Fall sein:

- Der Empfänger hat den Aufruf nicht erhalten.
- Der Empfänger hat den Aufruf erhalten und verarbeitet, aber die Rückmeldung ist nicht beim Absender angekommen.

Um solche Vorfälle zu vermeiden, besitzt die Anbindung über LocalServer und XML einen speziellen Retry-Mechanismus. Dieser Mechanismus identifiziert jede Kommunikation einer Transaktion eindeutig mit Hilfe der **communicationId**.

Wird der Response nicht innerhalb einer definierten Zeitspanne empfangen, die über die Konfigurationsvariable **Timeout** definiert wird, sollte der Request mit **retry = true** und der gleichen **communicationId** erneut gesendet werden.

Mit dem Retry können drei Situationen eintreten:

- Wenn Pago den Request vor dem Retry nicht erhalten hat, wird der Request nach dem Retry von Pago verarbeitet und der entsprechende Response zurückgesendet.
- Wenn Pago den Request vor dem Retry erhalten und auch verarbeitet hat, wird nach dem Retry das Ergebnis dieser Verarbeitung im Response zurückgesendet.
- Wenn Pago den Request vor dem Retry erhalten hat und die Verarbeitung noch andauert, wird nach dem Retry im Response die Meldung zurückgesendet: **this transaction is still in progress**. Dann muss der Request mit einem weiteren Retry wiederholt werden.

Das folgende Beispiel zeigt den Response im Falle der noch andauernden Verarbeitung. Nähere Informationen hierzu finden Sie in Abschnitt 5.2 „Header-Block“.

```
</header>
<response>
  <errorNotification type="recoverable" issuer="PagoServer"
    service="MultipayInitGuiPayment">
    <errorText> <![CDATA[this transaction is still in progress]]> </errorText>
    <errorNumber> 30007 </errorNumber>
  </errorNotification>
</response>
</PAGOxml>
```

Wenn das Ergebnis der Verarbeitung vorliegt, kann es in eine der folgenden Klassen eingestuft werden:

- 1) Erfolgreiche Durchführung
- 2) Die Transaktion wurde mit nicht behebbaren Fehlern durchgeführt (**fatal**). Eine Wiederholung wird kein anderes Ergebnis hervorbringen. Daher sollten keine Wiederholungsaufrufe durchgeführt werden.

- 3) Die Transaktion wurde mit einem Fehler durchgeführt, der zu einem späterem Zeitpunkt möglicherweise nicht mehr auftritt. Zur Kennzeichnung des möglichen späteren Wiederholungsversuchs wird der Fehler mit **recoverable** klassifiziert.
Wenn in einem Fehlerszenario die Bank bei einer Autorisierung nicht erreicht wird, kann der Betrag auch nicht bei der betreffenden Kreditkarte reserviert werden. Die Initialisierung ist nicht abgeschlossen. Die Transaktion wird mit dem negativen Ergebnis **recoverable** abgeschlossen.
Die Autorisierung kann zu einem späteren Zeitpunkt erneut ausgeführt werden. Der erneute Autorisierungsversuch wird mit alter **communicationId** und gleicher Fremdbelegnummer (**CustomerReference**) initialisiert.

Die Aufrufwiederholung ist mehrfach ausführbar. Wenn Sie also nach einem Aufruf kein Ergebnis zurückerhalten haben, müssen Sie im Zweifelsfall den Aufruf solange wiederholen, bis Sie Klarheit über das Ergebnis haben. Wählen Sie dabei bitte einen sinnvollen Zeitabstand zwischen den Wiederholungsanfragen. Einen Funktionsaufruf, der wegen Timeout nicht abgeschlossen werden konnte, können Sie sofort wiederholen. Wenn eines der angebundenen Systeme (z. B. ein Banksystem) nicht zur Verfügung steht, wiederholen Sie den Aufruf nach einigen Minuten.

6.3.2 Handhabung der Sequenznummer

Die Sequenznummer dient der Einhaltung der Verarbeitungsreihenfolge. Pro Fremdbelegnummer muss die Sequenznummer aufsteigend und lückenlos nummeriert werden. Die Auftragsinitialisierung trägt die implizite Sequenznummer „0“. Verwenden Sie deshalb bei der als nächstes aufgerufenen Methode die Sequenznummer „1“. Eine falsche Sequenznummer wird mit einem Fehler abgelehnt. Für weitere Informationen zu Fehlermeldungen siehe auch Abschnitt 5.5 „Fehlermanagement“.

6.3.3 Verhalten der Methoden bei Wiederholungsaufrufen

Nach einem Wiederholungsaufruf mit gesetztem Retry-Flag wird das Ergebnis der ersten erfolgreichen bzw. fatalen Verarbeitung im Response zurückgeliefert. Die folgende Angabe gilt für Transaktionen, die durch das retryable-Flag wiederholbar sind:

```
<errorNotification issuer="PagoServer"
    service="MultipayInitPayment" type="retryable">
```

Die Funktionen aus Pago Multipay verhalten sich folgendermaßen:

MultipayInitPaymentRequest wird identifiziert durch

- **ClientName**
- **SalesChannel**
- **Branch**
- **CustomerReference**.

Wiederholungsanfragen zu fehlgeschlagenen oder abgelehnten Autorisierungen werden erneut durchgeführt. Wiederholungsanfragen zu akzeptierten Autorisierungen liefern das vorher erhaltene Ergebnis zurück, wenn sie in den anderen Parametern übereinstimmen. Andernfalls werden sie abgelehnt.

MultipayInsertDebitRequest, **MultipaySettleAccountRequest** und **MultipayReduceAmountRequest** werden identifiziert durch

- **ClientName**
- **SalesChannel**
- **Branch**
- **CustomerReference**
- **SequenceNumber**

Wiederholungsanfragen liefern das vorher erhaltene Ergebnis zurück. Anfragen mit falscher Sequenznummer werden abgelehnt.

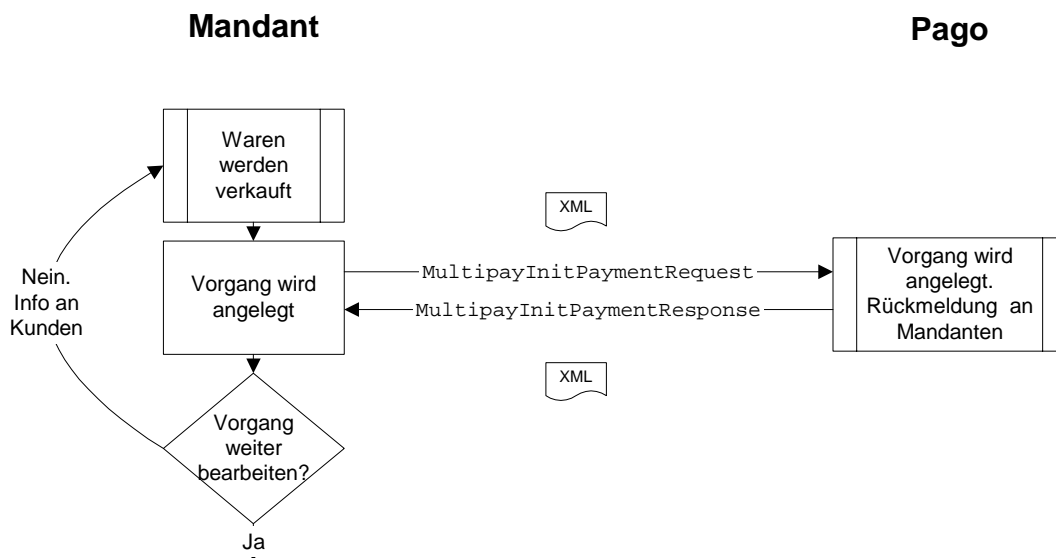
6.4 Anwendungsbeispiele für Multipay mit Modus 0

Von den drei Modi „0“, „1“ und „2“ bietet der Modus „0“ die meisten Einsatzmöglichkeiten. In diesem Modus wird das gesamte Payment mit Multipay gesteuert. Durch die Integration verschiedener Anwendungen in eine Payment-Schnittstelle können Sie die unterschiedlichsten Business Cases über diesen Service abwickeln.

In den folgenden Abschnitten stellen wir Ihnen die vier elementaren Anwendungsfälle beispielhaft vor. Die Beispiele veranschaulichen Ihnen Ihre Möglichkeiten und bieten außerdem eine Diskussionsgrundlage für die Unterstützung durch Pago Consulting.

6.4.1 Autorisierung für einen Vorgang

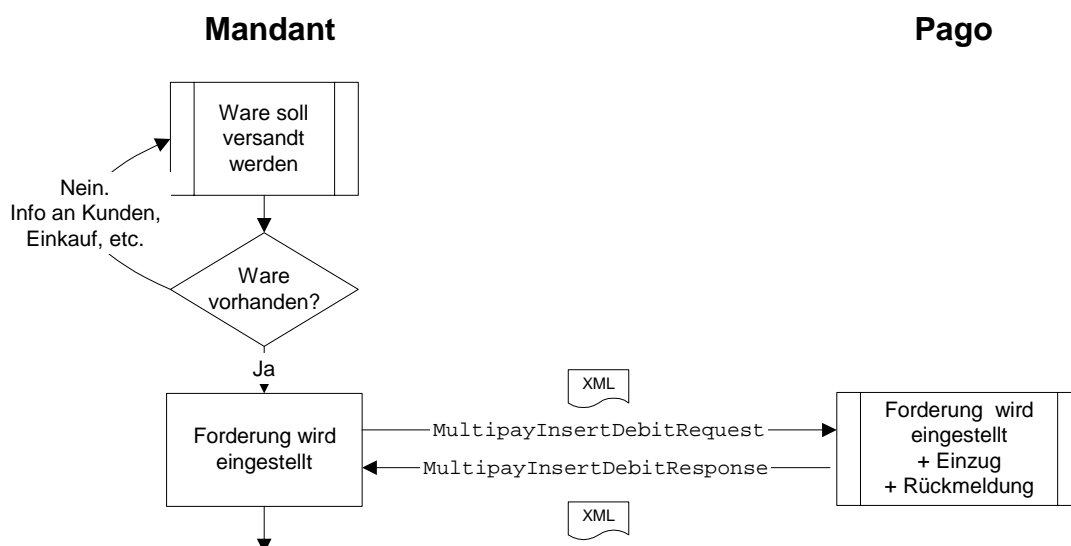
Nach dem Kaufvorgang wird ein Vorgang bei Pago angelegt und die Autorisierung vorgenommen:



6.4.2 Einstellen einer Forderung mit anschließendem Einzug

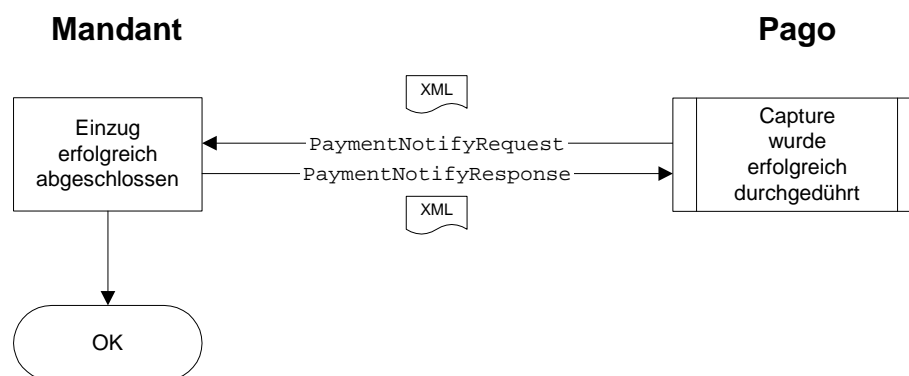
Es wird sichergestellt, dass die Ware am Lager ist; dann wird eine Forderung eingestellt. Die Ware wird versendet und der Forderungsbetrag eingelöst.

(Der Parameter **CollectNow = true** bedeutet, dass der Betrag eingezogen wird.)

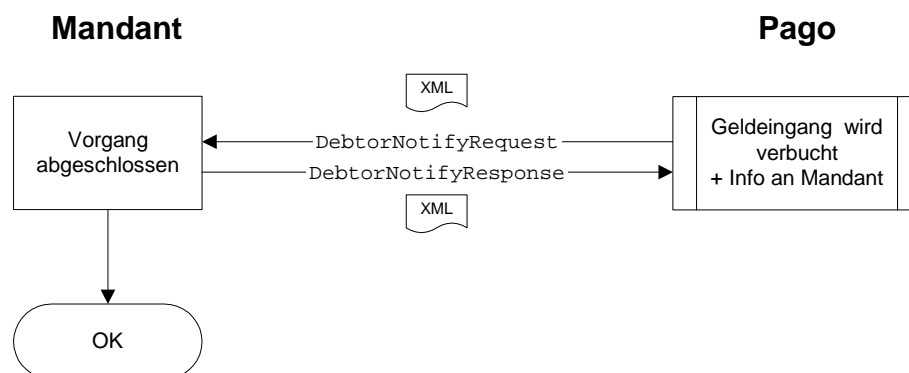


6.4.3 Rückmeldung mit dem Ergebnis der Autorisierung und dem veränderten Debitorenstatus

Nach dem Einstellen der Forderung mit **CollectNow = true** wird der Saldo aus den Forderungen bei Kreditkarte, Lastschrift und Paybox eingelöst. Bei einer Kreditkartentransaktion wird dieser Vorgang „Capture“ genannt. Nach dem Capture wird das Ergebnis mittels **PaymentNotifyRequest** an das Mandantensystem übermittelt.



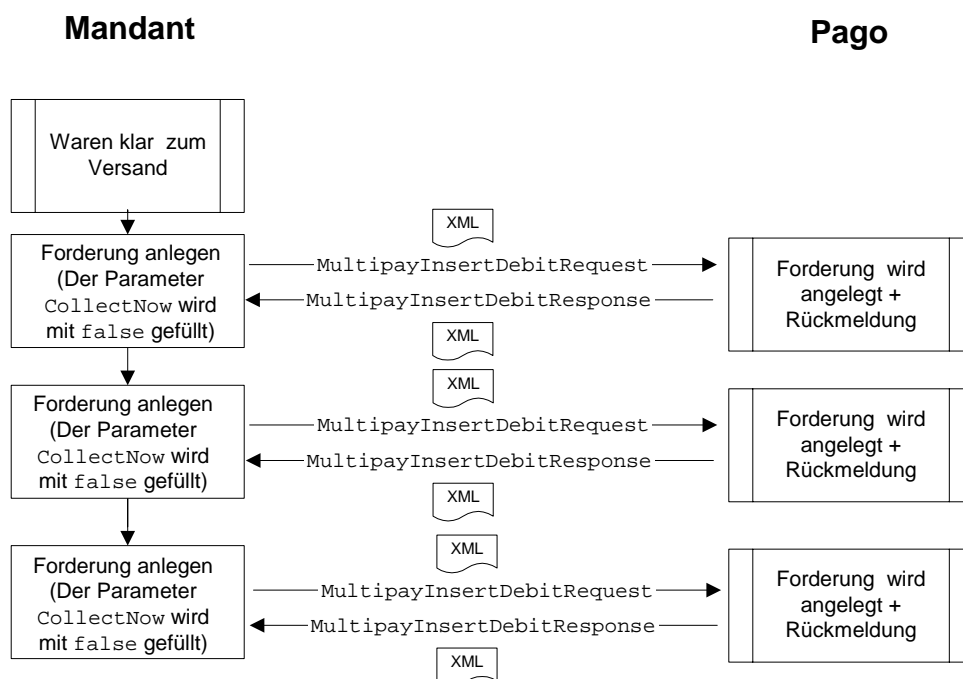
Der Betrag geht bei Pago ein, und die Forderung wird ausgeglichen. Der Mandant (z. B. Online Shop) wird über den veränderten Debitorenstatus informiert.



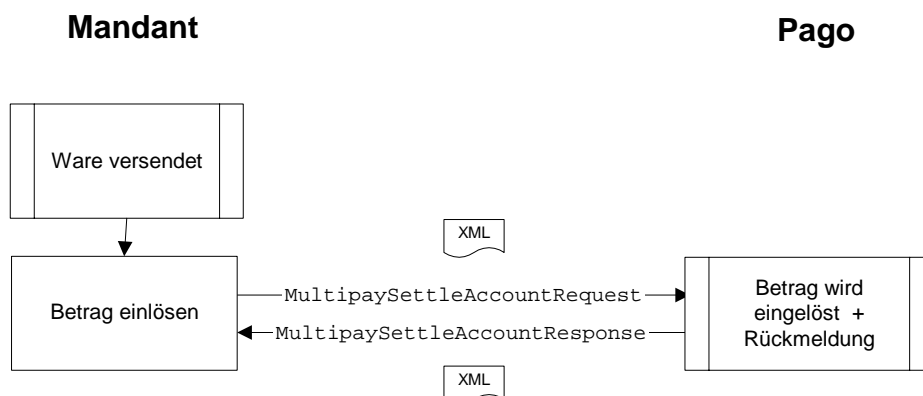
6.4.4 Einstellen mehrerer Forderungen mit anschließendem Einzug des Saldos

Die Einteilung in unterschiedliche Forderungen ist sinnvoll für die Unterscheidung zwischen Warenwert, Mehrwertsteuer, Versandkosten. Die Forderungen können auch mit unterschiedlichen MwSt-Sätzen eingestellt und in den Reports aus dem Debitorenmanagement einschließlich der MwSt-Sätze dargestellt werden.

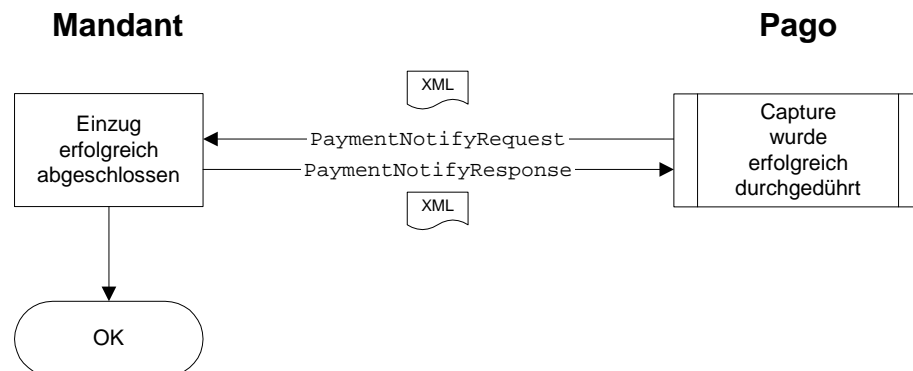
Ein anderer Grund für das Einstellen einzelner Warenpositionen ist die unterschiedliche Verfügbarkeit der Waren auf Lager oder gar in mehreren Lagern. In diesem Beispiel wird gezeigt, wie mit mehreren Forderungssätzen für Warenwert, Versandkosten und Mehrwertsteuer gearbeitet werden kann.



Der Geschäftsvorgang (Auslieferung) ist über ein Warenwirtschaftssystem abgeschlossen und der Saldo der offenen Forderungen soll nun eingelöst werden. Der Einzug wird über **MultipaySettleAccountRequest** ausgelöst, da vorher die Forderungen mit dem Parameter **CollectNow = false** eingestellt wurden. Durch **MultipaySettleAccountRequest** werden die Forderungen für einen Einzug zusammengefasst.



Nach dem Capture wird das Ergebnis mit **PaymentNotifyRequest** an das Mandantensystem übermittelt.

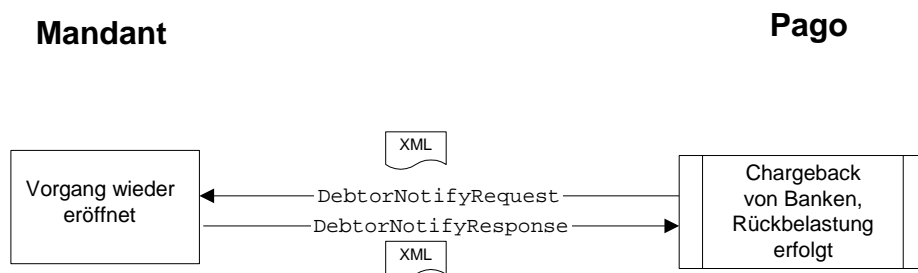


Der Debitorenstatus gibt Auskunft über den Geldeingang aus dem Debitorensystem.

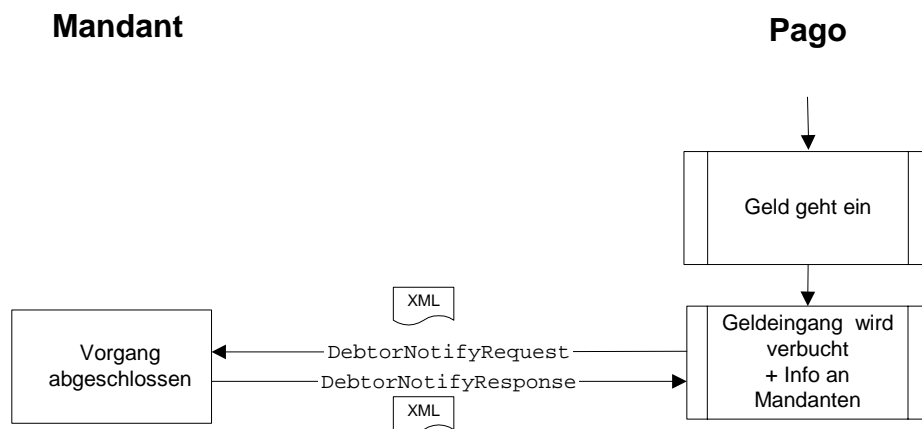


6.4.5 Meldung nach Chargeback

Statusinformationen aus dem Debitorenmanagement können auch nach dem Abschluss eines Vorgangs auftreten. In diesem Beispiel nehmen wir ein Chargeback auf eine Kreditkartenzahlung oder eine Rücklastschrift auf eine Einzugsermächtigung an. Der Debitorenstatus wird von Pago mittels **DebtorNotifyRequest** bereitgestellt.



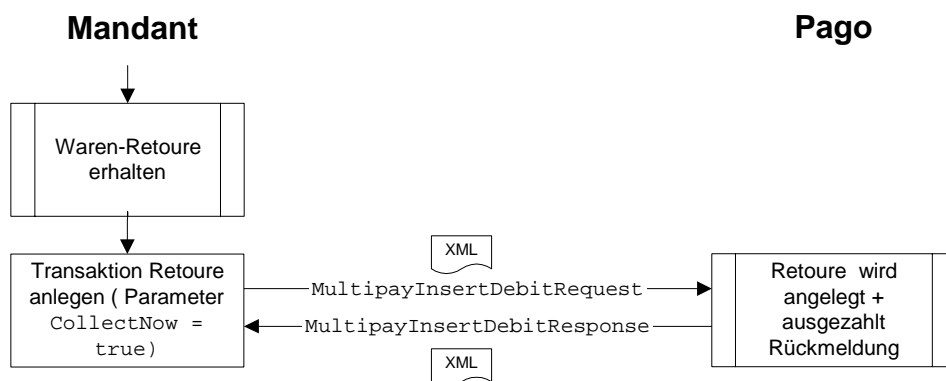
Das Mahnwesen wird in diesem Fall über Pago abgewickelt. Der Kunde reagiert auf die Mahnung und zahlt. Der Geldeingang wird über den Debitorenstatus mitgeteilt.



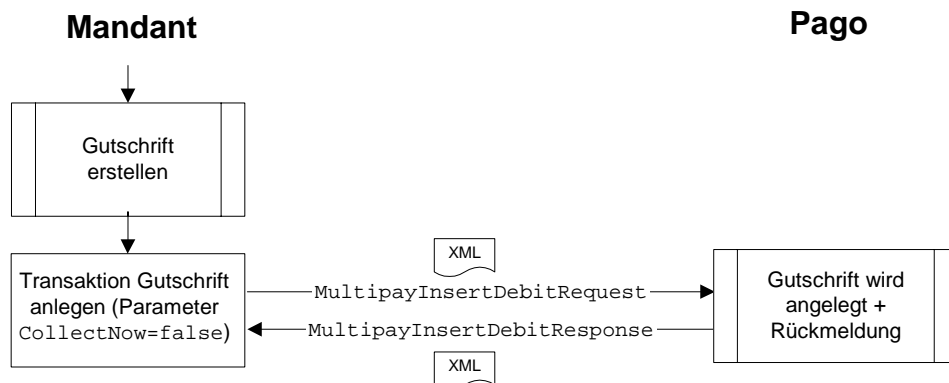
6.4.6 Gutschrift

Gutschriften werden mit **MultipayInsertDebitRequest** eingestellt. (Achten Sie hierbei auf die Regeln aus Abschnitt 6.1 „Anwendungsregeln für den Einsatz von Multipay im Modus 0“).

Dieses Beispiel zeigt den möglichen Ablauf für die Verarbeitung einer Retoure und einer Gutschrift. Die Retoure wird mit sofortiger Auszahlung angelegt.

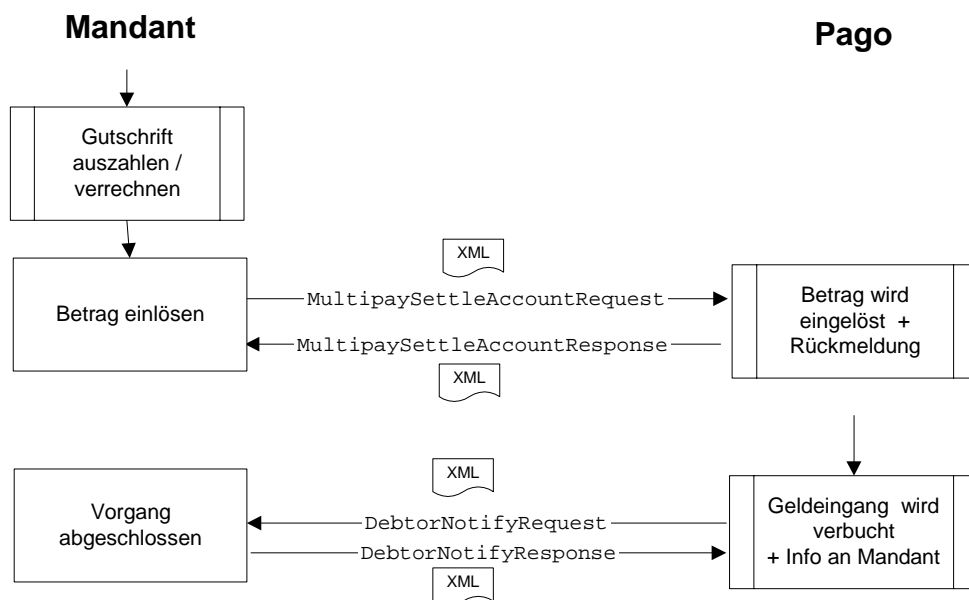


Wenn eine Gutschrift nicht ausgezahlt, sondern mit Forderungen verrechnet werden soll, verwenden Sie den Parameter **CollectNow = false**. Die Verrechnung erfolgt mit dem nächsten **MultipaySettleAccountRequest** oder dem nächsten **MultipayInsertDebitRequest**, wenn der Parameter dann **CollectNow = true** gesetzt ist.



Nach **MultipaySettleAccountRequest** wird die Gutschrift ausgeführt. Wenn in der Zwischenzeit eine neue Forderung eingestellt wird, wird diese mit der Gutschrift verrechnet und nur noch der Saldo ausgezahlt.

! *Durch eine Gutschrift wird eine Forderung nicht aufgehoben. Die Summe der Forderungen muss weiterhin kleiner oder gleich dem Betrag des **MultipayInitPaymentRequest** sein.*



Wenn mehrere Teilvorgänge eines Payment-Vorgangs gleichzeitig zur Verarbeitung anstehen, werden sie zu einem einzigen Payment-Vorgang zusammengefasst. In dem Fall wird die Zusammenfassung zurückgemeldet. Deshalb sind die Zahlen der einzelnen Teilvorgänge in der Rückmeldung nicht mehr ersichtlich. Bei der Verarbeitung innerhalb der Bank kann es weitere Zusammenfassungen geben, die ebenfalls in zusammengefasster Form zurückgemeldet werden.

6.5 Technische Schnittstellenbeschreibung

Der Service Pago Multipay wird integriert durch das Senden und Empfangen von XML-Dokumenten via LocalServer. Der Aufbau der benötigten XML-Dokumente wird nachfolgend beschrieben.

Die Pago-Multipay-Services im Modus 0 werden über folgende XML-Requests bereitgestellt:

- **MultipayCheckBankRequest** zum Prüfen einer Bankverbindung
- **MultipayCheckCreditCardRequest** zum Prüfen einer Kreditkarte
- **MultipayInitPaymentRequest** zum Initiieren von Zahlungen
- **MultipayInsertDebitRequest** zum Einstellen von Forderungsbuchungen
- **MultipaySettleAccountRequest** zum Anstoßen des Einzugs
- **MultipayReduceAmountRequest** zur Reduzierung des reservierten Betrags.

Die Fremdbelegnummer bezieht sich auf einen Vorgang im Mandantensystem, der über Pago Multipay abgewickelt wird (z. B. ein Kauf). Sie muss dem entsprechenden Vorgang beim Mandanten eindeutig zugeordnet werden können. Für die Fremdbelegnummer sollten ausschließlich Ziffern und Großbuchstaben verwendet werden, da das System Kleinbuchstaben automatisch in Großbuchstaben umsetzt. Falls zwei Fremdbelegnummern angelegt werden, die sich nur durch Groß- und Kleinschreibung voneinander unterscheiden, können sie vom System nicht mehr eindeutig zugeordnet werden.

! Halten Sie sich bei Datumseingaben genau an das vorgegebene Format (YYYYMMDD oder DD.MM.YYYY). Lassen Sie keine Stelle aus, auch nicht die führende „0“ (z. B.: „01.11.2000“).

Legende

Die Parameter des Request und des Response sind in Tabellen dargestellt. Für die Tabellen gilt folgende Legende:

Tag:	Feldbezeichnung
Datentyp:	„an“ = alphanumerisch / „n“ = numerisch
Feldlänge:	Anzahl der Zeichen
Pflichtfeld:	Angabe notwendig (Ja/Nein)



Bei numerischen Feldern, in denen eine Zahl mit Nachkommastellen eingetragen werden kann, sind unter „Feldlänge“ zwei Zahlen angegeben. Die Zahl vor dem Komma gibt die maximale Anzahl der gesamten Stellen an, die Zahl hinter dem Komma die maximale Anzahl der Nachkommastellen.

Beispiel: **10,2** --> max. 10 Stellen, davon max. 2 Nachkommastellen.

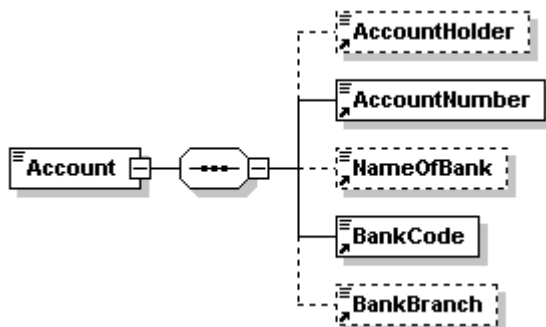
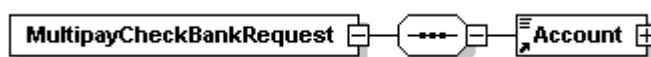
6.5.1 Prüfen einer Bankverbindung

Eine Bankverbindung wird mit **MultipayCheckBankRequest** überprüft. Hierzu dienen

- die Bankleitzahl
- die Kontonummer
- das jeweilige Prüfziffernverfahren.

Die Prüfung der Bankverbindung sollte nach einer Änderung von Stammdaten erfolgen. So können unplausible Daten früh erkannt und beseitigt werden.

Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:



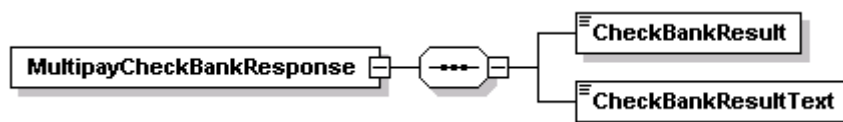
Tag	Daten typ	Feldlänge	Pflichtfeld	Bemerkung
Account				
Account-Holder	an	30	Nein	Kontoinhaber (nur bei Zahlart ELV) Fixwert: „“ (Leerstring)

Tag	Daten- typ	Feld- länge	Pflicht- feld	Bemerkung
Account- Number	an	10	Ja	Kontonummer (nur bei Zahlart ELV)
NameOfBank	an	30	Nein	Bankname (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)
BankCode	an	8	Ja	Bankleitzahl (nur bei Zahlart ELV)
BankBranch	an	30	Nein	Bankfiliale (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)

Beispiel

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:00:30+02:00-000000000123-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>>false</retry>
  </header>
  <request>
    <MultipayCheckBankRequest>
      <Account>
        <AccountNumber>1234567890</AccountNumber>
        <BankCode>87654321</BankCode>
      </Account>
    </MultipayCheckBankRequest>
  </request>
</PAGOxml>
```



```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>2001-12-31T12:30:30+02:00-001122334455-00601@www.pago.de
    </communicationId>
    <Server>PagoServer</Server>
    <retry>>false</retry>
    <inReplyTo>2001-12-31T12:00:30+02:00-000000000123-00001@www.client.url
```

```
</inReplyTo>
</header>
<response>
<MultipayCheckBankResponse>
  <CheckBankResult>0</CheckBankResult>
  <CheckBankResultText>ACCOUNT VALID</CheckBankResultText>
</MultipayCheckBankResponse>
</response>
</PAGOxml>
```

6.5.2 Prüfen einer Kreditkarte

Die Kreditkarte wird mit **MultipayCheckCreditCardRequest** überprüft. Dies geschieht nach folgenden Kriterien:

- Kreditkarten-Anfangsziffern
- Kreditkartengesellschaft
- Prüzfiffer der Kreditkartennummer
- Verfalldatum der Kreditkarte.

Die Prüfung der Kreditkarte sollte nach einer Änderung von Stammdaten erfolgen. So können unplausible Daten womöglich früh erkannt und beseitigt werden.

MultipayCheckCreditCard liefert Ihnen zwei Ergebnisfelder:

- **CheckCreditCardResult**
- **CheckCreditCardResultText**.

CheckCredit-CardResult	CheckCreditCardResultText		Ergebnisklasse
0	Kreditkarte existiert		OK
-1	Kreditkarte abgelaufen		fatal
-2	Falsche Kartennummernlänge		fatal
-3	Ungültige Eingabezeichen		fatal
-4	Keine numerische Kartennummer		fatal
-5	Unbekannter Kreditkartentyp		fatal
-6	Ungültige Checksumme		fatal
-7	Falsches Datumsformat		fatal
-8	Ablehnungsgrund der Prüfstelle (nur bei Online-Prüfung) Original-Fehlertext des Netzbetreibers der Kreditkarte wird ausgegeben. Liste der Fehlercodes:		
	02	Anrufen der AS-Rufnummer	fatal
	03	Nummer des Vertragsunternehmens unbekannt	fatal

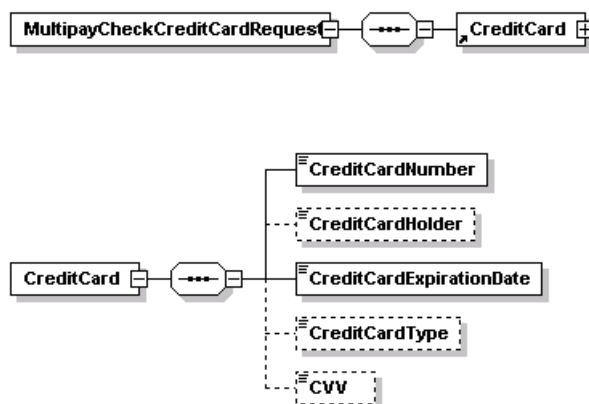
CheckCredit-CardResult	CheckCreditCardResultText		Ergebnisklasse
	04	Karte einbehalten	fatal
	05	Autorisierung abgelehnt	fatal
	06	Filetransfer falsch, Wiederholung ab mitgeteilter Blocknummer möglich	fatal
	09	Verzögerte Bearbeitung der Autorisierung	fatal
	12	Ungültige Transaktion	fatal
	13	Verfügbarer Betrag überschritten	fatal
	14	Kartenummer nicht bekannt	fatal
	21	Kein Vorgang vorgenommen	fatal
	24	Filetransfer nicht unterstützt	fatal
	29	Filetransfer nicht erfolgreich	fatal
	30	Formatfehler	fatal
	31	Kartenherausgeber nicht zugelassen	fatal
	33	Verfalldatum überschritten	fatal
	34	Manipulationsverdacht	fatal
	40	Funktion ungültig	fatal
	43	Gestohlene Karte; Karte einbehalten	fatal
	50	Autorisierung doppelt	fatal
	51	Limit überschritten; „Doch“-Funktion möglich	fatal
	54	EC-Chipkarte nicht mehr gültig	fatal
	55	PIN falsch	fatal
	56	Karte nicht im Bestand	fatal
	57	Andere Karte als bei der Bezugstransaktion benutzt wurde (bei Storno; Kreditkarten auch bei Erweiterungsreservierung, Endsummenbuchung, Trinkgeld-Update)	fatal
	58	Terminal-ID nicht erkannt	fatal
	59	Manipulationsverdacht bei EC-Chipkarte	fatal
	61	Karte gesperrt in lokaler Sperrliste	fatal
	62	Karte gesperrt	fatal

CheckCredit-CardResult	CheckCreditCardResultText		Ergebnisklasse
	64	Transaktionsbetrag weicht von Bezugs- transaktion ab (bei Storno; Kreditkarten auch bei Endsummenbuchung)	fatal
	65	Limit der Transaktionsfrequenz über- schritten; „Doch“-Funktion möglich	fatal
	75	Zulässige Anzahl von PIN-Eingaben über- schritten	fatal
	76	Schlüsselindex unzulässig bzw. Fehler in BMP 57	fatal
	77	PIN-Eingabe erforderlich	fatal
	78	Sequenzfehler in BMP 62	fatal
	80	Betrag ist nicht verfügbar	fatal
	81	Initialisierung fehlerhaft; Wiederholung erforderlich	fatal
	82	(Vor-)Initialisierung unzulässig (Terminal gesperrt)	fatal
	83	PIN-Pad-Wechsel nicht möglich	fatal
	84	Datex-P-Nr. falsch	fatal
	85	Ablehnung vom Kreditkarteninstitut	fatal
	86	Stammdaten unbekannt; bei Init1 OPT- fähiger Terminals auch: ZKA-Nummer falsch	fatal
	87	Terminal (Pin-Pad) unbekannt	fatal
	88	Beim Laden gegen EC-Karte: PIN nicht aktiv	fatal
	89	CRC falsch	fatal
	91	Kartenherausgeber oder Netz nicht ver- fügbar	fatal
	92	Kartentyp wird nicht vom Autorisierungs- system verarbeitet	fatal
	96	Verarbeitung zur Zeit nicht möglich	fatal
	97	Sicherheitsverletzung: MAC-Prüfung zeigt Fehlerzustand	fatal
	98	Datum / Uhrzeit nicht plausibel	fatal
	99	Fehler in der PAC-Verschlüsselung	fatal

CheckCreditCardResult	CheckCreditCardResultText	Ergebnisklasse
99 und 999	Nicht klassifizierte technische Fehler im Payment	recoverable



Die Liste der Original-Fehlertexte des Netzbetreibers hat keinen Anspruch auf Vollständigkeit.



Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:

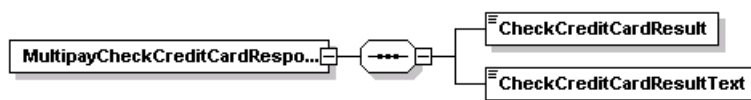
Tag	Datentyp	Feldlänge	Pflichtfeld	Bemerkung
CreditCard				
CreditCard-Number	an	19	Ja	Kreditkartennummer
CreditCard-Holder	an		Nein	(Reserviert für weitere Funktionen)
CreditCard-Expiration-Date	an	6	Ja	Verfalldatum Format: MM/YY

Tag	Datentyp	Feldlänge	Pflichtfeld	Bemerkung
CreditCard-Type	an	1	Ja	Kreditkartentyp V: Visa A: AMEX E: EUROCARD/MasterCard D: Diners J: JCB G: Galeria
CVV	n	4	Nein	Kreditkartenprüfnummer Ziffernfolge mit max. vier Stellen (nähere Informationen siehe Abschnitt 1.3)

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000124-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>true</retry>
  </header>
  <request>
    <MultipayCheckCreditCardRequest>
      <CreditCard>
        <CreditCardNumber>49494949494949</CreditCardNumber>
        <CreditCardExpirationDate>12/02</CreditCardExpirationDate>
        <CreditCardType>V</CreditCardType>
      </CreditCard>
    </MultipayCheckCreditCardRequest>
  </request>
</PAGOxml>
```

Der **MultipayCheckCreditCardResponse** ist wie folgt aufgebaut:



Tag	Daten typ	Feld- länge	Bemerkung
CheckCredit- CardResult	n	5	Ergebnis als Nummer (siehe Informationen weiter oben)
CheckCredit- CardResultText	an	40	Ergebnis als Text

```
<?xml version="1.0" encoding="ISO-8859-1"?>

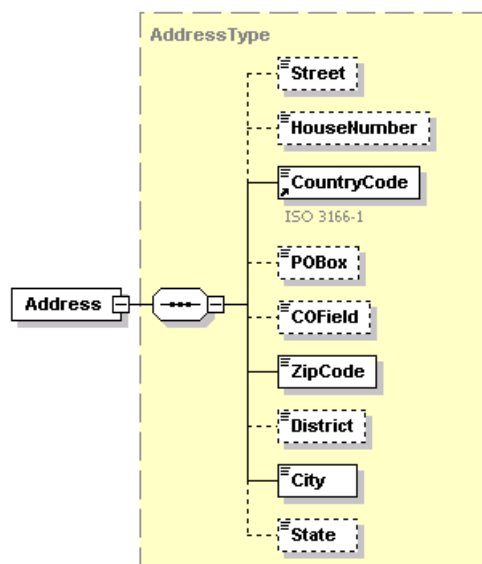
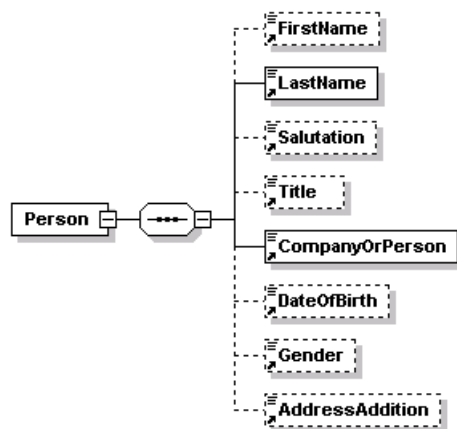
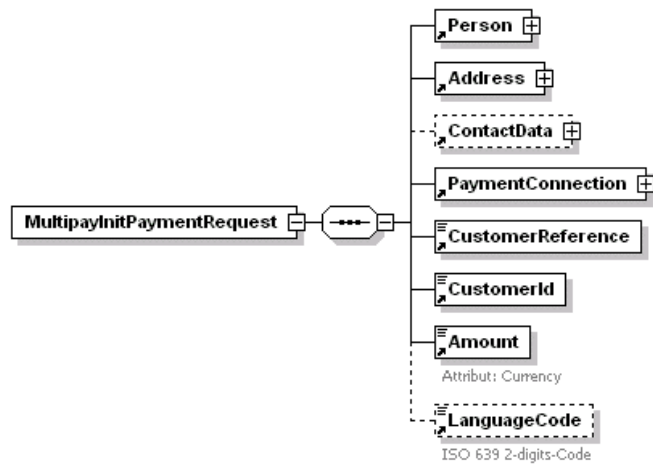
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
<header>
<communicationId>2001-12-31T12:31:30+02:00-001122334456-00601@www.pago.de
</communicationId>
<Server>PagoServer</Server>
<retry>false</retry>
<inReplyTo>2001-12-31T12:02:30+02:00-000000000124-00001@www.client.url
</inReplyTo>
</header>
<response>
<MultipayCheckCreditCardResponse>
  <CheckCreditCardResult>-6</CheckCreditCardResult>
  <CheckCreditCardResultText>INVALID CHECKSUM
  </CheckCreditCardResultText>
</MultipayCheckCreditCardResponse>
</response>
</PAGOxml>
```

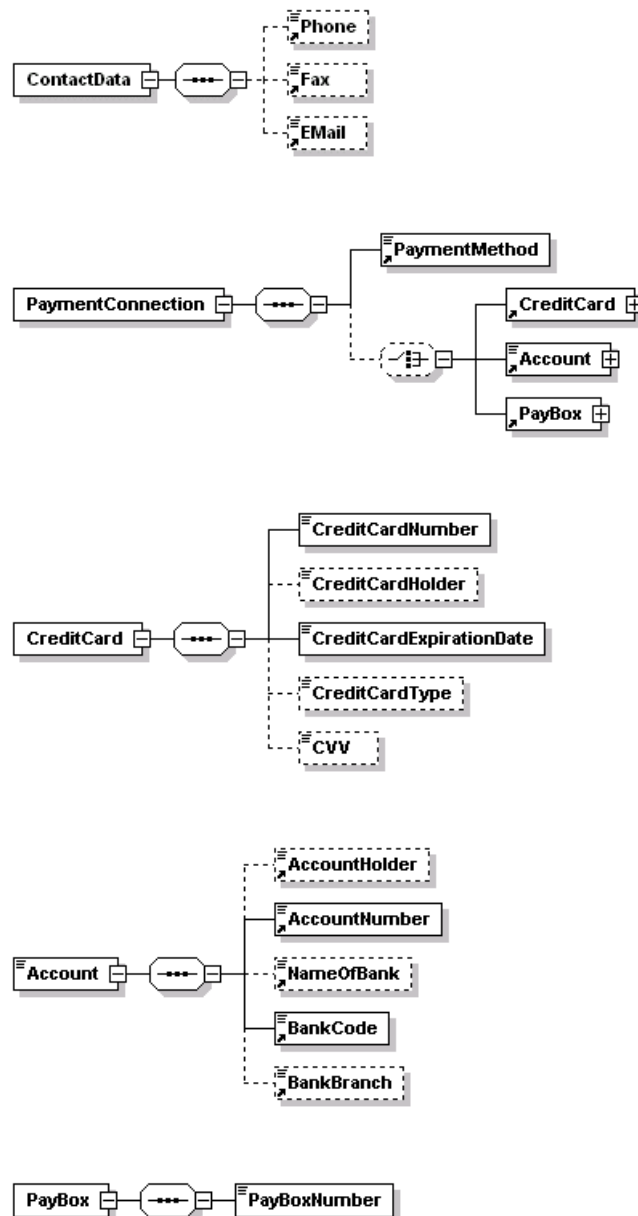
Fehlercode	Grund	Abhilfe
106.500	Aus DB: MCB-KLIENT_UNGUELTIG	

6.5.3 Initiieren von Zahlungen

Zahlungen werden mit **MultipayInitPaymentRequest** initiiert. Der Vorgang wird durch eine Fremdbelegnummer eindeutig gekennzeichnet.

Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:





Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
Mode	n	3	Ja	<p>Modus 0:</p> <p>Anlegen des Vorgangs bei allen Zahlarten und Prüfung/Autorisierung bei ELV, KKE, PBX mit späterem Geldfluss</p> <p>Weitere Informationen zum Modus und dem damit verbundenen Businessmodell finden Sie in Kapitel 3.</p>

Tag	Daten- typ	Feld- länge	Pflicht- feld	Bemerkung
Person				
FirstName	an	30	Nein	Vorname
LastName	an	30	Ja	Nachname oder Firmenname
Salutation	an	10	Nein	Anrede, z. B. „Herr“, „Frau“, „Firma“
Title	an	20	Nein	Titel, z. B. „Dr.“, „Prof.“
CompanyOr- Person	char	1	Ja	F (Firma) oder P (Person)
DateOfBirth	an	10	Nein	Geburtsdatum Format: YYYYMMDD oder DD.MM.YYYY
Gender	char	1	Nein	Geschlecht Fixwert: „“ (Leerstring)
Address- Addition	an	60	Nein	Adresszusatz z. B. „7. Stock“, „bei Maier“, „Firma Müller“, z. H. Herrn Schmidt
Adresse				
Street	an	46	Ja	Straßenname
HouseNumber	an	35	Ja	Hausnummer
CountryCode	an	3	Ja	gültiges Länderkennzeichen nach ISO 3166 (siehe Kapitel 10)
POBox	n	35	Nein	Postfach-Nummer
COField	an	40	Nein	Großempfänger-Name
ZipCode	an	9	Ja	Postleitzahl
District	an	40	Nein	Ortsteil
City	an	40	Ja	Ortsname
State	an	40	Nein	Staat (US-Bundesstaat)
ContactData				
Phone	an	20	Nein	Telefonnummer laut Pago Bill Muss mit „+“ und Ländervorwahl beginnen (z. B. „+49“).
Fax	an	20	Nein	Faxnummer laut Pago Bill Muss mit „+“ und Ländervorwahl beginnen (z. B. „+49“).

Tag	Datentyp	Feldlänge	Pflichtfeld	Bemerkung
E-Mail	an	80	Nein	E-Mail-Adresse
PaymentConnection				
PaymentMethod	an	3	Ja	Mit Pago abgesprochene Zahlart, z. B.: ELV Lastschriftverfahren (DTA) KKE Kreditkarteneinzug NN Nachnahme PBX Paybox REC Rechnungskauf VOR Vorkasse
CreditCard (nur bei KKE)				
CreditCard-Number	an	19	Ja	Kreditkartennummer
CreditCard-Holder	an		Nein	(Reserviert für weitere Funktionen)
CreditCard-Expiration-Date	an	6	Ja	Verfalldatum Format: MM/YY
CreditCard-Type	an	1	Ja	Kreditkartentyp V: VISA A: AMEX E: EUROCARD/MasterCard D: DINERS J: JCB G: GALERIA
CVV	n	4	Nein	Kreditkartenprüfnummer Ziffernfolge mit max. vier Stellen (nähere Informationen siehe Abschnitt 1.3)
Account (nur bei ELV)				
Account-Holder	an	30	Nein	Kontoinhaber (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)
Account-Number	an	10	Ja	Kontonummer (nur bei Zahlart ELV)
NameOfBank	an	30	Nein	Bankname (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)
BankCode	an	8	Ja	Bankleitzahl (nur bei Zahlart ELV)
BankBranch	an	30	Nein	Bankfiliale (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)

Tag	Datentyp	Feldlänge	Pflichtfeld	Bemerkung
Paybox (nur bei PBX)				
PayBoxNumber	an	16	Ja	Paybox-Nummer Die Zahlart PBX ist Autorisierung und anschließende Buchung. Modus 0 ist daher nicht notwendig.
CustomerReference	an	20	Ja	Ihre eindeutige Vorgangsreferenz unter ClientName / SalesChannel / Branch (siehe auch Abschnitt 6.5) Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
CustomerId	an	20	Ja	Ihre eindeutige Kundenreferenz unter ClientName / SalesChannel / Branch
Amount				
Amount	n	10,2	Ja	positiver Bruttobetrag inkl. MwSt
Currency	an	3	Ja	Währung gemäß ISO 4217 (z. B. EUR, USD) Currency ist ein Attribut von Amount
LanguageCode	an	3	Nein	Sprachkennzeichen des Kunden Verwendung zukünftig im Mahnwesen

Details über die Verwendung der Adressfelder bei Rechnungsdruck und Mahnwesen finden Sie in dem Dokument „PrintLayout“.



Wiederholte Aufrufe zur selben Fremdbelegnummer erhalten das gleiche alte Ergebnis (Returncode, Message).

Beispiel:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

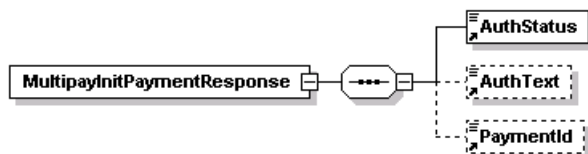
```
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000125-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>true</retry>
  </header>
  <request>
    <MultipayInitPaymentRequest Mode="0">
      <Person>
        <FirstName>HANS</FirstName>
        <LastName>Debtor</LastName>
        <Salutation>Herr</Salutation>
        <CompanyOrPerson>P</CompanyOrPerson>
      </Person>
      <Address>
        <Street>reperbaan</Street>
        <HouseNumber>1</HouseNumber>
        <CountryCode>De</CountryCode>
        <ZipCode>2000</ZipCode>
        <City>hammburg</City>
      </Address>
      <PaymentConnection>
        <PaymentMethod>KKE</PaymentMethod>
        <CreditCard>
          <CreditCardNumber>4321432143214321</CreditCardNumber>
          <CreditCardExpirationDate>12/01</CreditCardExpirationDate>
        </CreditCard>
      </PaymentConnection>
      <CustomerReference>987654323</CustomerReference>
      <CustomerId>13447</CustomerId>
      <Amount Currency="EUR">10.0</Amount>
    </MultipayInitPaymentRequest>
  </request>
</PAGOxml>
```

Für die Zahlung auf Rechnung wird nur **PaymentMethod** angegeben. Das folgende Beispiel soll diesen Sachverhalt verdeutlichen.

```
<request>
  <MultipayInitPaymentRequest Mode="0">
    <Person>
      <FirstName>HANS</FirstName>
      <LastName>Debtor</LastName>
      <Salutation>Herr</Salutation>
      <CompanyOrPerson>P</CompanyOrPerson>
    </Person>
    <Address>
      <Street>reperbaan</Street>
      <HouseNumber>1</HouseNumber>
      <CountryCode>De</CountryCode>
      <ZipCode>2000</ZipCode>
      <City>hammburg</City>
    </Address>
    <PaymentConnection>
      <PaymentMethod>REC</PaymentMethod>
    </PaymentConnection>
    <CustomerReference>987654323</CustomerReference>
    <CustomerId>13447</CustomerId>
    <Amount Currency="EUR">10.0</Amount>
```

```
</MultipayInitPaymentRequest>
</request>
```

Die Antwort auf **MultipayInitPaymentRequest** ist **MultipayInitPaymentResponse**. Darin wird der Autorisierungsstatus mit einem Zusatztext geliefert.



MultipayInitPaymentResponse hat folgenden Aufbau:

Tag	Daten typ	Feld- länge	Bemerkung
MultipayInitPaymentResponse			
AuthStatus	n	1	Autorisierungsstatus 0: OK 1: abgelehnt (nur bei Zahlarten mit Autorisierung)
AuthText	an	20	Text zum Autorisierungsstatus
PaymentId	n	38	ID für Rückfragen bei Pago (wird nur intern bei Pago verwendet)

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-10-23T16:10:19+02:00">
<header>
<communicationId>2001-12-31T12:30:32+02:00-001122334457-00601@www.pago.de
</communicationId>
<Server>PagoServer</Server>
<retry>>false</retry>
<inReplyTo>2001-12-31T12:02:30+02:00-000000000125-00001@www.client.url
</inReplyTo>
</header>
<response>
<MultipayInitPaymentResponse>
  <AuthStatus>0</AuthStatus>
  <AuthText>AUTH CODE:6236</AuthText>
  <PaymentId>3422</PaymentId>
</MultipayInitPaymentResponse>
</response>
</PAGOxml>
```

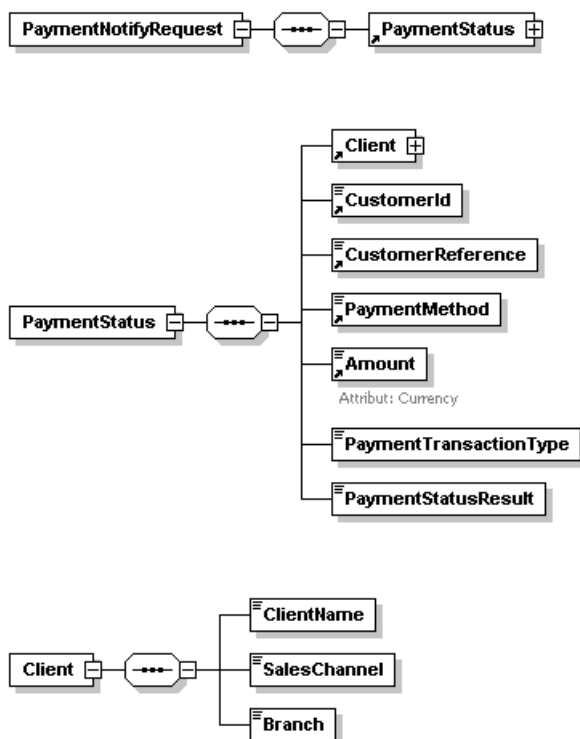
Wenn eine Autorisierung abgelehnt wird, hat der Tag **response** beispielhaft folgenden Aufbau:

```
<response>
<MultipayInitPaymentResponse>
  <AuthStatus>1</AuthStatus>
  <AuthText>AUTHORIZATION DECLINED</AuthText>
  <PaymentId>3411</PaymentId>
</MultipayInitPaymentResponse>
</response>
```

Fehlercode	Grund	Abhilfe
101.001	Kein gültiger numerischer Wert	Zahl verwenden
101.004	Currency ist für diesen Dienst Pflichtfeld	Currency muss gefüllt sein
101.005	Betrag war ≤ 0	Positiven Betrag verwenden
101.006	Mode-Attribut war <0 oder >3	0, 1, 2 oder 3 verwenden
101.007	PaymentMethod passt nicht zum Mode	
101.008	Parameterlänge zu lang oder zu kurz	Parametername steht im Twister-Log
101.009	go_to_payment Ergebnis war „999“	Autor fragen
101.500	Ungültiger Mandant	
101.501	Ungültige Personendaten	
101.502	Ungültige Adressdaten	
101.504	Ungültige Währung	
101.506	Unbekannter Mandant	
101.509	Transaktionsart fehlt	
101.511	Zahlart ist unbekannt	
101.512	Stammdaten sind ungültig	
101.513	Bereits autorisiert	
101.514	Autorisierung in Bearbeitung	
101.515	Auftrag abgelaufen	
101.517	Sequenznummer ist ungültig	

6.5.4 Rückmeldung des Payment-Status

Wenn die Zahlung mit **MultipayInitPaymentRequest** durchgeführt wurde, teilt Pago Ihnen mit **MultipayInitPaymentResponse** (synchrone Rückmeldung) das Ergebnis der Zahlung sofort mit. Das Ergebnis eines Capture wird nach der Batch-Verarbeitung bei Pago und der Rückmeldung seitens der Kreditkartengesellschaft bzw. der Bank über den Payment-Status zu einem späteren Zeitpunkt durch **PaymentNotifyRequest** (asynchrone Rückmeldung) übermittelt. Der Vorgang ist über **CustomerReference** definiert.



Tag	Daten typ	Feld-länge	Bemerkung
PaymentStatus			
Client			
ClientName	an	20	Name des Mandanten Konstante in Absprache mit Pago
SalesChannel	an	20	Vertriebslinie Konstante in Absprache mit Pago
Branch	an	20	Filiale Konstante in Absprache mit Pago
CustomerId	an	20	Ihre eindeutige Kundenreferenz unter ClientName / SalesChannel / Branch

Tag	Daten typ	Feld- länge	Bemerkung
Customer- Reference	an	20	Ihre eindeutige Vorgangsreferenz unter ClientName / SalesChannel / Branch (siehe auch Abschnitt 6.5) Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
PaymentMethod	an	3	Mit Pago abgesprochene Zahlart, z. B. SEL : Lastschriftverfahren (DTA) SKK : Kreditkarteneinzug PBX : Paybox SBX : Stackbox (Initialisierung über Modus GUI) NN : Nachnahme REC : Rechnungskauf VOR : Vorkasse
Amount			
Amount	n	10,2	Saldo des Vorgangskontos negativer Betrag: Guthaben positiver Betrag: Forderung
Currency	an	3	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest

Tag	Daten typ	Feld- länge	Bemerkung
Payment- TransactionType	an	25	Transaktionsart Authorization: Betragsautorisierung Capture: Einlösen auf eine Autorisierung Credit: Auszahlung Gutschrift Exception: Verarbeitung wurde unterbrochen; neuen Status abwarten PBXCapture: Einlösen auf eine Paybox-Autorisierung PBXCaptureCancel: Einlösen widerrufen und damit auszahlen Payment: Sofortbelastung des Betrags auf der Kre- ditkarte PaymentAfterFailedCapture: Sofortbelastung des Betrags nach fehlge- schlagenem Einlösen auf eine Autorisie- rung
PaymentStatus- Result	an	5	Ergebnis der Verarbeitung OK: Verarbeitung erfolgreich durch- geführt Failed: Verarbeitung fehlgeschlagen Error: Fehler während der Ausführung

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:33+02:00">
<header>
  <communicationId>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de>
  <communicationId>
  <Server>PagoServer</Server>
  <retry>false</retry>
</header>
<request>
  <PaymentNotifyRequest>
    <PaymentStatus>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
      <CustomerId>123456789001</CustomerId>
      <CustomerReference>abcdefg2001</CustomerReference>
      <PaymentMethod>ELV</PaymentMethod>
      <Amount Currency="EUR">195.95</Amount>
      <PaymentTransactionType>Payment</PaymentTransactionType>
      <PaymentStatusResult>OK</PaymentStatusResult>
    </PaymentStatus>
  </PaymentNotifyRequest>
</request>
```

</PAGOxml>

Auf **PaymentNotifyRequest** wird von Pago ein **PaymentNotifyResponse** erwartet. Erst mit dem Eingang eines leeren **PaymentNotifyResponse** ohne **errorNotification** gilt die Zustellung des Status als abgeschlossen.

PaymentNotifyResponse

Damit hat das Tag **PaymentNotifyResponse** keine definierten Elemente.

Tag	Datentyp	Feldlänge	Bemerkung
PaymentNotifyResponse			

Das folgende Beispiel zeigt einen möglichen **DebtorNotifyResponse**:

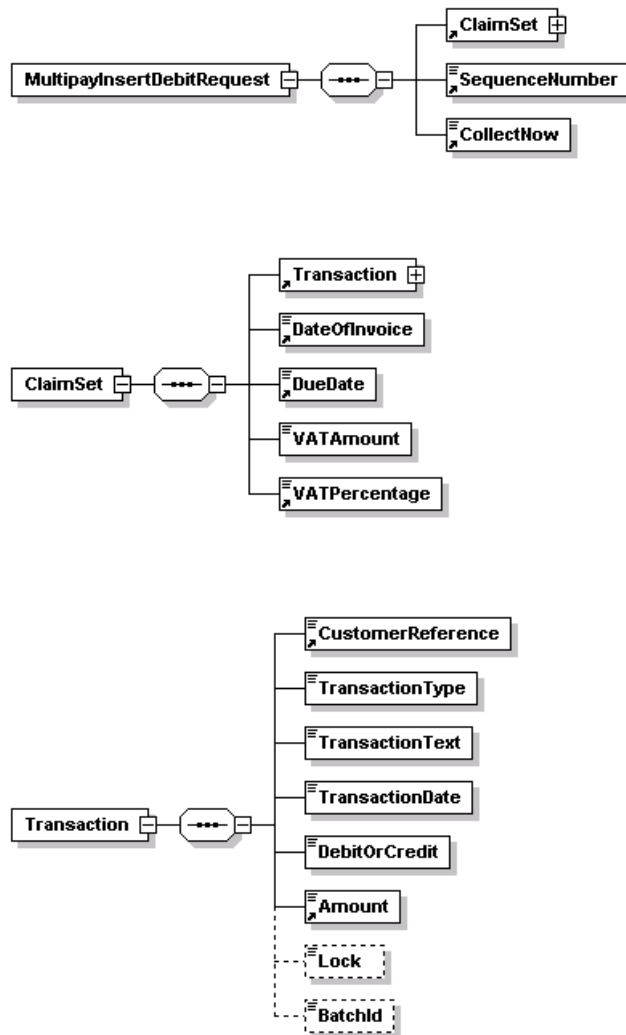
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml timeStamp="2001-10-30T16:45:25+01:00" version="2.0">
  <header>
    <communicationId>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url>
    <communicationId>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
    <retry>false</retry>
    <inReplyTo>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de
    </inReplyTo>
  </header>
  <response>
    <PaymentNotifyResponse/>
  </response>
</PAGOxml>
```

6.5.5 Einstellen von Forderungsbuchungen

Eine Forderungsbuchung stellen Sie mit **MultipayInsertDebitRequest** ein.



Achten Sie darauf, dass die Summe der eingestellten Forderungen nicht höher ist als der bei **MultipayInitPaymentRequest** angegebene Betrag.



Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:

Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
ClaimSet				
Transaction				

Tag		Daten typ	Feld- länge	Pflicht- feld	Bemerkung
	Customer-Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz wie in MultipayInitPaymentRequest angegeben Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
	Transaction-Type	an	6	Ja	Transaktionsart FD = Forderung Ware / Dienstleistung VD = Versandkosten GS = Gutschrift RT = Retoure VO = Voucher ! Die Transaktionsart bestimmt nicht, ob der Betrag im Soll oder Haben gebucht wird.
	Transaction-Text	an	30	Ja	Transaktionstext für Buchhaltung / Kontoauszug
	Transaction-Date	an	10	Ja	Transaktionsdatum (Kaufdatum) Format: YYYYMMDD oder DD.MM.YYYY
	DebitOrCredit	an	1	Ja	Soll oder Haben S : Soll (debit)(FD/VD/GS/RT) Für alle Buchungen ist S zu verwenden. H : Haben (credit) nur intern bei Pago
	Amount				
	Amount	n	10,2	Ja	Bruttobetrag inkl. MwSt + : Forderung oder Stornierung einer Gutschrift - : Gutschrift oder Stornierung einer Forderung
	Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest

Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
Lock	-	-	Nein	Reserviert
BatchId	an	30	Nein	Batch-ID Dient bei Bedarf zur Gruppierung von Sätzen (nur nach Absprache)
DateOfInvoice	an	10	Ja	Rechnungsdatum Format: YYYYMMDD oder DD.MM.YYYY
DueDate	an	10	Ja	Fälligkeitsdatum Format: YYYYMMDD oder DD.MM.YYYY
VATAmount	n	10,2	Ja	Mehrwertsteuerbetrag oder -1 Bei -1 keine Angabe. Wird von Pago berechnet, wenn ein MwSt-Satz angegeben ist.
VATPercentage	n	4,2	Ja	Mehrwertsteuer in Prozent -1 : keine Angabe (Pago bucht keine MwSt im Debitorensystem)
SequenceNumber	n	6	Ja	Sequenznummer Muss mit „1“ beginnen und wächst mit jedem Folgeaufruf um 1 (siehe auch Abschnitt 6.3.2).
CollectNow	n	1	Ja	Einzug der bisher über MultipayInsertDebitRequest eingestellten Sätze. Je MultipayInitPaymentRequest sind mehrere Einzüge möglich. Jeder Einzug kann mehrere Debit-Sätze enthalten. true = jetzt einziehen false =nicht einziehen

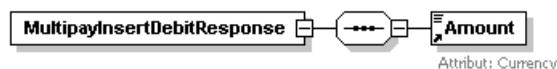
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="Thu May 17 13:42:15 2001">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
```

```

        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
      <retry>false</retry>
    </header>
    <request>
      <MultipayInsertDebitRequest>
        <ClaimSet>
          <Transaction>
            <CustomerReference>gfdgfdgfd54543453</CustomerReference>
            <TransactionType>FD</TransactionType>
            <TransactionText>Beispiel Forderung</TransactionText>
            <TransactionDate>30.12.2002</TransactionDate>
            <DebitOrCredit>S</DebitOrCredit>
            <Amount Currency="EUR" >99.99</Amount>
          <BatchId></BatchId>
        </Transaction>
        <DateOfInvoice>16.07.2001</DateOfInvoice>
        <DueDate>16.10.2001</DueDate>
        <VATAmount>13.79</VATAmount>
        <VATPercentage>16.0</VATPercentage>
      </ClaimSet>
      <SequenceNumber>1</SequenceNumber>
      <CollectNow>false</CollectNow>
    </MultipayInsertDebitRequest>
  </request>
</PAGOxml>

```

Als Rückgabewert auf **MultipayInsertDebitRequest** wird neben dem Ergebnis der einzuziehende Betrag gemeldet („0“ bei **CollectNow** = **false**).



Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
Amount				
Amount	n	10,2	Ja	Einzuziehender Betrag
Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPayment- Request

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml timeStamp="2001-10-30T16:45:25+01:00" version="2.0">
  <header>
    <communicationId>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de
    </communicationId>
    <Server> PagoServer </Server>
    <retry> false </retry>
    <inReplyTo>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url

```

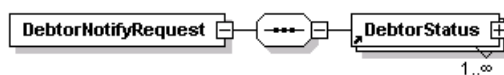


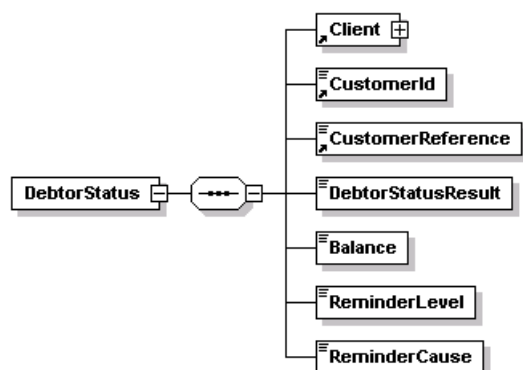
```
</inReplyTo>
</header>
<response>
  <MultipayInsertDebitResponse>
    <Amount Currency="EUR"> 0 </Amount>
  </MultipayInsertDebitResponse>
</response>
</PAGOxml>
```

Fehlercode	Grund	Abhilfe
103.001	Nicht-boole'scher Wert	true oder false verwenden
103.002	Kein gültiger numerischer Wert	Zahl verwenden
103.003	Parameterlänge zu lang oder zu kurz	Parametername steht im Twister-Log
103.500	Ungültige Mandanten-Identifikation	
103.505	Abweichende Währung	
103.506	Ungültige Transaktionsart	
103.507	Ungültige Autorisierung	
103.508	Systemfehler	
103.509	Systemfehler	
103.513	Einlösungsfehler	
103.516	Ungültige Sequenznummer	

6.5.6 Bereitstellen des aktuellen Debitorenstatus

Unabhängig von **MultipayInsertDebitRequest** kann sich der Debitorenstatus verändern. Der jeweils aktuelle Debitorenstatus wird (ohne Historie) von Pago via **Debtor-NotifyRequest** bereitgestellt. Dabei werden mit einem Request alle zur Verfügung stehenden Debitorenstatus versendet.





Tag	Daten typ	Feld- länge	Bemerkung
DebtorStatus			
Client			
ClientName	an	20	Name des Mandanten Konstante in Absprache mit Pago
SalesChannel	an	20	Vertriebslinie Konstante in Absprache mit Pago
Branch	an	20	Filiale Konstante in Absprache mit Pago
CustomerId	an	20	Ihre eindeutige Kundenreferenz unter ClientName / SalesChannel / Branch
Customer- Reference	an	20	Ihre eindeutige Vorgangsreferenz unter ClientName / SalesChannel / Branch (siehe auch Abschnitt 6.5) Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
DebtorStatus- Result	an	1	Debitorenstatus N : nicht bezahlt B : bezahlt, Vorgang ist ausgeglichen Z : überzahlt F : überfällig/in Mahnung, Vorgang ist nicht ausgeglichen (Ursache kann eine fehlge- schlagene Zahlung oder eine Rückforde- rung sein.)

Tag	Daten typ	Feld- länge	Bemerkung
Balance			
Balance	n	15,2	Saldo des Vorgangskontos Negativer Betrag: Guthaben Positiver Betrag: Forderung
Currency	an	3	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest
ReminderLevel	an	1	Aktuelle Mahnstufe 0: keine Mahnung ... 4: Inkasso A: Ausgebucht
ReminderCause	an	3	Mahngrund RL0: keine Deckung RL1: Konto erloschen RL2: Konto-Nr. / Name nicht identisch (falsch oder Sparkonto) RL3: kein Abbuchungsauftrag bzw. keine Einzugsermächtigung RL4: Rückruf RL5: Widerspruch RL6: Rückgabe LSV: Zahlart Lastschrift, nicht einziehbar CB: Chargeback KKE: Zahlart Kreditkarte, nicht einziehbar

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
<header>
  <communicationId> 2001-12-31T12:30:30+02:00-001122334455-00601@10.10.10.10
</communicationId>
  <Server>
    PagoServer
  </Server>
  <retry>
    true
  </retry>
</header>
<request>
  <DebtorNotifyRequest>
    <DebtorStatus>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
    </DebtorStatus>
  </DebtorNotifyRequest>
</request>
</PAGOxml>
```

```

    <CustomerId>123456789001</CustomerId>
    <CustomerReference>abcdefg2001</CustomerReference>
    <DebtorStatusResult>B</DebtorStatusResult>
    <Balance Currency="EUR">0</Balance>
    <ReminderLevel>0</ReminderLevel>
    <ReminderCause> </ReminderCause>
  </DebtorStatus>
  <DebtorStatus>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <CustomerId>123456789002</CustomerId>
    <CustomerReference>abcdefg1999</CustomerReference>
    <DebtorStatusResult>B</DebtorStatusResult>
    <Balance Currency="EUR">0</Balance>
    <ReminderLevel>0</ReminderLevel>
    <ReminderCause> </ReminderCause>
  </DebtorStatus>
  <DebtorStatus>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <CustomerId>123456789000</CustomerId>
    <CustomerReference>abcdefg2000</CustomerReference>
    <DebtorStatusResult>F</DebtorStatusResult>
    <Balance Currency="EUR">34,95</Balance>
    <ReminderLevel>A</ReminderLevel>
    <ReminderCause>LSV</ReminderCause>
  </DebtorStatus>
</DebtorNotifyRequest>
</request>
</PAGOfxml>

```

Auf **DebtorNotifyRequest** wird von Pago ein **DebtorNotifyResponse** erwartet. Erst mit dem Eingang eines leeren **DebtorNotifyResponse** ohne **errorNotification** gilt die Zustellung des Status als abgeschlossen.

DebtorNotifyResponse

Damit hat das Tag **DebtorNotifyResponse** keine definierten Elemente.

Tag	Daten typ	Feld- länge	Bemerkung
DebtorNotifyResponse			

Das folgende Beispiel zeigt einen möglichen **DebtorNotifyResponse**:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOfxml SYSTEM "http://xml.pago.de/2.0/PAGOfxml.dtd">
<PAGOfxml timeStamp="2001-10-30T16:45:25+01:00" version="2.0">
  <header>
    <communicationId>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url>

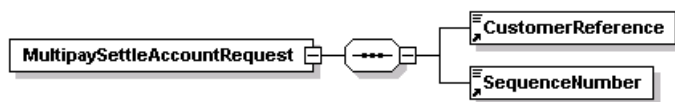
```

```
<communicationId>
  <Client>
    <ClientName>123</ClientName>
    <SalesChannel>123</SalesChannel>
    <Branch>123</Branch>
  </Client>
  <retry>false</retry>
  <inReplyTo>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de
</inReplyTo>
</header>
<response>
  <DebtorNotifyResponse/>
</response>
</PAGOfxml>
```

6.5.7 Anstoßen des Einzugs

Nach dem Einstellen von einer oder mehreren Forderungsbuchungen wird mit **MultipaySettleAccountRequest** der Einzug freigegeben. Der Einzug erfolgt in der Regel asynchron. Ergebnisse werden von Pago über **DebtorNotifyRequest** gemeldet.

Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:



Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
Customer- Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz wie in MultipayInitPaymentRequest angegeben Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
SequenceNumber	n	6	Ja	Sequenznummer Muss mit „1“ beginnen und wächst mit jedem Folgeaufruf um 1 (siehe auch Abschnitt 6.3.2).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOfxml SYSTEM "http://xml.pago.de/2.0/PAGOfxml.dtd">
```

```
<PAGOfxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>false</retry>
  </header>
  <request>
    <MultipaySettleAccountRequest>
      <CustomerReference>gfdgdfgd54543453</CustomerReference>
      <SequenceNumber>2</SequenceNumber>
    </MultipaySettleAccountRequest>
  </request>
</PAGOfxml>
```

Zu **MultipaySettleAccountRequest** wird mittels **MultipaySettleAccount-Response** der einzuziehende Betrag zurückgemeldet.



Tag	Daten typ	Feld- länge	Bemerkung
Amount			
Amount	n	10,2	Einzuziehender Betrag
Currency	an	3	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PAGOfxml SYSTEM "http://xml.pago.de/2.0/PAGOfxml.dtd">
<PAGOfxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>2001-12-31T12:34:30+02:00-001122334459-00601@www.pago.de
    </communicationId>
    <Server> PagoServer </Server>
    <retry> false </retry>
    <inReplyTo>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </inReplyTo>
  </header>
  <response>
    <MultipaySettleAccountResponse>
      <Amount Currency="EUR"> 99.99 </Amount>
    </MultipaySettleAccountResponse>
  </response>
</PAGOfxml>
```

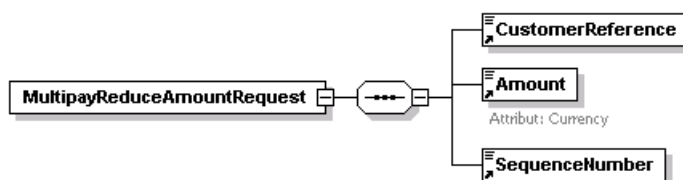
Fehlercode	Grund	Abhilfe
105.002	Kein numerischer Wert	Numerischen Wert einsetzen
105.004	Currency ist Pflicht in diesem Service	Währung einsetzen
105.500	Aus DB: MSA_UNGUELTTIGER_KLIENT	
105.501	Aus DB: MSA_INVALID_PARAMETERS	
105.502	Aus DB: MSA_UNKNOWN_TRANSACTION_IDENTIFICATION	
105.503	Aus DB: MSA_INVALID_AUTHORIZATION	
105.505	Aus DB: MSA_INVALID_SEQUENCE_NUMBER	

6.5.8 Reduzierung des reservierten Betrags

Der Request **MultipayReduceAmountRequest** informiert Pago, wenn der bei **MultipayInitPaymentRequest** angegebene Betrag reduziert wurde. Diese Reduzierung wird bei später eingestellten Debit-Sätzen berücksichtigt.

Wenn bei einem Vorgang mit Betragsreservierung (**KKE, PBX**) der Betrag in **MultipayReduceAmountRequest** auf „0“ gesetzt wird (Komplettstorno), wird die Reservierung storniert. Bei Teilstornos wird die Betragsreservierung aufrechterhalten.

Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:



Tag	Daten- typ	Feld- länge	Pflicht- feld	Bemerkung
Customer- Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz wie in MultipayInitPaymentRequest angegeben Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
Amount				
Amount	n	10,2	Ja	Neuer Gesamtbetrag des Vorgangs
Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest
SequenceNumber	n	6	Ja	Sequenznummer Muss mit „1“ beginnen und wächst mit jedem Folgeaufruf um 1 (siehe auch Abschnitt 6.3.2).

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000127-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>false</retry>
  </header>
  <request>
    <MultipayReduceAmountRequest>
      <CustomerReference>gfdgfdgfd54543453</CustomerReference>
      <Amount Currency="EUR">99.99</Amount>
      <SequenceNumber>3</SequenceNumber>
    </MultipayReduceAmountRequest>
  </request>
</PAGOxml>

```


Als Rückgabewert auf **MultipayReduceAmountRequest** wird der verbleibende Reservierungsbetrag gemeldet.



Tag	Datentyp	Feldlänge	Pflichtfeld	Bemerkung
Amount				
Amount	n	10,2	Ja	Verbleibender Reservierungsbetrag
Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>2001-12-31T12:35:30+02:00-001122334460-00601@www.pago.de
  </communicationId>
    <Server> PagoServer </Server>
    <retry> false </retry>
    <inReplyTo>2001-12-31T12:02:30+02:00-000000000127-00001@www.client.url </inReplyTo>
  </header>
  <response>
    <MultipayReduceAmountResponse>
      <Amount Currency="EUR"> 0 </Amount>
    </MultipayReduceAmountResponse>
  </response>
</PAGOxml>
```

Fehlercode	Grund	Abhilfe
104.002	Wert muss numerisch sein	Zahl einsetzen
104.004	Currency ist Pflicht für diesen Dienst	Währung angeben
104.500	Aus DB: MPSA_UNKNOWN_CLIENT	
104.504	Währung weicht von InitPayment ab	Gleiche Währung wie bei InitPayment verwenden

Fehlercode	Grund	Abhilfe
104.505	Ungültige Währung	Gleiche Währung wie bei InitPayment verwenden
104.506	Ungültige Autorisierung	InitPayment wurde nicht erfolgreich beendet
104.509	Ungültige Sequenznummer	

7 Multipay mit Modus 1

In diesem Kapitel wird die Handhabung des Modus 1 beschrieben. Detaillierte Informationen für die Auswahl des benötigten Modus finden Sie in Kapitel 2 „Entscheidungshilfen für die angemessene Payment-Methode“.

Im Modus 1 wird der Betrag aufgrund Ihrer Forderung sofort autorisiert und vollständig eingezogen. Der Modus 1 unterstützt die Zahlarten „Kreditkarte“ und „Lastschrift“.



Für Paybox müssen Sie trotz der Charakteristik des Netzbetriebs den Modus 0 verwenden.

7.1 Anwendungsregeln für den Einsatz von Multipay im Modus 1

Die gewünschten Zahlarten müssen Sie mit Pago absprechen. Gemäß dieser Absprache richtet Pago Bankkonten und Kreditkarten-Parameter ein und teilt sie, soweit notwendig, dem aufrufenden System mit.

Ein Auftrag wird mit dem Ziel eingestellt, einen Geldbetrag in einer vorher bekannten Maximalhöhe vom Kunden zu erhalten. Beim Anlegen des Auftrags wird dieser Gesamtbetrag an Pago übergeben. Für den Betrag gelten die drei folgenden Regeln:

- (§1) Die Summe aller Teilbeträge pro Auftrag darf den Gesamtbetrag nicht überschreiten.
- (§2) Der Gesamtbetrag darf nicht nachträglich erhöht werden.
- (§3) Die Summe aller Gutschriftenbeträge darf die Summe aller Forderungsbeträge nicht überschreiten.

Nicht nur die Forderungen, die aus der Bestellung des Kunden hervorgehen, können storniert werden, sondern auch Gutschriften, die z. B. bei Beschädigung der Ware oder Retoure ausgestellt werden.

Darüber hinaus wird zwischen Buchungen und Stornos unterschieden. Forderungsbuchungen und Gutschriftenstornos werden in positives Soll gebucht, Forderungsstornos und Gutschriftenbuchungen in negatives Soll. Dabei gelten folgende Regeln:

- (§4) Die Betragssumme der Stornos darf pro Satzart die Betragssumme der Buchungen nicht überschreiten.
- (§5) Der Betrag aller Forderungsbuchungen darf den Gesamtbetrag nicht überschreiten.

! Ein einmal stornierter Betrag (Forderungsstorno, Retoure, Versandkostenstorno) kann nicht mehr gefordert werden.
Eine Gutschrift (aus Kulanz) kann nur auf existierende Zahlungseingänge gewährt werden. Eine Gutschrift kann auch wieder storniert werden, die Regeln in §1 bis §5 sind jedoch immer verbindlich.

! Je nach Zahlart gibt es weitere Einschränkungen bezüglich Gutschriften und Stornos. Deshalb sollten Sie Forderungen möglichst zu einem Zeitpunkt einstellen, an dem Sie prozessbedingte Rückabwicklungen abschließen können.

7.2 Umgang mit Retouren und Gutschriften

Mit folgender Vorgehensweise bleiben die Warenbewegungen und Zahlungen je Vorgang bei allen Parteien übersichtlich und nachvollziehbar. Rückbelastungen aufgrund unklarer Abbuchungen des Kunden können reduziert werden.

- Situation 1: Dem Kunden wird nach einer Warenretoure Ersatz mit gleichem Wert geliefert:
Diese Situation ist für den Vorgang in Multipay nicht relevant.
- Situation 2: Dem Kunden wird nach einer Warenretoure Ersatz mit geringerem Wert geliefert. Zum Ausgleich soll er eine Gutschrift erhalten:
Die Gutschrift über den gewählten Betrag wird mittels **MultipayInsertDebitRequest** angelegt und, je nach Bedarf, entweder sofort oder mit **MultipaySettleAccount** zur Auszahlung gebracht.
- Situation 3: Dem Kunden wird nach einer Warenretoure der gezahlte Betrag erstattet. Zu einem späteren Zeitpunkt wird erneut eine Ware an ihn gesandt:
Mit der Warenretoure sollte der zugehörige Vorgang geschlossen werden. Dazu wird mit **MultipayInsertDebitRequest** eine Retoure eingestellt. Der Gutschriftsbetrag wird nach Bedarf sofort oder mit dem nächsten **MultipaySettleAccount** ausgezahlt.
Für die nächste Bestellung wird mit **MultipayInitPayment** ein neuer Vorgang angelegt.

7.3 Transaktionssicherheit

7.3.1 Aufrufwiederholung bei Verbindungsabbruch

In der Verbindung zwischen dem Mandantensystem und Pago kann es vorkommen, dass der Aufrufende kein Ergebnis erhält. Das kann u. a. aus zwei Gründen der Fall sein:

- Der Empfänger hat den Aufruf nicht erhalten.
- Der Empfänger hat den Aufruf erhalten und verarbeitet, aber die Rückmeldung ist nicht beim Absender angekommen.

Um solche Vorfälle zu vermeiden, besitzt die Anbindung über LocalServer und XML einen speziellen Retry-Mechanismus. Dieser Mechanismus identifiziert jede Kommunikation einer Transaktion eindeutig mit Hilfe der **communicationId**.

Wird der Response nicht innerhalb einer definierten Zeitspanne empfangen, die über die Konfigurationsvariable **Timeout** definiert wird, sollte der Request mit **retry = true** und der gleichen **communicationId** erneut gesendet werden.

Mit dem Retry können drei Situationen eintreten:

- Wenn Pago den Request vor dem Retry nicht erhalten hat, wird der Request nach dem Retry von Pago verarbeitet und der entsprechende Response zurückgesendet.
- Wenn Pago den Request vor dem Retry erhalten und auch verarbeitet hat, wird nach dem Retry das Ergebnis dieser Verarbeitung im Response zurückgesendet.
- Wenn Pago den Request vor dem Retry erhalten hat und die Verarbeitung noch andauert, wird nach dem Retry im Response die Meldung zurückgesendet: **this transaction is still in progress**. Dann muss der Request mit einem weiteren Retry wiederholt werden.

Das folgende Beispiel zeigt den Response im Falle der noch andauernden Verarbeitung. Nähere Informationen hierzu finden Sie in Abschnitt 5.2 „Header-Block“.

```
</header>
<response>
  <errorNotification type="recoverable" issuer="PagoServer"
    service="MultipayInitGuiPayment">
    <errorText> <![CDATA[this transaction is still in progress]]> </errorText>
    <errorNumber> 30007 </errorNumber>
  </errorNotification>
</response>
</PAGOxml>
```

Wenn das Ergebnis der Verarbeitung vorliegt, kann es in eine der folgenden Klassen eingestuft werden:

- 1) Erfolgreiche Durchführung
- 2) Die Transaktion wurde mit nicht behebbaren Fehlern durchgeführt (**fatal**). Eine Wiederholung wird kein anderes Ergebnis hervorbringen. Daher sollten keine Wiederholungsaufrufe durchgeführt werden.
- 3) Die Transaktion wurde mit einem Fehler durchgeführt, der zu einem späterem Zeitpunkt möglicherweise nicht mehr auftritt. Zur Kennzeichnung des möglichen späteren Wiederholungsversuchs wird der Fehler mit **recoverable** klassifiziert.

Wenn in einem Fehlerszenario die Bank bei einer Autorisierung nicht erreicht wird, kann der Betrag auch nicht bei der betreffenden Kreditkarte reserviert werden. Die Initialisierung ist nicht abgeschlossen. Die Transaktion wird mit dem negativen Ergebnis **recoverable** abgeschlossen.

Die Autorisierung kann zu einem späteren Zeitpunkt erneut ausgeführt werden. Der erneute Autorisierungsversuch wird mit alter **communicationId** und gleicher Fremdbelegnummer (**CustomerReference**) initialisiert.

Die Aufrufwiederholung ist mehrfach ausführbar. Wenn Sie also nach einem Aufruf kein Ergebnis zurückerhalten haben, müssen Sie im Zweifelsfall den Aufruf solange wiederholen, bis Sie Klarheit über das Ergebnis haben. Wählen Sie dabei bitte einen sinnvollen Zeitabstand zwischen den Wiederholungsanfragen. Einen Funktionsaufruf, der wegen Timeout nicht abgeschlossen werden konnte, können Sie sofort wiederholen. Wenn eines der angebundenen Systeme (z. B. ein Banksystem) nicht zur Verfügung steht, wiederholen Sie den Aufruf nach einigen Minuten.

7.3.2 Handhabung der Sequenznummer

Die Sequenznummer dient der Einhaltung der Verarbeitungsreihenfolge. Pro Fremdbelegnummer muss die Sequenznummer aufsteigend und lückenlos nummeriert werden. Die Auftragsinitialisierung trägt die implizite Sequenznummer „0“. Verwenden Sie deshalb bei der als nächstes aufgerufenen Methode die Sequenznummer „1“. Eine falsche Sequenznummer wird mit einem Fehler abgelehnt. Für weitere Informationen zu Fehlermeldungen siehe auch Abschnitt 5.5 „Fehlermanagement“.

7.3.3 Verhalten der Methoden bei Wiederholungsaufrufen

Nach einem Wiederholungsaufruf mit gesetztem Retry-Flag wird das Ergebnis der ersten erfolgreichen bzw. fatalen Verarbeitung im Response zurückgeliefert. Die folgende Angabe gilt für Transaktionen, die durch das Flag **retryable** wiederholbar sind:

```
<errorNotification issuer="PagoServer"
    service="MultipayInitPayment" type="retryable">
```

Die Funktionen aus Pago Multipay verhalten sich folgendermaßen:

MultipayInitPaymentRequest wird identifiziert durch

- **ClientName**
- **SalesChannel**
- **Branch**
- **CustomerReference**.

Wiederholungsanfragen zu fehlgeschlagenen oder abgelehnten Zahlungen werden erneut durchgeführt. Wiederholungsanfragen zu akzeptierten Zahlungen liefern das vorher erhaltene Ergebnis zurück, wenn sie in den anderen Parametern übereinstimmen. Andernfalls werden sie abgelehnt.

MultipayInsertDebitRequest und **MultipaySettleAccountRequest** werden identifiziert durch

- **ClientName**
- **SalesChannel**
- **Branch**
- **CustomerReference**
- **SequenceNumber**.

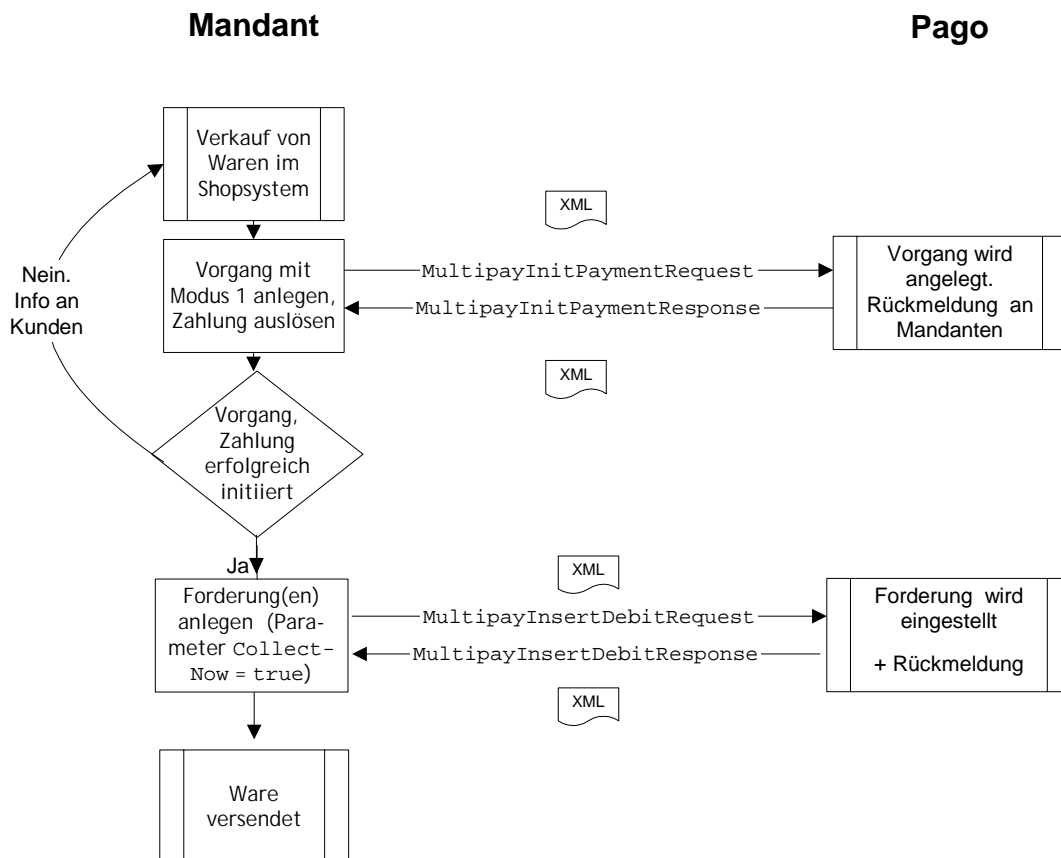
Wiederholungsanfragen liefern das vorher erhaltene Ergebnis zurück. Anfragen mit falscher Sequenznummer werden abgelehnt.

7.4 Anwendungsbeispiele für Multipay mit Modus 1

Bei Modus 1 wird neben der Autorisierung auch die sofortige Zahlung durchgeführt. Zusätzlich wird eine Forderung über den Gesamtbetrag bei Pago verwaltet. Dadurch bleibt die Möglichkeit der debitorischen Verwaltung erhalten. Gutschriften und aufgrund Rückbelastung notwendiges Mahnwesen können über Pago abgewickelt werden.

Die Beispiele dienen der Veranschaulichung Ihrer Möglichkeiten und bieten außerdem eine Diskussionsgrundlage für die Unterstützung durch Pago Consulting.

Im folgenden Beispiel besteht bereits zum Zeitpunkt der Auftragseinstellung eine Forderung gegen den Kunden (z. B. Download einer Software) in Höhe des Gesamtbetrages, und der Auftrag wird über eine Online-Zahlart beglichen. So kann auch der Einzug mit der zugehörigen Zahlungsbuchung zu diesem Zeitpunkt initiiert werden.



Autorisierung und Zahlung erfolgen in einem Schritt. Forderungen werden eingestellt.

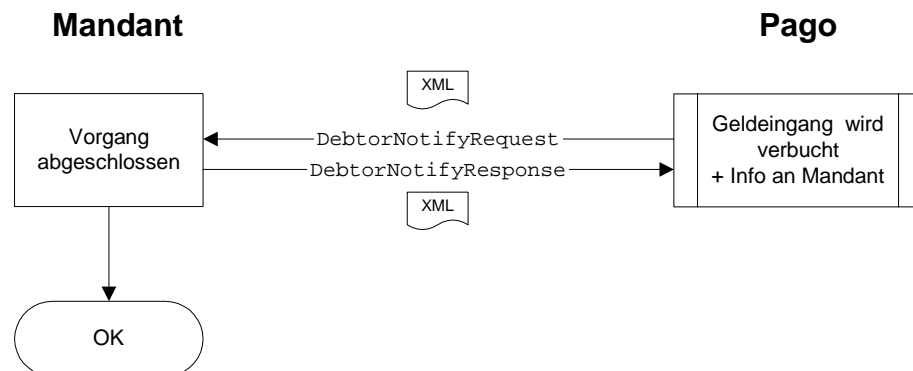
! *Der Modus 1 ist wie eine Vorkasse, da sofort eine Belastung erfolgt. Denn beim Modus 1 wird mit der Initialisierung sofort die Zahlung ausgelöst.*

*Beim Einstellen der Forderung mittels **MultipayInsertDebitRequest** muss der Parameter **CollectNow** unbedingt **false** sein.*

*Wenn **MultipayInsertDebitRequest** mit dem Parameter **Collect-Now = true** oder **MultipaySettleAccountRequest** verwendet wird, wird der Saldo der Forderungen ausgeglichen, bzw. führt in bestimmten Fällen zu Auswirkungen auf einzuziehende oder auszahlende Werte. Überschüssige Forderungen führen zum Einzug, unterschüssige Forderungen zu einer Gutschrift.*

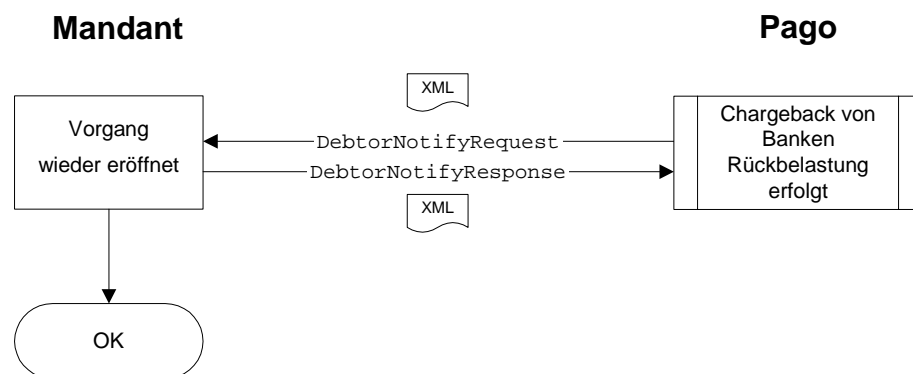
! *Gutschriften oder Stornos müssen mittels **MultipayInsertDebit-Request** mit dem Parameter **CollectNow = true** oder **CollectNow = false** und nachgelagertem **MultipaySettleAccountRequest** zur Auszahlung gebracht werden.*

Der Debitorenstatus gibt Auskunft über den Geldeingang.

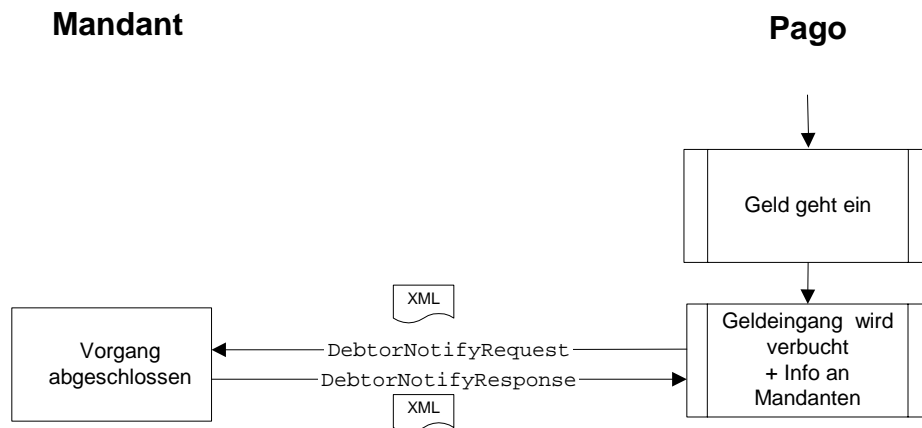


7.4.1 Meldung nach Chargeback

Statusinformationen aus dem Debitorenmanagement können auch nach dem Abschluss eines Vorgangs auftreten. In diesem Beispiel nehmen wir ein Chargeback auf eine Kreditkartenzahlung oder eine Rücklastschrift auf eine Einzugsermächtigung an. Der Debitorenstatus wird mittels **DebtorNotifyRequest** bereitgestellt.



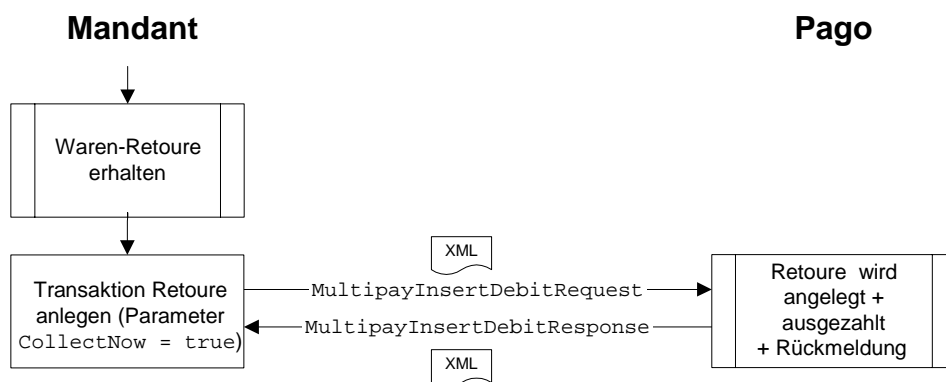
Das Mahnwesen wird in diesem Fall über Pago abgewickelt. Der Kunde reagiert auf die Mahnung und zahlt. Der Geldeingang wird über den Debitorenstatus mitgeteilt.



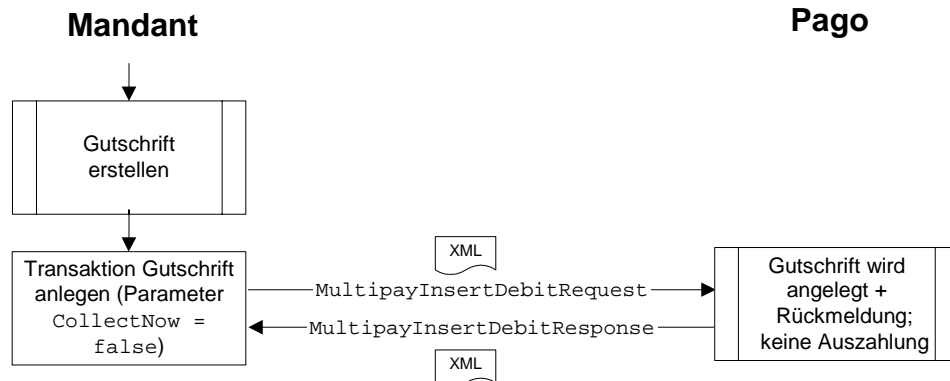
7.4.2 Gutschrift

Gutschriften werden analog zu einer Forderung mit **MultipayInsertDebitRequest** eingestellt. (Achten Sie hierbei auf die Regeln aus Abschnitt 7.1 „Anwendungsregeln für den Einsatz von Multipay im Modus 1“).

Dieses Beispiel zeigt den möglichen Ablauf für die Verarbeitung einer Retoure und einer Gutschrift. Die Retoure wird mit sofortiger Auszahlung angelegt.



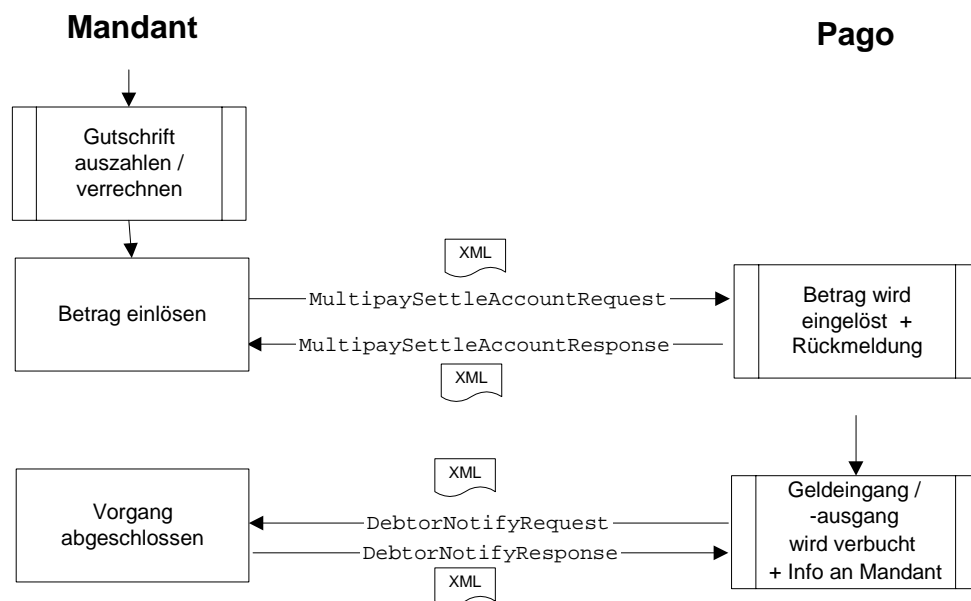
Wenn eine Gutschrift nicht ausgezahlt, sondern mit Forderungen verrechnet werden soll, verwenden Sie den Parameter **CollectNow = false**. Die Verrechnung erfolgt mit dem nächsten **MultipaySettleAccountRequest**.



7.4.3 Forderungskonto ausgleichen

Nach dem Aufruf der Funktion **MultipaySettleAccountRequest** wird die Gutschrift ausgeführt. Wenn in der Zwischenzeit eine neue Forderung eingestellt wird, wird diese mit der Gutschrift verrechnet und nur noch der Saldo ausgezahlt.

! *Durch eine Gutschrift wird eine Forderung nicht aufgehoben. Die Summe der Forderungen muss weiterhin kleiner oder gleich dem Betrag des **MultipayInitPaymentRequest** sein.*



7.5 Technische Schnittstellenbeschreibung

Der Service Pago Multipay wird integriert durch das Senden und Empfangen von XML-Dokumenten via LocalServer. Der Aufbau der benötigten XML-Dokumente wird nachfolgend beschrieben.

Die Pago-Multipay-Services im Modus 1 werden über folgende XML-Requests bereitgestellt:

- **MultipayCheckBankRequest** zum Prüfen einer Bankverbindung
- **MultipayCheckCreditCardRequest** zum Prüfen einer Kreditkarte
- **MultipayInitPaymentRequest** zum Initiieren von Zahlungen
- **MultipayInsertDebitRequest** zum Einstellen von Forderungsbuchungen
- **MultipaySettleAccountRequest** zum Anstoßen des Einzugs.

Die Fremdbelegnummer bezieht sich auf einen Vorgang im Mandantensystem, der über Pago Multipay abgewickelt wird (z. B. ein Kauf). Sie muss dem entsprechenden Vorgang beim Mandanten eindeutig zugeordnet werden können. Für die Fremdbelegnummer sollten ausschließlich Ziffern und Großbuchstaben verwendet werden, da das System Kleinbuchstaben automatisch in Großbuchstaben umsetzt. Falls zwei Fremdbelegnummern angelegt werden, die sich nur durch Groß- und Kleinschreibung voneinander unterscheiden, können sie vom System nicht mehr eindeutig zugeordnet werden.

! Halten Sie sich bei Datumseingaben genau an das vorgegebene Format (YYYYMMDD oder DD.MM.YYYY). Lassen Sie keine Stelle aus, auch nicht die führende „0“ (z. B.: „01.11.2000“).

Legende

Die Parameter des Request und des Response sind in Tabellen dargestellt. Für die Tabellen gilt folgende Legende:

Tag: Feldbezeichnung
 Datentyp: „an“ = alphanumerisch / „n“ = numerisch
 Feldlänge: Anzahl der Zeichen
 Pflichtfeld: Angabe notwendig (Ja/Nein)



Bei numerischen Feldern, in denen eine Zahl mit Nachkommastellen eingetragen werden kann, sind unter „Feldlänge“ zwei Zahlen angegeben. Die Zahl vor dem Komma gibt die maximale Anzahl der gesamten Stellen an, die Zahl hinter dem Komma die maximale Anzahl der Nachkommastellen.

Beispiel: **10,2** --> max. 10 Stellen, davon max. 2 Nachkommastellen.

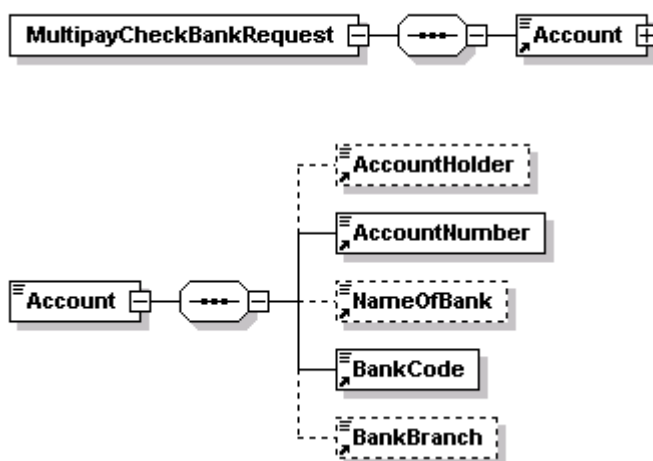
7.5.1 Prüfen einer Bankverbindung

Eine Bankverbindung wird mit **MultipayCheckBankRequest** überprüft. Hierzu dienen

- die Bankleitzahl
- die Kontonummer
- das jeweilige Prüfziffernverfahren.

Die Prüfung der Bankverbindung sollte vor dem Eintrag mittels **MultipayInitPaymentRequest** erfolgen, um klare Ausfälle auszuschließen.

Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:

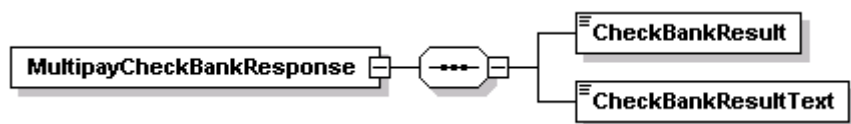


Tag	Daten typ	Feld-länge	Pflicht-feld	Bemerkung
Account				
Account-Holder	an	30	Nein	Kontoinhaber (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)
Account-Number	an	10	Ja	Kontonummer (nur bei Zahlart ELV)
NameOfBank	an	30	Nein	Bankname (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)
BankCode	an	8	Ja	Bankleitzahl (nur bei Zahlart ELV)
BankBranch	an	30	Nein	Bankfiliale (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)

Beispiel

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM „http://xml.pago.de/2.0/PAGOxml.dtd“>
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:00:30+02:00-000000000123-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>false</retry>
  </header>
  <request>
    <MultipayCheckBankRequest>
      <Account>
        <AccountNumber>1234567890</AccountNumber>
        <BankCode>87654321</BankCode>
      </Account>
    </MultipayCheckBankRequest>
  </request>
</PAGOxml>
```



```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM „http://xml.pago.de/2.0/PAGOxml.dtd“>
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>2001-12-31T12:30:30+02:00-001122334455-00601@www.pago.de
    </communicationId>
    <Server>PagoServer</Server>
    <retry>false</retry>
    <inReplyTo>2001-12-31T12:00:30+02:00-000000000123-00001@www.client.url
    </inReplyTo>
  </header>
  <response>
    <MultipayCheckBankResponse>
      <CheckBankResult>0</CheckBankResult>
      <CheckBankResultText>ACCOUNT VALID</CheckBankResultText>
    </MultipayCheckBankResponse>
  </response>
</PAGOxml>
```

Fehlercode	Grund	Abhilfe
106.500	Aus DB: MCB-KLIENT_UNGUELTIG	

7.5.2 Prüfen einer Kreditkarte

Die Kreditkarte wird mit **MultipayCheckCreditCardRequest** überprüft. Dies geschieht nach folgenden Kriterien:

- Kreditkarten-Anfangsziffern
- Kreditkartengesellschaft
- Prüfziffer der Kreditkartennummer
- Verfalldatum der Kreditkarte.

Prüfen Sie die Kreditkarte vor dem Eintrag mittels **MultipayInitPaymentRequest**, um klare Ausfälle auszuschließen.

MultipayCheckCreditCard liefert Ihnen zwei Ergebnisfelder:

- **CheckCreditCardResult**
- **CheckCreditCardResultText**.

CheckCredit-CardResult	CheckCreditCardResultText		Ergebnisklasse
0	Kreditkarte existiert		OK
-1	Kreditkarte abgelaufen		fatal
-2	Falsche Kartennummernlänge		fatal
-3	Ungültige Eingabezeichen		fatal
-4	Keine numerische Kartennummer		fatal
-5	Unbekannter Kreditkartentyp		fatal
-6	Ungültige Checksumme		fatal
-7	Falsches Datumsformat		fatal
-8	Ablehnungsgrund der Prüfstelle (nur bei Online-Prüfung) Original-Fehlertext des Netzbetreibers der Kreditkarte wird ausgegeben. Liste der Fehlercodes:		
	02	Anrufen der AS-Rufnummer	fatal
	03	Nummer des Vertragsunternehmens unbekannt	fatal
	04	Karte einbehalten	fatal
	05	Autorisierung abgelehnt	fatal
	06	Filetransfer falsch, Wiederholung ab mitgeteilter Blocknummer möglich	fatal
	09	Verzögerte Bearbeitung der Autorisierung	fatal
	12	Ungültige Transaktion	fatal

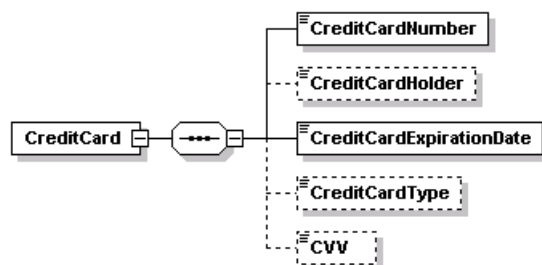
CheckCredit-CardResult	CheckCreditCardResultText		Ergebnisklasse
	13	Verfügbarer Betrag überschritten	fatal
	14	Kartenummer nicht bekannt	fatal
	21	Kein Vorgang vorgenommen	fatal
	24	Filetransfer nicht unterstützt	fatal
	29	Filetransfer nicht erfolgreich	fatal
	30	Formatfehler	fatal
	31	Kartenherausgeber nicht zugelassen	fatal
	33	Verfalldatum überschritten	fatal
	34	Manipulationsverdacht	fatal
	40	Funktion ungültig	fatal
	43	Gestohlene Karte; Karte einbehalten	fatal
	50	Autorisierung doppelt	fatal
	51	Limit überschritten; „Doch“-Funktion möglich	fatal
	54	EC-Chipkarte nicht mehr gültig	fatal
	55	PIN falsch	fatal
	56	Karte nicht im Bestand	fatal
	57	Andere Karte als bei der Bezugstransaktion benutzt wurde (bei Storno; Kreditkarten auch bei Erweiterungsreservierung, Endsummenbuchung, Trinkgeld-Update)	fatal
	58	Terminal-ID nicht erkannt	fatal
	59	Manipulationsverdacht bei EC-Chipkarte	fatal
	61	Karte gesperrt in lokaler Sperrliste	fatal
	62	Karte gesperrt	fatal
	64	Transaktionsbetrag weicht von Bezugstransaktion ab (bei Storno; Kreditkarten auch bei Endsummenbuchung)	fatal
	65	Limit der Transaktionsfrequenz überschritten; „Doch“-Funktion möglich	fatal
	75	Zulässige Anzahl von PIN-Eingaben überschritten	fatal
	76	Schlüsselindex unzulässig bzw. Fehler in BMP 57	fatal

CheckCredit-CardResult	CheckCreditCardResultText		Ergebnisklasse
	77	PIN-Eingabe erforderlich	fatal
	78	Sequenzfehler in BMP 62	fatal
	80	Betrag ist nicht verfügbar	fatal
	81	Initialisierung fehlerhaft; Wiederholung erforderlich	fatal
	82	(Vor-)Initialisierung unzulässig (Terminal gesperrt)	fatal
	83	PIN-Pad-Wechsel nicht möglich	fatal
	84	Datex-P-Nr. falsch	fatal
	85	Ablehnung vom Kreditkarteninstitut	fatal
	86	Stammdaten unbekannt; bei Init1 OPT-fähiger Terminals auch: ZKA-Nummer falsch	fatal
	87	Terminal (Pin-Pad) unbekannt	fatal
	88	Beim Laden gegen EC-Karte: PIN nicht aktiv	fatal
	89	CRC falsch	fatal
	91	Kartenherausgeber oder Netz nicht verfügbar	fatal
	92	Kartentyp wird nicht vom Autorisierungssystem verarbeitet	fatal
	96	Verarbeitung zur Zeit nicht möglich	fatal
	97	Sicherheitsverletzung: MAC-Prüfung zeigt Fehlerzustand	fatal
	98	Datum / Uhrzeit nicht plausibel	fatal
	99	Fehler in der PAC-Verschlüsselung	fatal
99 und 999	Nicht klassifizierte technische Fehler im Payment		recoverable



Die Liste der Original-Fehlertexte des Netzbetreibers hat keinen Anspruch auf Vollständigkeit.





Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:

Tag	Daten typ	Feld-länge	Pflicht-feld	Bemerkung
CreditCard				
CreditCard-Number	an	19	Ja	Kreditkartennummer
CreditCard-Holder	an		Nein	(Reserviert für weitere Funktionen)
CreditCard-Expiration-Date	an	6	Ja	Verfalldatum Format: MM/YY
CreditCard-Type	an	1	Ja	Kreditkartentyp V: Visa A: AMEX E: EUROCARD/MasterCard D: Diners J: JCB G: Galeria
CVV	n	4	Nein	Kreditkartenprüfnummer Ziffernfolge mit max. vier Stellen (nähere Informationen siehe Abschnitt 1.3)

```

<?xml version="1.0" encoding="ISO-8859-1"?>

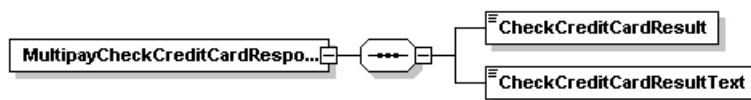
<!DOCTYPE PAGOxml SYSTEM „http://xml.pago.de/2.0/PAGOxml.dtd“>
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000124-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>true</retry>
  
```

```

</header>
<request>
  <MultipayCheckCreditCardRequest>
    <CreditCard>
      <CreditCardNumber>49494949494949</CreditCardNumber>
      <CreditCardExpirationDate>12/02
      </CreditCardExpirationDate>
      <CreditCardType>V</CreditCardType>
    </CreditCard>
  </MultipayCheckCreditCardRequest>
</request>
</PAGOxml>

```

MultipayCheckCreditCardResponse ist wie folgt aufgebaut:



Tag	Daten typ	Feld- länge	Bemerkung
CheckCredit- CardResult	n	5	Ergebnis als Nummer (siehe Informationen weiter oben)
CheckCredit- CardResultText	an	40	Ergebnis als Text

```

<?xml version="1.0" encoding="ISO-8859-1"?>

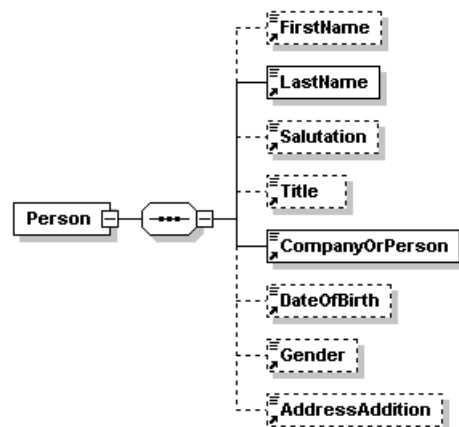
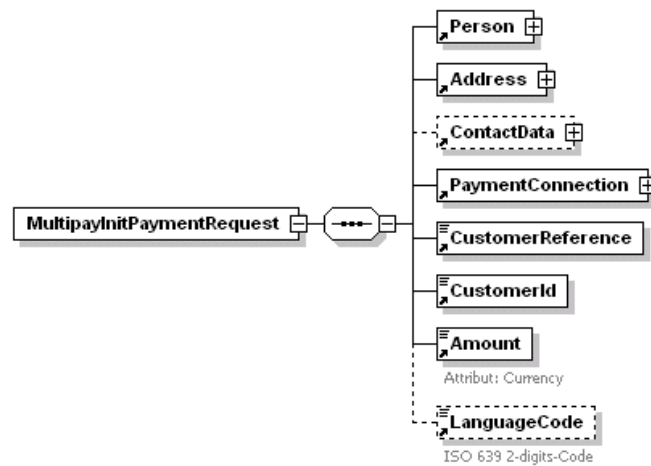
<!DOCTYPE PAGOxml SYSTEM „http://xml.pago.de/2.0/PAGOxml.dtd“>
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
<header>
<communicationId>2001-12-31T12:31:30+02:00-001122334456-00601@www.pago.de
</communicationId>
<Server>PagoServer</Server>
<retry>>false</retry>
<inReplyTo>2001-12-31T12:02:30+02:00-000000000124-00001@www.client.url
</inReplyTo>
</header>
<response>
<MultipayCheckCreditCardResponse>
  <CheckCreditCardResult>-6</CheckCreditCardResult>
  <CheckCreditCardResultText>INVALID CHECKSUM
  </CheckCreditCardResultText>
</MultipayCheckCreditCardResponse>
</response>
</PAGOxml>

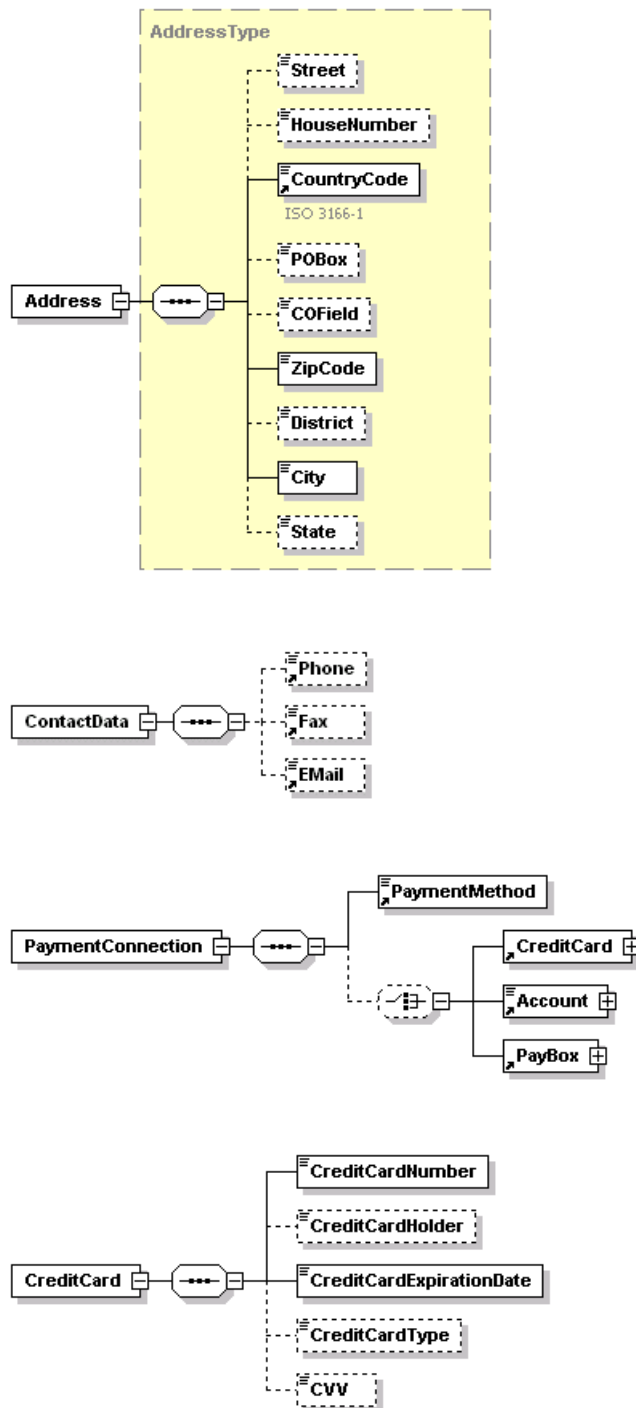
```

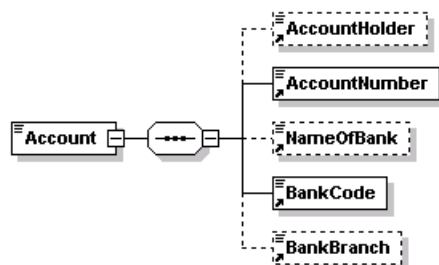
7.5.3 Initiiieren von Zahlungen

Zahlungen werden mit **MultipayInitPaymentRequest** initiiert. Der Vorgang wird durch eine Fremdbelegnummer eindeutig gekennzeichnet.

Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:







Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
Mode	n	3	Ja	Modus 1: Anlegen des Vorgangs bei allen Zahlarten und Prüfung/Autorisie- rung bei ELV , KKE mit späterem Geldfluss Weitere Informationen zum Modus und dem damit verbunde- nen Businessmodell finden Sie in Kapitel 3.
Person				
FirstName	an	30	Nein	Vorname
LastName	an	30	Ja	Nachname oder Firmenname
Salutation	an	10	Nein	Anrede, z. B. „Herr“, „Frau“, „Firma“
Title	an	20	Nein	Titel, z. B. „Dr.“, „Prof.“
CompanyOr- Person	char	1	Ja	F (Firma) oder P (Person)
DateOfBirth	an	10	Nein	Geburtsdatum Format: YYYYMMDD oder DD.MM.YYYY
Gender	char	1	Nein	Geschlecht Fixwert: „ “ (Leerstring)
Address- Addition	an	60	Nein	Adresszusatz z. B. „7. Stock“, „bei Maier“, „Firma Müller“, z. H. Herrn Schmidt“
Adresse				
Street	an	46	Ja	Straßenname
HouseNumber	an	35	Ja	Hausnummer

Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
CountryCode	an	3	Ja	gültiges Länderkennzeichen nach ISO 3166 (siehe Kapitel 10)
POBox	n	35	Nein	Postfach-Nummer
COField	an	40	Nein	Großempfänger-Name
ZipCode	an	9	Ja	Postleitzahl
District	an	40	Nein	Ortsteil
City	an	40	Ja	Ortsname
State	an	40	Nein	Staat (US-Bundesstaat)
ContactData				
Phone	an	20	Nein	Telefonnummer laut Pago Bill Muss mit „+“ und Ländervorwahl beginnen (z. B. „+49“).
Fax	an	20	Nein	Faxnummer laut Pago Bill Muss mit „+“ und Ländervorwahl beginnen (z. B. „+49“).
E-Mail	an	80	Nein	E-Mail-Adresse
PaymentConnection				
PaymentMethod	an	3	Ja	Mit Pago abgesprochene Zahlart, z. B.: ELV Lastschriftverfahren (DTA) KKE Kreditkarteneinzug
CreditCard (nur bei KKE)				
CreditCard- Number	an	19	Ja	Kreditkartennummer
CreditCard- Holder	an		Nein	(Reserviert für weitere Funktio- nen)
CreditCard- Expiration- Date	an	6	Ja	Verfalldatum Format: MM/YY
CreditCard- Type	an	1	Ja	Kreditkartentyp V: VISA A: AMEX E: EUROCARD/MasterCard D: DINERS J: JCB G: GALERIA

Tag	Datentyp	Feldlänge	Pflichtfeld	Bemerkung
CVV	n	4	Nein	Kreditkartenprüfnummer Ziffernfolge mit max. vier Stellen (nähere Informationen siehe Abschnitt 1.3)
Account (nur bei ELV)				
Account-Holder	an	30	Nein	Kontoinhaber (nur bei Zahlart ELV) Fixwert: „“ (Leerstring)
Account-Number	an	10	Ja	Kontonummer (nur bei Zahlart ELV)
NameOfBank	an	30	Nein	Bankname (nur bei Zahlart ELV) Fixwert: „“ (Leerstring)
BankCode	an	8	Ja	Bankleitzahl (nur bei Zahlart ELV)
BankBranch	an	30	Nein	Bankfiliale (nur bei Zahlart ELV) Fixwert: „“ (Leerstring)
Customer-Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz unter ClientName / Sales-Channel / Branch (siehe auch Abschnitt 7.5) Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
CustomerId	an	20	Ja	Ihre eindeutige Kundenreferenz unter ClientName / Sales-Channel / Branch
Amount				
Amount	n	10,2	Ja	positiver Bruttobetrag inkl. MwSt
Currency	an	3	Ja	Währung gemäß ISO 4217 (z. B. EUR, USD) Currency ist ein Attribut von Amount
LanguageCode	an	3	Nein	Sprachkennzeichen des Kunden Verwendung zukünftig im Mahnwesen

Details über die Verwendung der Adressfelder bei Rechnungsdruck und Mahnwesen finden Sie in dem Dokument „PrintLayout“.



Wiederholte Aufrufe zur selben Fremdbelegnummer erhalten das gleiche alte Ergebnis (Returncode, Message).

Beispiel:

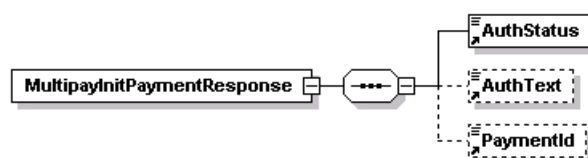
```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE PAGOxml SYSTEM „http://xml.pago.de/2.0/PAGOxml.dtd“>
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000125-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>true</retry>
  </header>
  <request>
    <MultipayInitPaymentRequest Mode="1">
      <Person>
        <FirstName>HANS</FirstName>
        <LastName>Debtor</LastName>
        <Salutation>Herr</Salutation>
        <CompanyOrPerson>P</CompanyOrPerson>
      </Person>
      <Address>
        <Street>reperbaan</Street>
        <HouseNumber>1</HouseNumber>
        <CountryCode>De</CountryCode>
        <ZipCode>2000</ZipCode>
        <City>hammburg</City>
      </Address>
      <PaymentConnection>
        <PaymentMethod>KKE</PaymentMethod>
        <CreditCard>
          <CreditCardNumber>4321432143214321</CreditCardNumber>
          <CreditCardExpirationDate>12/01</CreditCardExpirationDate>
        </CreditCard>
      </PaymentConnection>
      <CustomerReference>987654323</CustomerReference>
      <CustomerId>13447</CustomerId>
      <Amount Currency="EUR">10.0</Amount>
    </MultipayInitPaymentRequest>
  </request>
</PAGOxml>
```

Für die Zahlung auf Rechnung wird nur **PaymentMethod** angegeben. Das folgende Beispiel soll diesen Sachverhalt verdeutlichen.

```
<request>
```

```
<MultipayInitPaymentRequest Mode="1">
  <Person>
    <FirstName>HANS</FirstName>
    <LastName>Debtor</LastName>
    <Salutation>Herr</Salutation>
    <CompanyOrPerson>P</CompanyOrPerson>
  </Person>
  <Address>
    <Street>reperbaan</Street>
    <HouseNumber>1</HouseNumber>
    <CountryCode>De</CountryCode>
    <ZipCode>2000</ZipCode>
    <City>hammburg</City>
  </Address>
  <PaymentConnection>
    <PaymentMethod>REC</PaymentMethod>
  </PaymentConnection>
  <CustomerReference>987654323</CustomerReference>
  <CustomerId>13447</CustomerId>
  <Amount Currency="EUR">10.0</Amount>
</MultipayInitPaymentRequest>
</request>
```

Die Antwort auf **MultipayInitPaymentRequest** ist **MultipayInitPaymentResponse**. Darin wird der Autorisierungsstatus mit einem Zusatztext geliefert.



Der **MultipayInitPaymentResponse** hat folgenden Aufbau:

Tag	Daten typ	Feld- länge	Bemerkung
MultipayInitPaymentResponse			
AuthStatus	n	1	Autorisierungsstatus 0: OK 1: abgelehnt (nur bei Zahlarten mit Autorisierung)
AuthText	an	20	Text zum Autorisierungsstatus
PaymentId	n	38	ID für Rückfragen bei Pago (wird nur intern bei Pago verwendet)

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM „http://xml.pago.de/2.0/PAGOxml.dtd“>
<PAGOxml version="2.0" timeStamp="2001-10-23T16:10:19+02:00">
```

```
<header>
<communicationId>2001-12-31T12:30:32+02:00-001122334457-00601@www.pago.de
</communicationId>
<Server>PagoServer</Server>
<retry>false</retry>
  <inReplyTo>2001-12-31T12:02:30+02:00-000000000125-00001@www.client.url
</inReplyTo>
</header>
<response>
<MultipayInitPaymentResponse>
  <AuthStatus>0</AuthStatus>
  <AuthText>AUTH CODE:6236</AuthText>
  <PaymentId>3422</PaymentId>
</MultipayInitPaymentResponse>
</response>
</PAGOxml>
```

Wenn eine Autorisierung abgelehnt wird, hat der Tag **response** beispielhaft folgenden Aufbau:

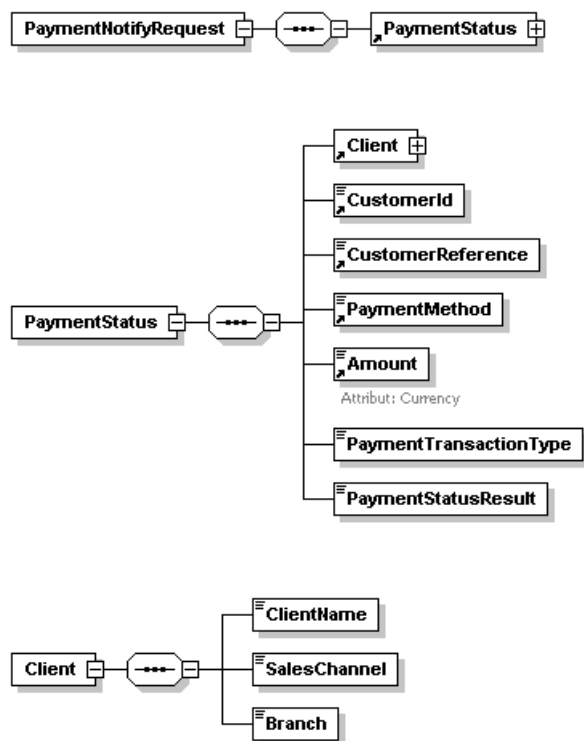
```
<response>
<MultipayInitPaymentResponse>
  <AuthStatus>1</AuthStatus>
  <AuthText>AUTHORIZATION DECLINED</AuthText>
  <PaymentId>3411</PaymentId>
</MultipayInitPaymentResponse>
</response>
```

Fehlercode	Grund	Abhilfe
101.001	Kein gültiger numerischer Wert	Zahl verwenden
101.004	Currency ist für diesen Dienst Pflichtfeld	Currency muss gefüllt sein
101.005	Betrag war <=0	Positiven Betrag verwenden
101.006	Mode-Attribut war <0 oder >3	0, 1, 2 oder 3 verwenden
101.007	PaymentMethod passt nicht zum Mode	
101.008	Parameterlänge zu lang oder zu kurz	Parametername steht im Twister-Log
101.009	go_to_payment Ergebnis war „999“	Autor fragen
101.500	Ungültiger Mandant	
101.501	Ungültige Personendaten	
101.502	Ungültige Adressdaten	
101.504	Ungültige Währung	
101.506	Unbekannter Mandant	

Fehlercode	Grund	Abhilfe
101.509	Transaktionsart fehlt	
101.511	Zahlart ist unbekannt	
101.512	Stammdaten sind ungültig	
101.513	Bereits autorisiert	
101.514	Autorisierung in Bearbeitung	
101.515	Auftrag abgelaufen	
101.517	Sequenznummer ist ungültig	

7.5.4 Rückmeldung des Payment-Status

Wenn die Zahlung mit **MultipayInitPaymentRequest** durchgeführt wurde, teilt Pago Ihnen mit **MultipayInitPaymentResponse** das Ergebnis der Zahlung sofort mit. Die Rückmeldung seitens der Kreditkartengesellschaft bzw. der Bank über den Payment-Status kommt erst zu einem späteren Zeitpunkt (asynchrone Rückmeldung). Der Vorgang ist über **CustomerReference** definiert.



Tag	Datentyp	Feldlänge	Bemerkung
PaymentStatus			
Client			
ClientName	an	20	Name des Mandanten Konstante in Absprache mit Pago
SalesChannel	an	20	Vertriebslinie Konstante in Absprache mit Pago
Branch	an	20	Filiale Konstante in Absprache mit Pago
CustomerId	an	20	Ihre eindeutige Kundenreferenz unter ClientName / SalesChannel / Branch
Customer-Reference	an	20	Ihre eindeutige Vorgangsreferenz unter ClientName / SalesChannel / Branch (siehe auch Abschnitt 7.5) Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
PaymentMethod	an	3	Mit Pago abgesprochene Zahlart, z. B. SEL : Lastschriftverfahren (DTA) SKK : Kreditkarteneinzug PBX : Paybox SBX : Stackbox (Initialisierung über Modus GUI) NN : Nachnahme REC : Rechnungskauf VOR : Vorkasse
Amount			
Amount	n	10,2	Saldo des Vorgangskontos negativer Betrag: Guthaben positiver Betrag: Forderung
Currency	an	3	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest

Tag	Daten typ	Feld- länge	Bemerkung
Payment- TransactionType	an	25	Transaktionsart Authorization: Betragsautorisierung Capture: Einlösen auf eine Autorisierung Credit: Auszahlung Gutschrift Exception: Verarbeitung wurde unterbrochen; neuen Status abwarten PBXCapture: Einlösen auf eine Paybox-Autorisierung PBXCaptureCancel: Einlösen widerrufen und damit auszahlen Payment: Sofortbelastung des Betrags auf der Kreditkarte PaymentAfterFailedCapture: Sofortbelastung des Betrags nach fehlgeschlagenem Einlösen auf eine Autorisierung
PaymentStatus- Result	an	5	Ergebnis der Verarbeitung OK: Verarbeitung erfolgreich durchgeführt Failed: Verarbeitung fehlgeschlagen Error: Fehler während der Ausführung

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:33+02:00">
<header>
  <communicationId>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de>
  <communicationId>
  <Server>PagoServer</Server>
  <retry>false</retry>
</header>
<request>
  <PaymentNotifyRequest>
    <PaymentStatus>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
      <CustomerId>123456789001</CustomerId>
      <CustomerReference>abcdefg2001</CustomerReference>
      <PaymentMethod>ELV</PaymentMethod>
      <Amount Currency="EUR">195.95</Amount>
      <PaymentTransactionType>Payment</PaymentTransactionType>
      <PaymentStatusResult>OK</PaymentStatusResult>
    </PaymentStatus>
  </PaymentNotifyRequest>
</request>
```

</PAGOxml>

Auf **PaymentNotifyRequest** wird von Pago ein **PaymentNotifyResponse** erwartet. Erst mit dem Eingang eines leeren **PaymentNotifyResponse** ohne **errorNotification** gilt die Zustellung des Status als abgeschlossen.

PaymentNotifyResponse

Das Tag **PaymentNotifyResponse** hat keine definierten Elemente.

Tag	Datentyp	Feldlänge	Bemerkung
PaymentNotifyResponse			

Das folgende Beispiel zeigt einen möglichen **PaymentNotifyResponse**:

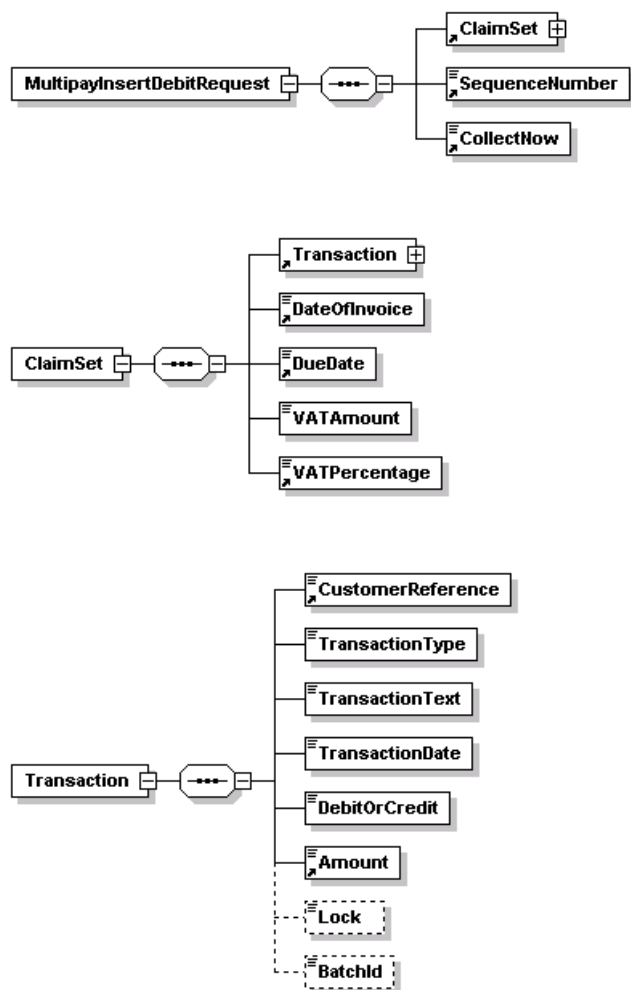
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml timeStamp="2001-10-30T16:45:25+01:00" version="2.0">
  <header>
    <communicationId>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url>
    <communicationId>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
    <retry>>false</retry>
    <inReplyTo>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de
    </inReplyTo>
  </header>
  <response>
    <PaymentNotifyResponse/>
  </response>
</PAGOxml>
```

7.5.5 Einstellen von Forderungsbuchungen

Eine Forderungsbuchung stellen Sie mit **MultipayInsertDebitRequest** ein.



Achten Sie darauf, dass die Summe der eingestellten Forderungen nicht höher ist als der bei **MultipayInitPaymentRequest** angegebene Betrag.



Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:

Tag	Daten typ	Feld-länge	Pflicht-feld	Bemerkung
ClaimSet				
Transaction				
Customer-Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz wie in MultipayInit-PaymentRequest angegeben Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).

Tag		Daten typ	Feld- länge	Pflicht- feld	Bemerkung
	Transaction-Type	an	6	Ja	Transaktionsart FD = Forderung Ware / Dienstleistung VD = Versandkosten GS = Gutschrift RT = Retoure VO = Voucher ! Die Transaktionsart bestimmt nicht, ob der Betrag im Soll oder Haben gebucht wird.
	Transaction-Text	an	30	Ja	Transaktionstext für Buchhaltung / Kontoauszug
	Transaction-Date	an	10	Ja	Transaktionsdatum (Kaufdatum) Format: YYYYMMDD oder DD.MM.YYYY
	DebitOrCredit	an	1	Ja	Soll oder Haben S : Soll (debit)(FD/VD/GS/RT) Für alle Buchungen ist S zu verwenden. H : Haben (credit) nur intern bei Pago
Amount					
	Amount	n	10,2	Ja	Bruttobetrag inkl. MwSt + : Forderung oder Stornierung einer Gutschrift - : Gutschrift oder Stornierung einer Forderung
	Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPayment-Request
	Lock			Nein	Reserviert
	BatchId	an	30	Nein	Batch-ID Dient bei Bedarf zur Gruppierung von Sätzen (nur nach Absprache)
	DateOfInvoice	an	10	Ja	Rechnungsdatum Format: YYYYMMDD oder DD.MM.YYYY

Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
DueDate	an	10	Ja	Fälligkeitsdatum Format: YYYYMMDD oder DD.MM.YYYY
VATAmount	n	10,2	Ja	Mehrwertsteuerbetrag oder -1 Bei -1 keine Angabe. Wird von Pago berechnet, wenn ein MwSt-Satz angegeben ist.
VATPercentage	n	4,2	Ja	Mehrwertsteuer in Prozent -1: keine Angabe (Pago bucht keine MwSt im Debitorensystem)
SequenceNumber	n	6	Ja	Sequenznummer Muss mit „1“ beginnen und wächst mit jedem Folgeaufruf um 1 (siehe auch Abschnitt 7.3.2).
CollectNow	n	1	Ja	Der Tag CollectNow ist im Multipay Modus 1 für Forderun- gen gegenstandslos und sollte deshalb =false sein. Wenn eine Gutschrift einge- stellt wird, führt Collect- Now=true zur sofortigen Auszahlung. Bei Collect- Now=false muss die Auszah- lung über Multipay- SettleAccountRequest aus- gelöst werden.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM „http://xml.pago.de/2.0/PAGOxml.dtd“>
<PAGOxml version="2.0" timeStamp="Thu May 17 13:42:15 2001">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>false</retry>
  </header>
  <request>
    <MultipayInsertDebitRequest>
      <ClaimSet>
        <Transaction>
```

```

        <CustomerReference>gfdgfdgfd54543453</CustomerReference>
        <TransactionType>FD</TransactionType>
        <TransactionText>Beispiel Forderung</TransactionText>
        <TransactionDate>30.12.2002</TransactionDate>
        <DebitOrCredit>S</DebitOrCredit>
        <Amount Currency="EUR" >99.99</Amount>
        <BatchId></BatchId>
    </Transaction>
    <DateOfInvoice>16.07.2001</DateOfInvoice>
    <DueDate>16.10.2001</DueDate>
    <VATAmount>13.79</VATAmount>
    <VATPercentage>16.0</VATPercentage>
    </ClaimSet>
    <SequenceNumber>1</SequenceNumber>
    <CollectNow>false</CollectNow>
</MultipayInsertDebitRequest>
</request>
</PAGOfxml>
    
```

Als Rückgabewert auf **MultipayInsertDebitRequest** wird neben dem Ergebnis der einzuziehende Betrag gemeldet („0“ bei **CollectNow** = **false**).



Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
Amount				
Amount	n	10,2	Ja	Einzuziehender Betrag
Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPayment- Request

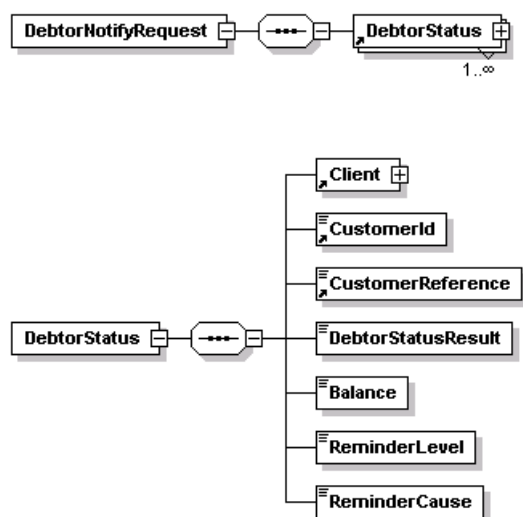
```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOfxml SYSTEM „http://xml.pago.de/2.0/PAGOfxml.dtd“>
<PAGOfxml timeStamp="2001-10-30T16:45:25+01:00" version="2.0">
    <header>
        <communicationId>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de
        </communicationId>
        <Server> PagoServer </Server>
        <retry> false </retry>
        <inReplyTo>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
        </inReplyTo>
    </header>
    <response>
        <MultipayInsertDebitResponse>
            <Amount Currency="EUR"> 0 </Amount>
        </MultipayInsertDebitResponse>
    </response>
</PAGOfxml>
    
```

Fehlercode	Grund	Abhilfe
103.001	Nicht-boole'scher Wert	true oder false verwenden
103.002	Kein gültiger numerischer Wert	Zahl verwenden
103.003	Parameterlänge zu lang oder zu kurz	Parametername steht im Twister-Log
103.500	Ungültige Mandanten-Identifikation	
103.505	Abweichende Währung	
103.506	Ungültige Transaktionsart	
103.507	Ungültige Autorisierung	
103.508	Systemfehler	
103.509	Systemfehler	
103.513	Einlösungsfehler	
103.516	Ungültige Sequenznummer	

7.5.6 Empfangen des Debitorenstatus

Unabhängig von **MultipayInsertDebitRequest** kann sich der Debitorenstatus verändern. Der jeweils aktuelle Debitorenstatus wird (ohne Historie) von Pago via **Debtor-NotifyRequest** bereitgestellt. Dabei werden mit einem Request alle zur Verfügung stehenden Debitorenstatus versendet.



Tag	Daten typ	Feld- länge	Bemerkung
DebtorStatus			
Client			
ClientName	an	20	Name des Mandanten Konstante in Absprache mit Pago
SalesChannel	an	20	Vertriebslinie Konstante in Absprache mit Pago
Branch	an	20	Filiale Konstante in Absprache mit Pago
CustomerId	an	20	Ihre eindeutige Kundenreferenz unter ClientName / SalesChannel / Branch
Customer- Reference	an	20	Ihre eindeutige Vorgangsreferenz unter ClientName / SalesChannel / Branch (siehe auch Abschnitt 7.5) Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
DebtorStatus- Result	an	1	Debitorenstatus N : nicht bezahlt B : bezahlt, Vorgang ist ausgeglichen Z : überzahlt F : überfällig/in Mahnung, Vorgang ist nicht ausgeglichen (Ursache kann eine fehlge- schlagene Zahlung oder eine Rückforde- rung sein.)
Balance			
Balance	n	15,2	Saldo des Vorgangskontos Negativer Betrag: Guthaben Positiver Betrag: Forderung
Currency	an	3	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest

Tag	Daten typ	Feld- länge	Bemerkung
ReminderLevel	an	1	Aktuelle Mahnstufe 0: keine Mahnung ... 4: Inkasso A: Ausgebucht
ReminderCause	an	3	Mahngrund RL0: keine Deckung RL1: Konto erloschen RL2: Konto-Nr. / Name nicht identisch (falsch oder Sparkonto) RL3: kein Abbuchungsauftrag bzw. keine Einzugsermächtigung RL4: Rückruf RL5: Widerspruch RL6: Rückgabe LSV: Zahlart Lastschrift, nicht einziehbar CB: Chargeback KKE: Zahlart Kreditkarte, nicht einziehbar

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM „http://xml.pago.de/2.0/PAGOxml.dtd“>
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
<header>
  <communicationId> 2001-12-31T12:30:30+02:00-001122334455-00601@10.10.10.10
  </communicationId>
  <Server>
    PagoServer
  </Server>
  <retry>
    true
  </retry>
</header>
<request>
  <DebtorNotifyRequest>
    <DebtorStatus>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
      <CustomerId>123456789001</CustomerId>
      <CustomerReference>abcdefg2001</CustomerReference>
      <DebtorStatusResult>B</DebtorStatusResult>
      <Balance Currency="EUR">0</Balance>
      <ReminderLevel>0</ReminderLevel>
      <ReminderCause> </ReminderCause>
    </DebtorStatus>
    <DebtorStatus>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
```

```
<CustomerId>123456789002</CustomerId>
<CustomerReference>abcdefg1999</CustomerReference>
<DebtorStatusResult>B</DebtorStatusResult>
<Balance Currency="EUR">0</Balance>
<ReminderLevel>0</ReminderLevel>
<ReminderCause> </ReminderCause>
</DebtorStatus>
<DebtorStatus>
  <Client>
    <ClientName>123</ClientName>
    <SalesChannel>123</SalesChannel>
    <Branch>123</Branch>
  </Client>
  <CustomerId>123456789000</CustomerId>
  <CustomerReference>abcdefg2000</CustomerReference>
  <DebtorStatusResult>F</DebtorStatusResult>
  <Balance Currency="EUR">34,95</Balance>
  <ReminderLevel>A</ReminderLevel>
  <ReminderCause>LSV</ReminderCause>
</DebtorStatus>
</DebtorNotifyRequest>
</request>
</PAGOxml>
```

Auf **DebtorNotifyRequest** wird von Pago ein **DebtorNotifyResponse** erwartet. Erst mit dem Eingang eines leeren **DebtorNotifyResponse** ohne **error-Notification** gilt die Zustellung des Status als abgeschlossen.

DebtorNotifyResponse

Damit hat das Tag **DebtorNotifyResponse** keine definierten Elemente.

Tag	Daten typ	Feld- länge	Bemerkung
DebtorNotifyResponse			

Das folgende Beispiel zeigt einen möglichen **DebtorNotifyResponse**.

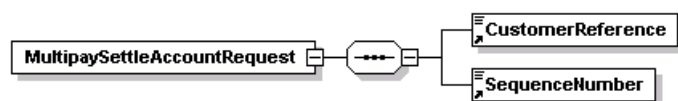
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml timeStamp="2001-10-30T16:45:25+01:00" version="2.0">
  <header>
    <communicationId>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url>
    <communicationId>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
    </communicationId>
    <retry>false</retry>
    <inReplyTo>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de
    </inReplyTo>
  </header>
  <response>
    <DebtorNotifyResponse/>
  </response>
</PAGOxml>
```

```
</response>
</PAGOxml>
```

7.5.7 Anstoßen des Einzugs

Nach dem Einstellen von einer oder mehreren Forderungsbuchungen wird mit dem **MultipaySettleAccountRequest** der Einzug freigegeben. Der Einzug erfolgt in der Regel asynchron. Ergebnisse werden von Pago über **DebtorNotifyRequest** gemeldet.

Folgende Informationen werden in dem Funktionsaufruf an den Pago-Server geleitet:



Tag	Daten typ	Feld-länge	Pflicht-feld	Bemerkung
Customer-Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz wie in MultipayInitPaymentRequest angegeben Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
SequenceNumber	n	6	Ja	Sequenznummer Muss mit „1“ beginnen und wächst mit jedem Folgeaufruf um 1 (siehe auch Abschnitt 7.3.2).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM „http://xml.pago.de/2.0/PAGOxml.dtd“>
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>false</retry>
  </header>
```



```
<request>
  <MultipaySettleAccountRequest>
    <CustomerReference>gfdgdfgd54543453</CustomerReference>
    <SequenceNumber>2</SequenceNumber>
  </MultipaySettleAccountRequest>
</request>
</PAGOxml>
```

Zu **MultipaySettleAccountRequest** wird mittels **MultipaySettleAccount-Response** der einzuziehende Betrag zurückgemeldet.



Tag	Daten typ	Feld- länge	Bemerkung
Amount			
Amount	n	10,2	Einzuziehender Betrag
Currency	an	3	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest

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

<!DOCTYPE PAGOxml SYSTEM „http://xml.pago.de/2.0/PAGOxml.dtd“>
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>2001-12-31T12:34:30+02:00-001122334459-00601@www.pago.de
    </communicationId>
    <Server> PagoServer </Server>
    <retry> false </retry>
    <inReplyTo>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </inReplyTo>
  </header>
  <response>
    <MultipaySettleAccountResponse>
      <Amount Currency="EUR"> 99.99 </Amount>
    </MultipaySettleAccountResponse>
  </response>
</PAGOxml>
```

Fehlercode	Grund	Abhilfe
105.002	Kein numerischer Wert	Numerischen Wert ein- setzen
105.004	Currency ist Pflicht in diesem Service	Währung einsetzen

8 Multipay mit Modus 2

In diesem Kapitel wird die Handhabung des Modus 2 beschrieben. Detaillierte Informationen für die Auswahl des benötigten Modus finden Sie in Kapitel 2 „Entscheidungshilfen für die angemessene Payment-Methode“.

Im Modus 2 wird der Betrag aufgrund der bei Pago vorhandenen Daten mit dem Initialisieren sofort vollständig eingezogen (**sales**), und es wird eine Forderung zur internen Verarbeitung angelegt. Der Modus 2 unterstützt die Zahlarten „Kreditkarte“ und „Lastschrift“. Die Methode **MultipayInsertDebitRequest** wird nur im Falle einer Gutschrift benötigt.



Für Paybox müssen Sie trotz der Charakteristik des Netzbetriebs den Modus 0 verwenden.

Beim Kreditkarten-Netzbetrieb (d. h. der Mandant hat einen eigenen Kreditkarten-Akzeptanzvertrag) erhält Pago keine Informationen über Chargeback-Buchungen. In einer solchen Konstellation ist es nicht möglich, automatisches Mahnwesen über Pago abzuwickeln.

Im Modus 2 werden die Forderungen mit Standardwerten zur Pago-eigenen Verarbeitung ins Debitoren-System eingestellt. Dies ist der Weg, wenn Pago keine debitorischen Zusatzleistungen erbringen soll, wie es z. B. beim Kreditkarten-Netzbetrieb der Fall ist.



Beim Kreditkarten-Netzbetrieb werden Kreditkartentransaktionen direkt über den Vertrag des Händlers verarbeitet. Wenn ein Kreditkarten-Vertrag von Pago genutzt wird, kann Pago mehr debitorische Dienstleistungen erbringen.

8.1 Anwendungsregeln für den Einsatz von Multipay im Modus 2

Die gewünschten Zahlarten müssen Sie mit Pago absprechen. Gemäß dieser Absprache richtet Pago Bankkonten und Kreditkarten-Parameter ein und teilt sie, soweit notwendig, dem aufrufenden System mit.

Ein Auftrag wird mit dem Ziel eingestellt, einen Geldbetrag in einer vorher bekannten Maximalhöhe vom Kunden zu erhalten. Beim Anlegen des Auftrags wird dieser Gesamtbetrag an Pago übergeben. Für den Betrag gelten die drei folgenden Regeln:

- (§1) Die Summe aller Einzüge pro Auftrag darf den Gesamtbetrag nicht überschreiten.
- (§2) Der Gesamtbetrag darf nicht nachträglich erhöht werden.
- (§3) Die Summe aller Gutschriftenbeträge darf die Summe aller Forderungsbeträge nicht überschreiten.

Wenn die Forderung erst zu einem späteren Zeitpunkt entsteht, kann die Zahlung zum Zeitpunkt der Auftragseinstellung lediglich autorisiert werden (Modus 0). Bei Online-Zahlarten wird der Einzug durch den Aufruf von **MultipaySettleAccountRequest** zum Zeitpunkt der (Teil-)Lieferung initiiert, und zwar in Höhe der bis zu diesem Zeitpunkt verbuchten noch nicht eingezogenen Forderungen. Wenn für den Vorgang lediglich eine Forderung benötigt wird, kann der Aufruf von **MultipaySettleAccountRequest** entfallen; in diesem Fall muss bei **MultipayInsertDebitRequest** der Tag **CollectNow = true** verwendet werden.

Nicht nur die Forderungen, die aus der Bestellung des Kunden hervorgehen, können storniert werden, sondern auch Gutschriften, die z. B. bei Beschädigung der Ware oder Retoure ausgestellt werden.

Darüber hinaus wird zwischen Buchungen und Stornos unterschieden. Forderungsbuchungen und Gutschriftenstornos werden in positives Soll gebucht, Forderungsstornos und Gutschriftenbuchungen in negatives Soll. Dabei gilt folgende Regel:

(§4) Die Summe der Stornos darf pro Satzart die Summe der Buchungen nicht überschreiten.

! *Ein einmal stornierter Betrag (Forderungsstorno, Retoure, Versandkostenstorno) kann nicht mehr gefordert werden.
Eine Gutschrift (aus Kulanz) kann nur auf existierende Zahlungseingänge gewährt werden. Eine Gutschrift kann auch wieder storniert werden, die Regeln in §1 bis §4 sind jedoch immer verbindlich.*

! *Je nach Zahlart gibt es weitere Einschränkungen bezüglich Gutschriften und Stornos. Deshalb sollten Sie Forderungen möglichst zu einem Zeitpunkt einstellen, an dem Sie prozessbedingte Rückabwicklungen abschließen können.*

8.2 Umgang mit Retouren und Gutschriften

Mit folgender Vorgehensweise bleiben die Warenbewegungen und Zahlungen je Vorgang bei allen Parteien übersichtlich und nachvollziehbar. Rückbelastungen aufgrund unklarer Abbuchungen können reduziert werden.

Situation 1: Dem Kunden wird nach einer Warenretoure Ersatz mit gleichem Wert geliefert:

Diese Situation ist für den Vorgang in Multipay nicht relevant.

- Situation 2: Dem Kunden wird nach einer Warenretoure Ersatz mit geringerem Wert geliefert. Zum Ausgleich soll er eine Gutschrift erhalten:
Die Gutschrift über den gewählten Betrag wird mittels **MultipayInsertDebitRequest** angelegt und, je nach Bedarf, entweder sofort oder mit **MultipaySettleAccount** zur Auszahlung gebracht.
- Situation 3: Dem Kunden wird nach einer Warenretoure der gezahlte Betrag erstattet. Zu einem späteren Zeitpunkt wird erneut eine Ware an ihn gesandt:
Mit der Warenretoure sollte der zugehörige Vorgang geschlossen werden. Dazu wird mit **MultipayInsertDebitRequest** eine Retoure eingestellt. Der Gutschriftsbetrag wird nach Bedarf sofort oder mit dem nächsten **MultipaySettleAccount** ausgezahlt.
Für die nächste Bestellung wird mit **MultipayInitPayment** ein neuer Vorgang angelegt.

8.3 Transaktionssicherheit

8.3.1 Aufrufwiederholung bei Verbindungsabbruch

In der Verbindung zwischen dem Mandantensystem und Pago kann es vorkommen, dass der Aufrufende kein Ergebnis erhält. Das kann u. a. aus zwei Gründen der Fall sein:

- Der Empfänger hat den Aufruf nicht erhalten.
- Der Empfänger hat den Aufruf erhalten und verarbeitet, aber die Rückmeldung ist nicht beim Absender angekommen.

Um solche Vorfälle zu vermeiden, besitzt die Anbindung über LocalServer und XML einen speziellen Retry-Mechanismus. Dieser Mechanismus identifiziert jede Kommunikation einer Transaktion eindeutig mit Hilfe der **communicationId**.

Wird der Response nicht innerhalb einer definierten Zeitspanne empfangen, die über die Konfigurationsvariable **Timeout** definiert wird, sollte der Request mit **retry = true** und der gleichen **communicationId** erneut gesendet werden.

Mit dem Retry können drei Situationen eintreten:

- Wenn Pago den Request vor dem Retry nicht erhalten hat, wird der Request nach dem Retry von Pago verarbeitet und der entsprechende Response zurückgesendet.
- Wenn Pago den Request vor dem Retry erhalten und auch verarbeitet hat, wird nach dem Retry das Ergebnis dieser Verarbeitung im Response zurückgesendet.
- Wenn Pago den Request vor dem Retry erhalten hat und die Verarbeitung noch andauert, wird nach dem Retry im Response die Meldung zurückgesendet: **this transaction is still in progress**. Dann muss der Request mit einem weiteren Retry wiederholt werden.

Das folgende Beispiel zeigt den Response im Falle der noch andauernden Verarbeitung. Nähere Informationen hierzu finden Sie in Abschnitt 5.2 „Header-Block“.

```
</header>
<response>
  <errorNotification type="recoverable" issuer="PagoServer"
    service="MultipayInitGuiPayment">
    <errorText> <![CDATA[this transaction is still in progress]]> </errorText>
    <errorNumber> 30007 </errorNumber>
  </errorNotification>
</response>
</PAGOxml>
```

Wenn das Ergebnis der Verarbeitung vorliegt, kann es in eine der folgenden Klassen eingestuft werden:

- 1) Erfolgreiche Durchführung
- 2) Die Transaktion wurde mit nicht behebbaren Fehlern durchgeführt (**fatal**). Eine Wiederholung wird kein anderes Ergebnis hervorbringen. Daher sollten keine Wiederholungsaufrufe durchgeführt werden.
- 3) Die Transaktion wurde mit einem Fehler durchgeführt, der zu einem späterem Zeitpunkt möglicherweise nicht mehr auftritt. Zur Kennzeichnung des möglichen späteren Wiederholungsversuchs wird der Fehler mit **recoverable** klassifiziert.
Wenn in einem Fehlerszenario die Bank bei einer Autorisierung nicht erreicht wird, kann der Betrag auch nicht bei der betreffenden Kreditkarte reserviert werden. Die Initialisierung ist nicht abgeschlossen. Die Transaktion wird mit dem negativen Ergebnis **recoverable** abgeschlossen.
Die Autorisierung kann zu einem späteren Zeitpunkt erneut ausgeführt werden. Der erneute Autorisierungsversuch wird mit alter **communicationId** und gleicher Fremdbelegnummer (**CustomerReference**) initialisiert.

Die Aufrufwiederholung ist mehrfach ausführbar. Wenn Sie also nach einem Aufruf kein Ergebnis zurückerhalten haben, müssen Sie im Zweifelsfall den Aufruf solange wiederholen, bis Sie Klarheit über das Ergebnis haben. Wählen Sie dabei bitte einen sinnvollen Zeitabstand zwischen den Wiederholungsanfragen. Einen Funktionsaufruf, der wegen Timeout nicht abgeschlossen werden konnte, können Sie sofort wiederholen. Wenn eines der angebundenen Systeme (z. B. ein Banksystem) nicht zur Verfügung steht, wiederholen Sie den Aufruf nach einigen Minuten.

8.3.2 Handhabung der Sequenznummer

Die Sequenznummer dient der Einhaltung der Verarbeitungsreihenfolge. Pro Fremdbelegnummer muss die Sequenznummer aufsteigend und lückenlos nummeriert werden. Die Auftragsinitialisierung trägt die implizite Sequenznummer „0“. Verwenden Sie deshalb bei der als nächstes aufgerufenen Methode die Sequenznummer „1“. Eine falsche Sequenznummer wird mit einem Fehler abgelehnt. Für weitere Informationen zu Fehlermeldungen siehe auch Abschnitt 5.5 „Fehlermanagement“.

8.3.3 Verhalten der Methoden bei Wiederholungsaufrufen

Nach einem Wiederholungsaufwurf mit gesetztem Retry-Flag wird das Ergebnis der ersten erfolgreichen bzw. fatalen Verarbeitung im Response zurückgeliefert. Die folgende Angabe gilt für Transaktionen, die durch das Flag **retryable** wiederholbar sind:

```
<errorNotification issuer="PagoServer"
    service="MultipayInitPayment" type="retryable">
```

Die Funktionen aus Pago Multipay verhalten sich folgendermaßen:

MultipayInitPaymentRequest wird identifiziert durch

- **ClientName**
- **SalesChannel**
- **Branch**
- **CustomerReference.**

Wiederholungsanfragen zu fehlgeschlagenen oder abgelehnten Zahlungen werden erneut durchgeführt. Wiederholungsanfragen zu akzeptierten Zahlungen liefern das vorher erhaltene Ergebnis zurück, wenn sie in den anderen Parametern übereinstimmen. Andernfalls werden sie abgelehnt.

MultipayInsertDebitRequest und **MultipaySettleAccountRequest** werden identifiziert durch

- **ClientName**
- **SalesChannel**
- **Branch**
- **CustomerReference**
- **SequenceNumber.**

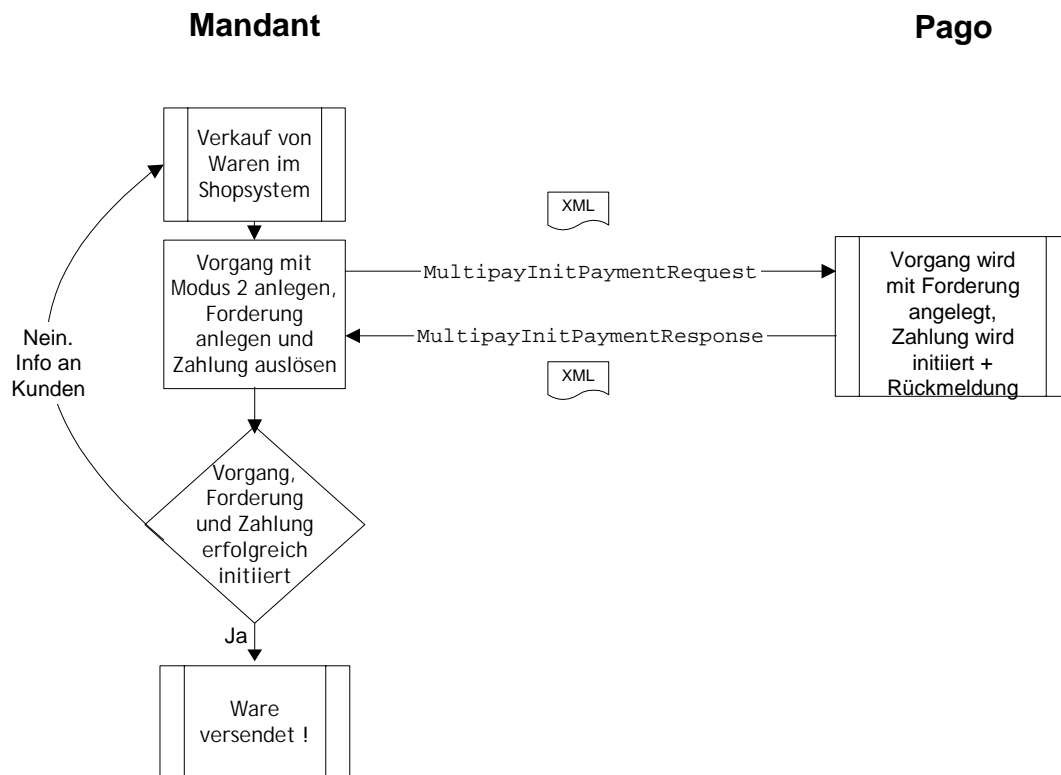
Wiederholungsanfragen liefern das vorher erhaltene Ergebnis zurück. Anfragen mit falscher Sequenznummer werden abgelehnt.

8.4 Anwendungsbeispiele für Multipay mit Modus 2

Die Beispiele dienen der Veranschaulichung Ihrer Möglichkeiten und bieten außerdem eine Diskussionsgrundlage für die Unterstützung durch Pago Consulting.

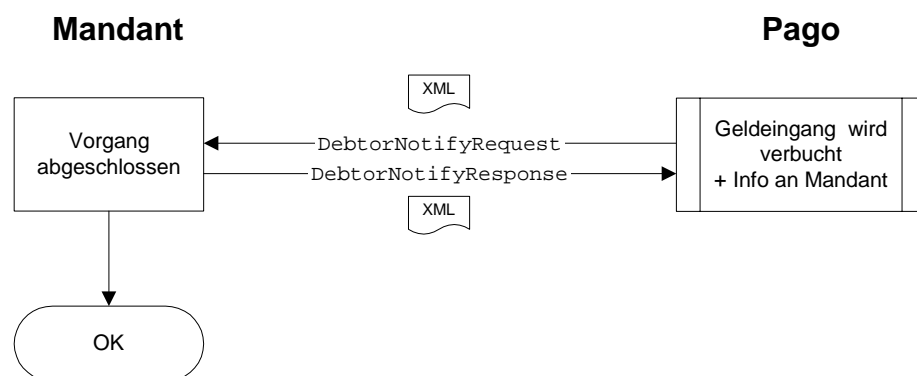
8.4.1 Autorisierung und Zahlung sofort, reines Payment

Autorisierung, Zahlung und default Forderungsbuchung erfolgen in einem Schritt. Anzuwenden bei Kreditkarten-Netzbetrieb.



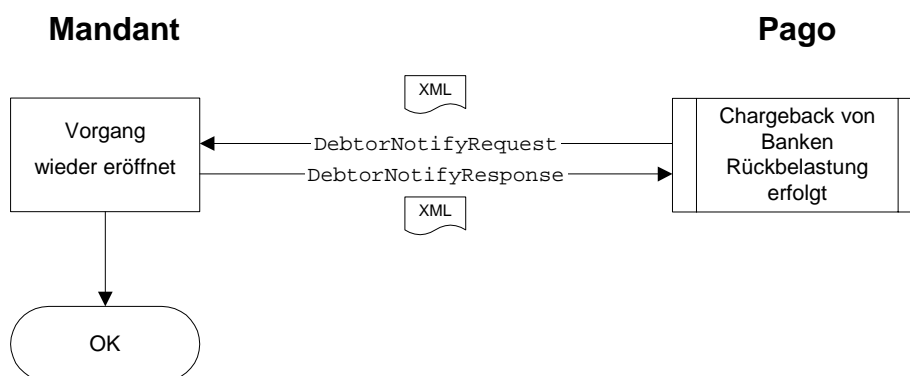
Im Modus 2 dürfen keine Forderungsbuchungen auftreten, da diese implizit erzeugt werden.

Der Debitorenstatus gibt wieder Auskunft über den Geldeingang.

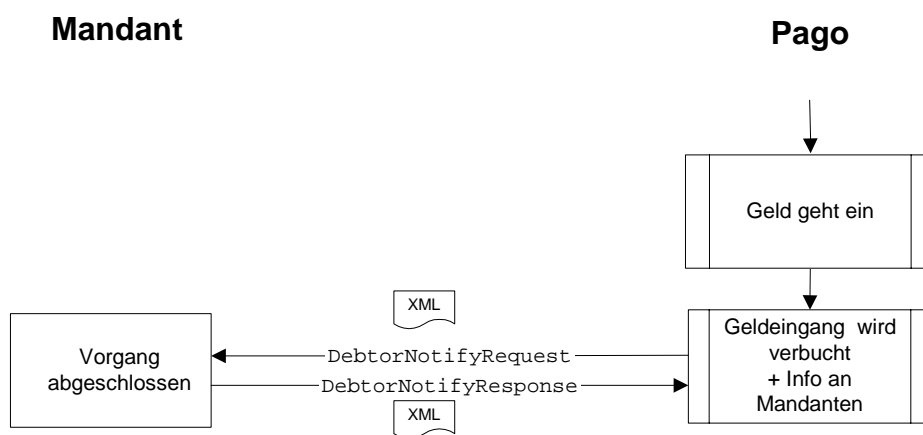


8.4.2 Meldung nach Chargeback

Statusinformationen aus dem Debitorenmanagement können auch nach dem Abschluss eines Vorgangs auftreten. In diesem Beispiel nehmen wir ein Chargeback auf eine Kreditkartenzahlung oder eine Rücklastschrift auf eine Einzugsermächtigung an. Der Debitorenstatus wird mittels **DebtorNotifyRequest** bereitgestellt.



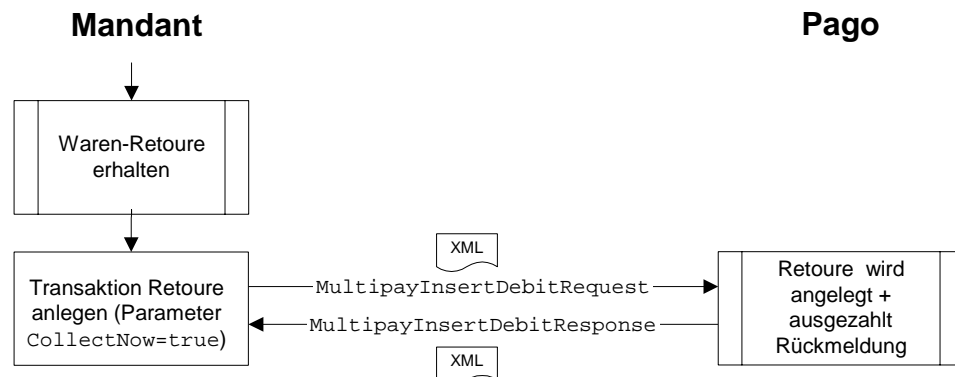
Das Mahnwesen wird in diesem Fall über Pago abgewickelt. Der Kunde reagiert auf die Mahnung und zahlt. Der Geldeingang wird über den Debitorenstatus mitgeteilt.



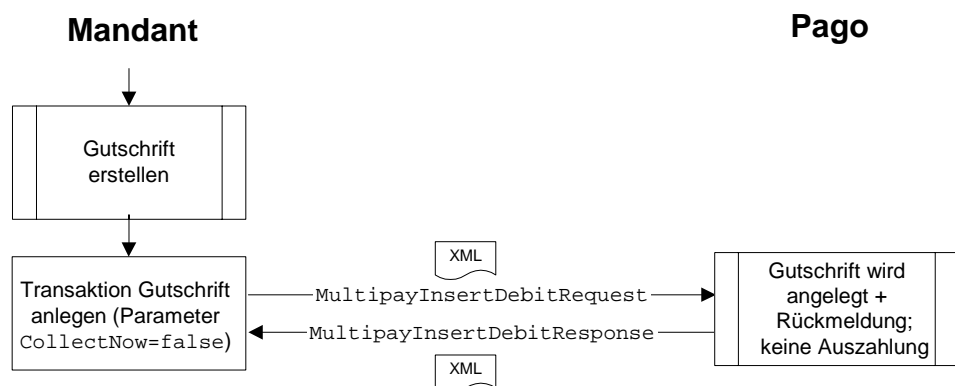
8.4.3 Gutschrift

Gutschriften werden mit **MultipayInsertDebitRequest** eingestellt. (Achten Sie hierbei auf die Regeln aus Abschnitt 8.1 „Anwendungsregeln für den Einsatz von Multipay im Modus 2“).

Dieses Beispiel zeigt den möglichen Ablauf für die Verarbeitung einer Retoure und einer Gutschrift. Die Retoure wird mit sofortiger Auszahlung angelegt.



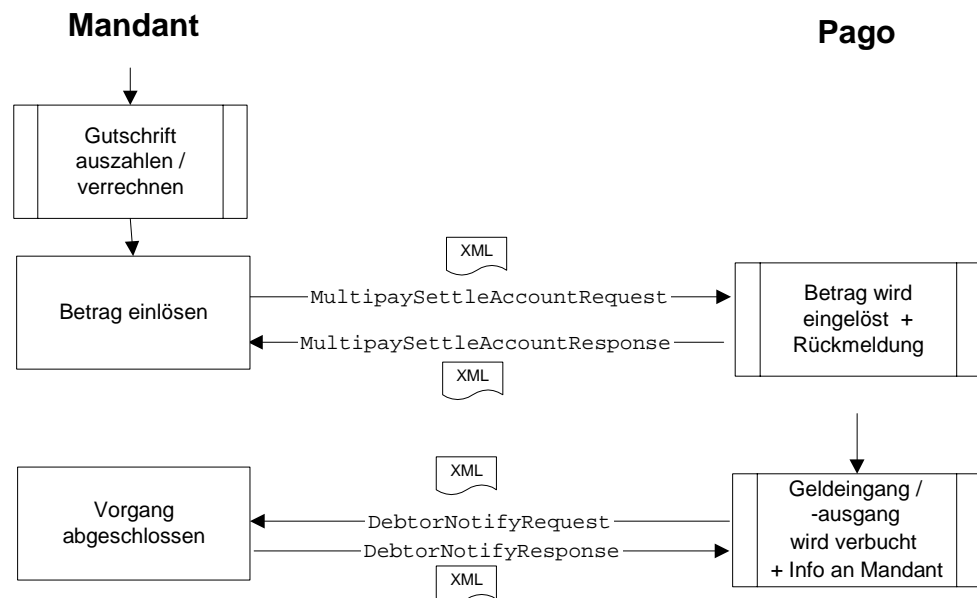
Wenn eine Gutschrift nicht ausgezahlt, sondern mit Forderungen verrechnet werden soll, verwenden Sie den Parameter **CollectNow = false**. Die Verrechnung erfolgt mit dem nächsten **MultipaySettleAccountRequest**.



8.4.4 Forderungskonto ausgleichen

Nach dem Aufruf der Funktion **MultipaySettleAccountRequest** wird die Gutschrift ausgeführt. Wenn in der Zwischenzeit eine neue Forderung eingestellt wird, wird diese mit der Gutschrift verrechnet und nur noch der Saldo ausgezahlt.

! *Durch eine Gutschrift wird eine Forderung nicht aufgehoben. Die Summe der Forderungen muss weiterhin kleiner oder gleich dem Betrag des **MultipayInitPaymentRequest** sein.*



8.5 Technische Schnittstellenbeschreibung

Der Service Pago Multipay wird integriert durch das Senden und Empfangen von XML-Dokumenten via LocalServer. Der Aufbau der benötigten XML-Dokumente wird nachfolgend beschrieben.

Die Pago-Multipay-Services im Modus 2 werden über folgende XML-Requests bereitgestellt:

- **MultipayCheckBankRequest** zum Prüfen einer Bankverbindung
- **MultipayCheckCreditCardRequest** zum Prüfen einer Kreditkarte
- **MultipayInitPaymentRequest** zum Initiieren von Zahlungen
- **MultipayInsertDebitRequest** zum Einstellen von Forderungsbuchungen
- **MultipaySettleAccountRequest** zum Anstoßen des Einzugs
- **MultipayReduceAmountRequest** zur Reduzierung des reservierten Betrags.

Die Fremdbelegnummer bezieht sich auf einen Vorgang im Mandantensystem, der über Pago Multipay abgewickelt wird (z. B. ein Kauf). Sie muss dem entsprechenden Vorgang beim Mandanten eindeutig zugeordnet werden können. Für die Fremdbelegnummer sollten ausschließlich Ziffern und Großbuchstaben verwendet werden, da das System Kleinbuchstaben automatisch in Großbuchstaben umsetzt. Falls zwei Fremdbelegnummern angelegt werden, die sich nur durch Groß- und Kleinschreibung voneinander unterscheiden, können sie vom System nicht mehr eindeutig zugeordnet werden.

! Halten Sie sich bei Datumseingaben genau an das vorgegebene Format (YYYYMMDD oder DD.MM.YYYY). Lassen Sie keine Stelle aus, auch nicht die führende „0“ (z. B.: „01.11.2000“).

Legende

Die Parameter des Request und des Response sind in Tabellen dargestellt. Für die Tabellen gilt folgende Legende:

Tag: Feldbezeichnung
Datentyp: „an“ = alphanumerisch / „n“ = numerisch
Feldlänge: Anzahl der Zeichen
Pflichtfeld: Angabe notwendig (Ja/Nein)



Bei numerischen Feldern, in denen eine Zahl mit Nachkommastellen eingetragen werden kann, sind unter „Feldlänge“ zwei Zahlen angegeben. Die Zahl vor dem Komma gibt die maximale Anzahl der gesamten Stellen an, die Zahl hinter dem Komma die maximale Anzahl der Nachkommastellen.

Beispiel: **10,2** --> max. 10 Stellen, davon max. 2 Nachkommastellen.

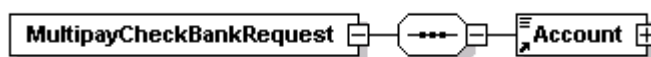
8.5.1 Prüfung einer Bankverbindung

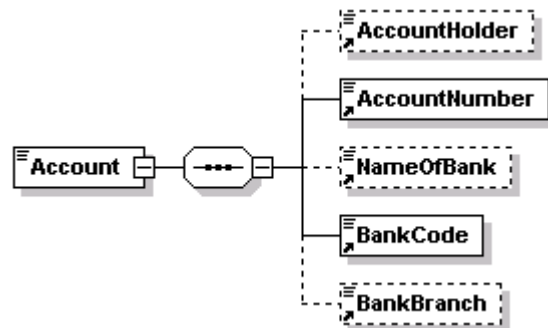
Eine Bankverbindung wird mit **MultipayCheckBankRequest** überprüft. Hierzu dienen

- die Bankleitzahl
- die Kontonummer
- das jeweilige Prüfziffernverfahren.

Die Prüfung der Bankverbindung sollte vor dem Eintrag mittels **MultipayInitPaymentRequest** erfolgen, um klare Ausfälle auszuschließen.

Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:



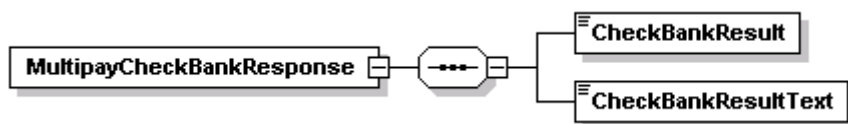


Tag	Daten typ	Feld- länge	Pflicht feld	Bemerkung
Account				
Account-Holder	an	30	Nein	Kontoinhaber (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)
Account-Number	an	10	Ja	Kontonummer (nur bei Zahlart ELV)
NameOfBank	an	30	Nein	Bankname (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)
BankCode	an	8	Ja	Bankleitzahl (nur bei Zahlart ELV)
BankBranch	an	30	Nein	Bankfiliale (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)

Beispiel:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:00:30+02:00-000000000123-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>false</retry>
  </header>
  <request>
    <MultipayCheckBankRequest>
      <Account>
        <AccountNumber>1234567890</AccountNumber>
        <BankCode>87654321</BankCode>
      </Account>
    </MultipayCheckBankRequest>
  </request>
</PAGOxml>
```



```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>2001-12-31T12:30:30+02:00-001122334455-00601@www.pago.de
    </communicationId>
    <Server>PagoServer</Server>
    <retry>false</retry>
    <inReplyTo>2001-12-31T12:00:30+02:00-000000000123-00001@www.client.url
    </inReplyTo>
  </header>
  <response>
    <MultipayCheckBankResponse>
      <CheckBankResult>0</CheckBankResult>
      <CheckBankResultText>ACCOUNT VALID</CheckBankResultText>
    </MultipayCheckBankResponse>
  </response>
</PAGOxml>
```

Fehlercode	Grund	Abhilfe
106.500	Aus DB: MCB-KLIENT_UNGUELTIG	

8.5.2 Prüfen einer Kreditkarte

Die Kreditkarte wird mit **MultipayCheckCreditCardRequest** überprüft. Dies geschieht nach folgenden Kriterien:

- Kreditkarten-Anfangsziffern
- Kreditkartengesellschaft
- Prüfziffer der Kreditkartennummer
- Verfalldatum der Kreditkarte.

Prüfen Sie die Kreditkarte vor dem Eintrag mittels **MultipayInitPaymentRequest**, um klare Ausfälle auszuschließen.

MultipayCheckCreditCard liefert Ihnen zwei Ergebnisfelder:

- **CheckCreditCardResult**
- **CheckCreditCardResultText**.

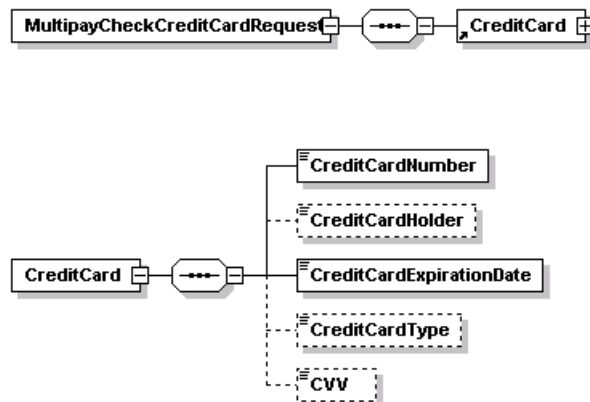
CheckCredit-CardResult	CheckCreditCardResultText		Ergebnisklasse
0	Kreditkarte existiert		OK
-1	Kreditkarte abgelaufen		fatal
-2	Falsche Kartennummernlänge		fatal
-3	Ungültige Eingabezeichen		fatal
-4	Keine numerische Kartennummer		fatal
-5	Unbekannter Kreditkartentyp		fatal
-6	Ungültige Checksumme		fatal
-7	Falsches Datumsformat		fatal
-8	Ablehnungsgrund der Prüfstelle (nur bei Online-Prüfung) Original-Fehlertext des Netzbetreibers der Kreditkarte wird ausgegeben. Liste der Fehlercodes:		
	02	Anrufen der AS-Rufnummer	fatal
	03	Nummer des Vertragsunternehmens unbekannt	fatal
	04	Karte einbehalten	fatal
	05	Autorisierung abgelehnt	fatal
	06	Filetransfer falsch, Wiederholung ab mitgeteilter Blocknummer möglich	fatal
	09	Verzögerte Bearbeitung der Autorisierung	fatal
	12	Ungültige Transaktion	fatal

CheckCredit-CardResult	CheckCreditCardResultText		Ergebnisklasse
	13	Verfügbarer Betrag überschritten	fatal
	14	Kartenummer nicht bekannt	fatal
	21	Kein Vorgang vorgenommen	fatal
	24	Filetransfer nicht unterstützt	fatal
	29	Filetransfer nicht erfolgreich	fatal
	30	Formatfehler	fatal
	31	Kartenherausgeber nicht zugelassen	fatal
	33	Verfalldatum überschritten	fatal
	34	Manipulationsverdacht	fatal
	40	Funktion ungültig	fatal
	43	Gestohlene Karte; Karte einbehalten	fatal
	50	Autorisierung doppelt	fatal
	51	Limit überschritten; „Doch“-Funktion möglich	fatal
	54	EC-Chipkarte nicht mehr gültig	fatal
	55	PIN falsch	fatal
	56	Karte nicht im Bestand	fatal
	57	Andere Karte als bei der Bezugstransaktion benutzt wurde (bei Storno; Kreditkarten auch bei Erweiterungsreservierung, Endsummenbuchung, Trinkgeld-Update)	fatal
	58	Terminal-ID nicht erkannt	fatal
	59	Manipulationsverdacht bei EC-Chipkarte	fatal
	61	Karte gesperrt in lokaler Sperrliste	fatal
	62	Karte gesperrt	fatal
	64	Transaktionsbetrag weicht von Bezugstransaktion ab (bei Storno; Kreditkarten auch bei Endsummenbuchung)	fatal
	65	Limit der Transaktionsfrequenz überschritten; „Doch“-Funktion möglich	fatal
	75	Zulässige Anzahl von PIN-Eingaben überschritten	fatal
	76	Schlüsselindex unzulässig bzw. Fehler in BMP 57	fatal

CheckCredit-CardResult	CheckCreditCardResultText		Ergebnisklasse
	77	PIN-Eingabe erforderlich	fatal
	78	Sequenzfehler in BMP 62	fatal
	80	Betrag ist nicht verfügbar	fatal
	81	Initialisierung fehlerhaft; Wiederholung erforderlich	fatal
	82	(Vor-)Initialisierung unzulässig (Terminal gesperrt)	fatal
	83	PIN-Pad-Wechsel nicht möglich	fatal
	84	Datex-P-Nr. falsch	fatal
	85	Ablehnung vom Kreditkarteninstitut	fatal
	86	Stammdaten unbekannt; bei Init1 OPT-fähiger Terminals auch: ZKA-Nummer falsch	fatal
	87	Terminal (Pin-Pad) unbekannt	fatal
	88	Beim Laden gegen EC-Karte: PIN nicht aktiv	fatal
	89	CRC falsch	fatal
	91	Kartenherausgeber oder Netz nicht verfügbar	fatal
	92	Kartentyp wird nicht vom Autorisierungssystem verarbeitet	fatal
	96	Verarbeitung zur Zeit nicht möglich	fatal
	97	Sicherheitsverletzung: MAC-Prüfung zeigt Fehlerzustand	fatal
	98	Datum / Uhrzeit nicht plausibel	fatal
	99	Fehler in der PAC-Verschlüsselung	fatal
99 und 999	Nicht klassifizierte technische Fehler im Payment		recoverable



Diese Liste der Original-Fehlertexte des Netzbetreibers hat keinen Anspruch auf Vollständigkeit.



Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:

Tag	Daten typ	Feld- länge	Pflicht feld	Bemerkung
CreditCard				
CreditCard- Number	an	19	Ja	Kreditkartennummer
CreditCard- Holder	an		Nein	(Reserviert für weitere Funktio- nen)
CreditCard- Expiration- Date	an	6	Ja	Verfalldatum Format: MM/YY
CreditCard- Type	an	1	Ja	Kreditkartentyp V: Visa A: AMEX E: EUROCARD/MasterCard D: Diners J: JCB G: Galeria
CVV	n	4	Nein	Kreditkartenprüfnummer Ziffernfolge mit max. vier Stellen (nähere Informationen siehe Abschnitt 1.3)

```

<?xml version="1.0" encoding="ISO-8859-1"?>

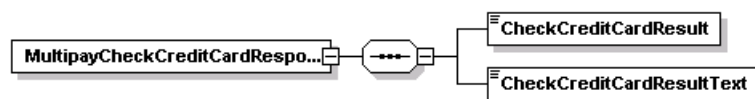
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000124-00001@www.client.url
    
```

```

        </communicationId>
        <Client>
            <ClientName>123</ClientName>
            <SalesChannel>123</SalesChannel>
            <Branch>123</Branch>
        </Client>
        <retry>true</retry>
    </header>
    <request>
        <MultipayCheckCreditCardRequest>
            <CreditCard>
                <CreditCardNumber>49494949494949</CreditCardNumber>
                <CreditCardExpirationDate>12/02</CreditCardExpirationDate>
                <CreditCardType>V</CreditCardType>
            </CreditCard>
        </MultipayCheckCreditCardRequest>
    </request>
</PAGOxml>

```

MultipayCheckCreditCardResponse ist wie folgt aufgebaut:



Tag	Daten typ	Feld- länge	Bemerkung
CheckCredit- CardResult	n	5	Ergebnis als Nummer (siehe Informationen weiter oben)
CheckCredit- CardResultText	an	40	Ergebnis als Text

```

<?xml version="1.0" encoding="ISO-8859-1"?>

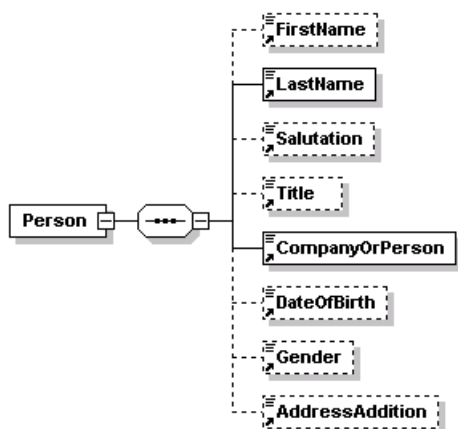
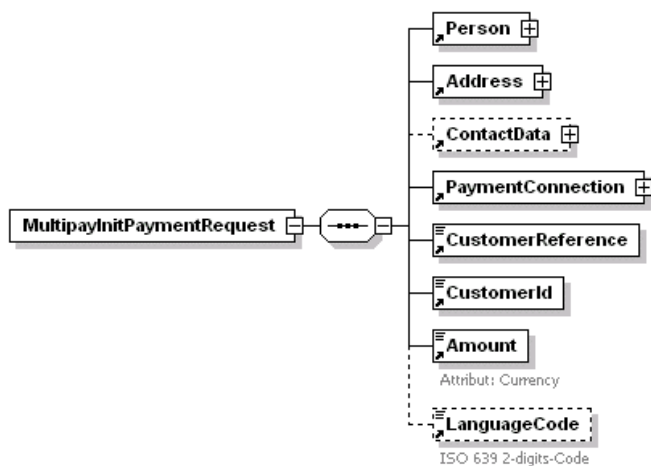
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
<header>
<communicationId>2001-12-31T12:31:30+02:00-001122334456-00601@www.pago.de
</communicationId>
<Server>PagoServer</Server>
<retry>false</retry>
<inReplyTo>2001-12-31T12:02:30+02:00-000000000124-00001@www.client.url
</inReplyTo>
</header>
<response>
<MultipayCheckCreditCardResponse>
    <CheckCreditCardResult>-6</CheckCreditCardResult>
    <CheckCreditCardResultText>INVALID CHECKSUM
    </CheckCreditCardResultText>
</MultipayCheckCreditCardResponse>
</response>
</PAGOxml>

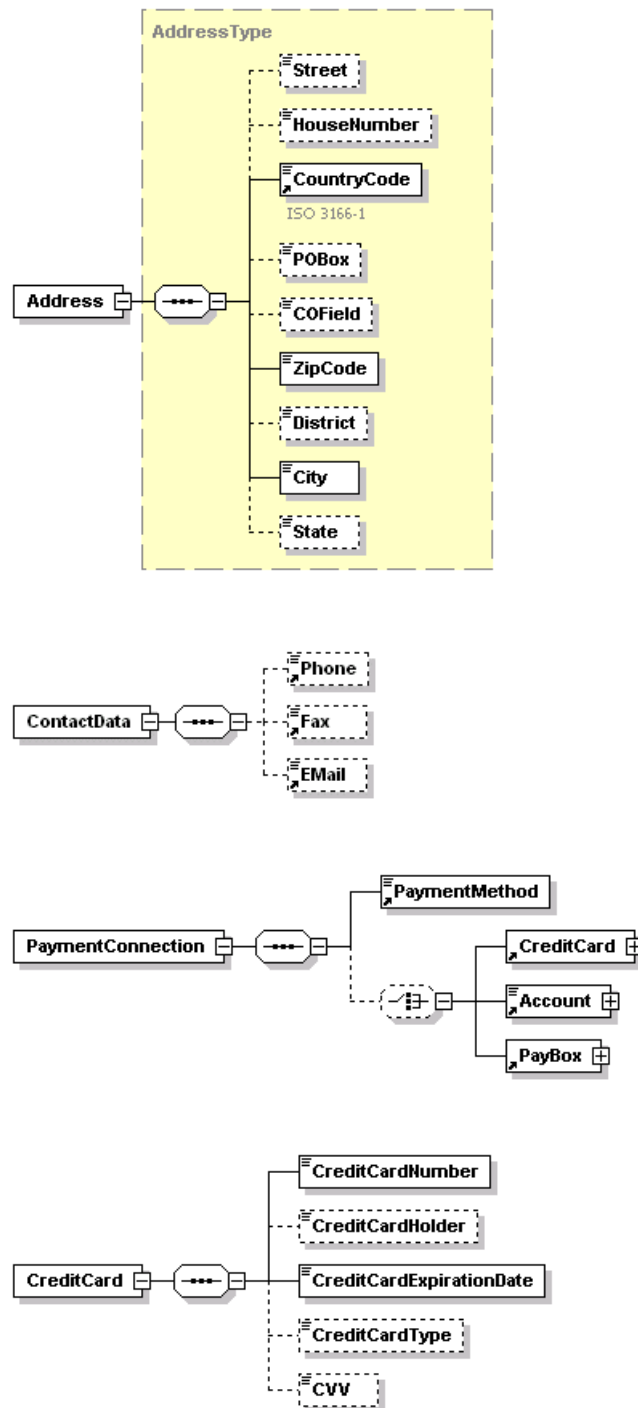
```

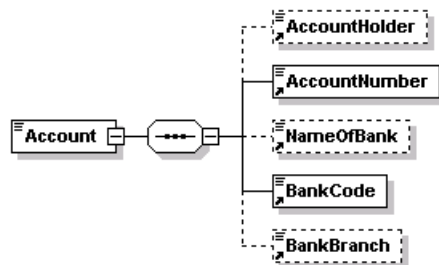
8.5.3 Initiieren von Zahlungen

Zahlungen werden mit **MultipayInitPaymentRequest** initiiert. Der Vorgang wird durch eine Fremdbelegnummer eindeutig gekennzeichnet.

Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:







Tag	Daten typ	Feld- länge	Pflicht feld	Bemerkung
Mode	n	3	Ja	Modus 2: Anlegen des Vorgangs bei allen Zahlarten und Prüfung/Autorisie- rung bei ELV , KKE mit späterem Geldfluss Weitere Informationen zum Modus und dem damit verbundenen Busi- nessmodell finden Sie in Kapitel 3.
Person				
FirstName	an	30	Nein	Vorname
LastName	an	30	Ja	Nachname oder Firmenname
Salutation	an	10	Nein	Anrede, z. B. „Herr“, „Frau“, „Firma“
Title	an	20	Nein	Titel, z. B. „Dr.“, „Prof.“
CompanyOr- Person	char	1	Ja	F (Firma) oder P (Person)
DateOfBirth	an	10	Nein	Geburtsdatum Format: YYYYMMDD oder DD.MM.YYYY
Gender	char	1	Nein	Geschlecht Fixwert: „“ (Leerstring)
Address- Addition	an	60	Nein	Adresszusatz z. B. „7. Stock“, „bei Maier“, „Firma Müller“, z. H. Herrn Schmidt“
Adresse				
Street	an	46	Ja	Straßenname
HouseNumber	an	35	Ja	Hausnummer

Tag	Daten typ	Feld- länge	Pflicht feld	Bemerkung
CountryCode	an	3	Ja	gültiges Länderkennzeichen nach ISO 3166 (siehe Kapitel 10)
POBox	n	35	Nein	Postfach-Nummer
COField	an	40	Nein	Großempfänger-Name
ZipCode	an	9	Ja	Postleitzahl
District	an	40	Nein	Ortsteil
City	an	40	Ja	Ortsname
State	an	40	Nein	Staat (US-Bundesstaat)
ContactData				
Phone	an	20	Nein	Telefonnummer laut Pago Bill Muss mit „+“ und Ländervorwahl beginnen (z. B. „+49“).
Fax	an	20	Nein	Faxnummer laut Pago Bill Muss mit „+“ und Ländervorwahl beginnen (z. B. „+49“).
EMail	an	80	Nein	E-Mail-Adresse
PaymentConnection				
PaymentMethod	an	3	Ja	Mit Pago abgesprochene Zahlart, z. B.: ELV Lastschriftverfahren (DTA) KKE Kreditkarteneinzug
CreditCard (nur bei KKE)				
CreditCard- Number	an	19	Ja	Kreditkartennummer (ohne Leerzeichen)
CreditCard- Holder	an		Nein	(Reserviert für weitere Funktio- nen)
CreditCard- Expiration- Date	an	6	Ja	Verfalldatum Format: MM/YY
CreditCard- Type	an	1	Ja	Kreditkartentyp V: VISA A: AMEX E: EUROCARD/MasterCard D: DINERS J: JCB G: GALERIA

Tag	Daten typ	Feld- länge	Pflicht feld	Bemerkung
CVV	n	4	Nein	Kreditkartenprüfnummer Ziffernfolge mit max. vier Stellen (nähere Informationen siehe Abschnitt 10)
Account (nur bei ELV)				
Account-Holder	an	30	Nein	Kontoinhaber (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)
Account-Number	an	10	Ja	Kontonummer (nur bei Zahlart ELV)
NameOfBank	an	30	Nein	Bankname (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)
BankCode	an	8	Ja	Bankleitzahl (nur bei Zahlart ELV)
BankBranch	an	30	Nein	Bankfiliale (nur bei Zahlart ELV) Fixwert: „ " (Leerstring)
Customer-Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz unter ClientName / Sales-Channel / Branch (siehe auch Abschnitt 8.5) Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
CustomerId	an	20	Ja	Ihre eindeutige Kundenreferenz unter ClientName / Sales-Channel / Branch
Amount				
Amount	n	10,2	Ja	positiver Bruttobetrag inkl. MwSt
Currency	an	3	Ja	Währung gemäß ISO 4217 (z. B. EUR, USD) Currency ist ein Attribut von Amount
LanguageCode	an	3	Nein	Sprachkennzeichen des Kunden Verwendung zukünftig im Mahnwesen

Details über die Verwendung der Adressfelder bei Rechnungsdruck und Mahnwesen finden Sie in dem Dokument „PrintLayout“.



Wiederholte Aufrufe zur selben Fremdbelegnummer erhalten das gleiche alte Ergebnis (Returncode, Message).

Beispiel:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000125-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>true</retry>
  </header>
  <request>
    <MultipayInitPaymentRequest Mode="2">
      <Person>
        <FirstName>HANS</FirstName>
        <LastName>Debtor</LastName>
        <Salutation>Herr</Salutation>
        <CompanyOrPerson>P</CompanyOrPerson>
      </Person>
      <Address>
        <Street>reperbaan</Street>
        <HouseNumber>1</HouseNumber>
        <CountryCode>De</CountryCode>
        <ZipCode>2000</ZipCode>
        <City>hammburg</City>
      </Address>
      <PaymentConnection>
        <PaymentMethod>KKE</PaymentMethod>
        <CreditCard>
          <CreditCardNumber>4321432143214321</CreditCardNumber>
          <CreditCardExpirationDate>12/01</CreditCardExpirationDate>
        </CreditCard>
      </PaymentConnection>
      <CustomerReference>987654323</CustomerReference>
      <CustomerId>13447</CustomerId>
      <Amount Currency="EUR">10.0</Amount>
    </MultipayInitPaymentRequest>
  </request>
</PAGOxml>
```

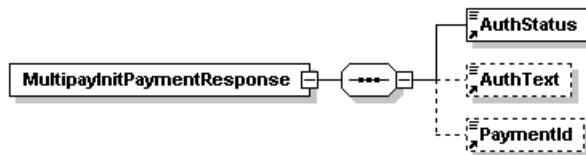
Für die Zahlung auf Rechnung wird nur **PaymentMethod** angegeben. Das folgende Beispiel soll diesen Sachverhalt verdeutlichen.

```
<request>
  <MultipayInitPaymentRequest Mode="2">
    <Person>
      <FirstName>HANS</FirstName>
      <LastName>Debtor</LastName>
      <Salutation>Herr</Salutation>
```



```
<CompanyOrPerson>P</CompanyOrPerson>
</Person>
<Address>
  <Street>reperbaan</Street>
  <HouseNumber>1</HouseNumber>
  <CountryCode>De</CountryCode>
  <ZipCode>2000</ZipCode>
  <City>hamburg</City>
</Address>
<PaymentConnection>
  <PaymentMethod>REC</PaymentMethod>
</PaymentConnection>
<CustomerReference>987654323</CustomerReference>
<CustomerId>13447</CustomerId>
<Amount Currency="EUR">10.0</Amount>
</MultipayInitPaymentRequest>
</request>
```

Die Antwort auf **MultipayInitPaymentRequest** ist **MultipayInitPaymentResponse**. Darin wird der Autorisierungsstatus mit einem Zusatztext geliefert.



MultipayInitPaymentResponse hat folgenden Aufbau:

Tag	Daten typ	Feld- länge	Bemerkung
MultipayInitPaymentResponse			
AuthStatus	n	1	Autorisierungsstatus 0: OK 1: abgelehnt (nur bei Zahlarten mit Autorisierung)
AuthText	an	20	Text zum Autorisierungsstatus
PaymentId	n	38	ID für Rückfragen bei Pago (wird nur intern bei Pago verwendet)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-10-23T16:10:19+02:00">
  <header>
    <communicationId>2001-12-31T12:30:32+02:00-001122334457-00601@www.pago.de
  </communicationId>
  <Server>PagoServer</Server>
  <retry>false</retry>
```

```
<inReplyTo>2001-12-31T12:02:30+02:00-000000000125-00001@www.client.url
</inReplyTo>
</header>
<response>
<MultipayInitPaymentResponse>
  <AuthStatus>0</AuthStatus>
  <AuthText>AUTH CODE:6236</AuthText>
  <PaymentId>3422</PaymentId>
</MultipayInitPaymentResponse>
</response>
</PAGOxml>
```

Wenn eine Autorisierung abgelehnt wird, hat der Tag **response** beispielhaft folgenden Aufbau:

```
<response>
<MultipayInitPaymentResponse>
  <AuthStatus>1</AuthStatus>
  <AuthText>AUTHORIZATION DECLINED</AuthText>
  <PaymentId>3411</PaymentId>
</MultipayInitPaymentResponse>
</response>
```

Fehlercode	Grund	Abhilfe
101.001	Kein gültiger numerischer Wert	Zahl verwenden
101.004	Currency ist für diesen Dienst Pflichtfeld	Currency muss gefüllt sein
101.005	Betrag war <=0	Positiven Betrag verwenden
101.006	Mode-Attribut war <0 oder >3	0, 1, 2 oder 3 verwenden
101.007	PaymentMethod passt nicht zum Mode	
101.008	Parameterlänge zu lang oder zu kurz	Parametername steht im Twister-Log
101.009	go_to_payment Ergebnis war „999“	Autor fragen
101.500	Ungültiger Mandant	
101.501	Ungültige Personendaten	
101.502	Ungültige Adressdaten	
101.504	Ungültige Währung	
101.506	Unbekannter Mandant	
101.509	Transaktionsart fehlt	
101.511	Zahlart ist unbekannt	
101.512	Stammdaten sind ungültig	

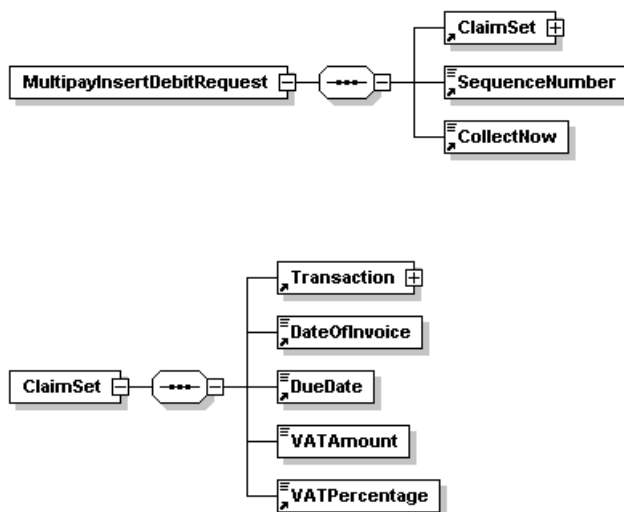
Fehlercode	Grund	Abhilfe
101.513	Bereits autorisiert	
101.514	Autorisierung in Bearbeitung	
101.515	Auftrag abgelaufen	
101.517	Sequenznummer ist ungültig	

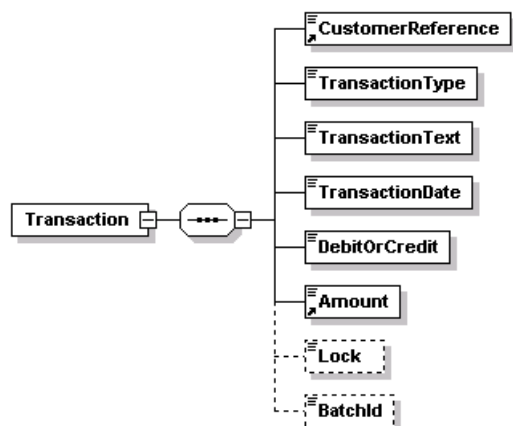
8.5.4 Einstellen von Gutschriften mit InsertDebit

Eine Gutschrift stellen Sie im Modus 2 mit **MultipayInsertDebitRequest** ein.



Achten Sie darauf, dass die Summe der eingestellten Gutschriften nicht höher ist als der bei **MultipayInitPaymentRequest** angegebene Betrag.





Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:

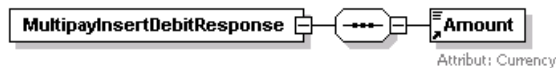
Tag	Daten typ	Feld- länge	Pflicht feld	Bemerkung
ClaimSet				
Transaction				
Customer-Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz wie in MultipayInit-PaymentRequest angegeben Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
Transaction-Type	an	6	Ja	Transaktionsart GS = Gutschrift RT = Retoure ! Die Transaktionsart bestimmt nicht, ob der Betrag im Soll oder Haben gebucht wird.
Transaction-Text	an	30	Ja	Transaktionstext für Buchhaltung / Kontoauszug
Transaction-Date	an	10	Ja	Transaktionsdatum (Kaufdatum) Format: YYYYMMDD oder DD.MM.YYYY

Tag		Daten typ	Feld- länge	Pflicht feld	Bemerkung
	DebitOrCredit	an	1	Ja	Soll oder Haben S : Soll (debit)(FD/VD/GS/RT) Für alle Buchungen ist S zu verwenden. H : Haben (credit) nur intern bei Pago
	Amount				
	Amount	n	10,2	Ja	Bruttobetrag inkl. MwSt + : Forderung oder Stornierung einer Gutschrift - : Gutschrift oder Stornierung einer Forderung
	Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPayment- Request
	Lock			Nein	Reserviert
	BatchId	an	30	Nein	Batch-ID Dient bei Bedarf zur Gruppierung von Sätzen (nur nach Absprache)
	DateOfInvoice	an	10	Ja	Rechnungsdatum Format: YYYYMMDD oder DD.MM.YYYY
	DueDate	an	10	Ja	Fälligkeitsdatum Format: YYYYMMDD oder DD.MM.YYYY
	VATAmount	n	10,2	Ja	Mehrwertsteuerbetrag oder -1 Bei -1 keine Angabe. Wird von Pago berechnet, wenn ein MwSt- Satz angegeben ist.
	VATPercentage	n	4,2	Ja	Mehrwertsteuer in Prozent -1 : keine Angabe (Pago bucht keine MwSt im Debitorensystem)
	SequenceNumber	n	6	Ja	Sequenznummer Muss mit „1“ beginnen und wächst mit jedem Folgeaufruf um 1 (siehe auch Abschnitt 8.3.2).

Tag	Daten typ	Feld- länge	Pflicht feld	Bemerkung
CollectNow	n	1	Ja	Der Tag CollectNow ist im Modus 2 für Forderungen gegenstandslos und sollte deshalb = false sein. Wenn eine Gutschrift eingestellt wird, führt CollectNow = true zur sofortigen Auszahlung. Bei CollectNow = false muss die Auszahlung über Multipay-SettleAccountRequest ausgelöst werden.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="Thu May 17 13:42:15 2001">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>>false</retry>
  </header>
  <request>
    <MultipayInsertDebitRequest>
      <ClaimSet>
        <Transaction>
          <CustomerReference>gfdgfdgfd54543453</CustomerReference>
          <TransactionType>FD</TransactionType>
          <TransactionText>Beispiel Forderung</TransactionText>
          <TransactionDate>30.12.2002</TransactionDate>
          <DebitOrCredit>S</DebitOrCredit>
          <Amount Currency="EUR" >99.99</Amount>
          <BatchId></BatchId>
        </Transaction>
        <DateOfInvoice>16.07.2001</DateOfInvoice>
        <DueDate>16.10.2001</DueDate>
        <VATAmount>13.79</VATAmount>
        <VATPercentage>16.0</VATPercentage>
      </ClaimSet>
      <SequenceNumber>1</SequenceNumber>
      <CollectNow>>false</CollectNow>
    </MultipayInsertDebitRequest>
  </request>
</PAGOxml>
```

Als Rückgabewert auf **MultipayInsertDebitRequest** wird neben dem Ergebnis der einzuziehende Betrag gemeldet („0“ bei **CollectNow = false**).



Tag	Daten typ	Feld-länge	Pflicht feld	Bemerkung
Amount				
Amount	n	10,2	Ja	Einzuziehender Betrag
Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPayment-Request

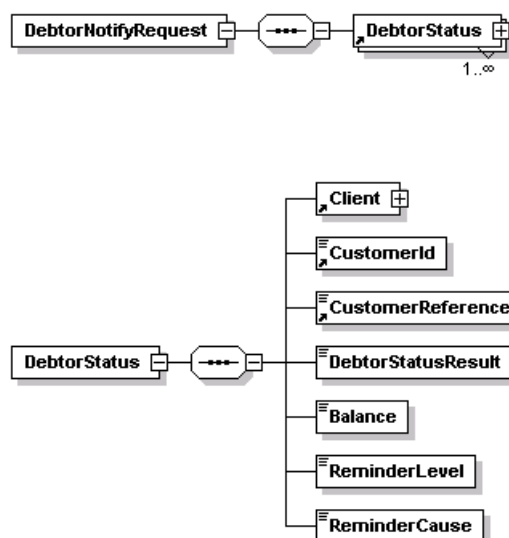
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml timeStamp="2001-10-30T16:45:25+01:00" version="2.0">
  <header>
    <communicationId>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de
    </communicationId>
    <Server> PagoServer </Server>
    <retry> false </retry>
    <inReplyTo>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </inReplyTo>
  </header>
  <response>
    <MultipayInsertDebitResponse>
      <Amount Currency="EUR"> 0 </Amount>
    </MultipayInsertDebitResponse>
  </response>
</PAGOxml>
```

Fehlercode	Grund	Abhilfe
103.001	Nicht-boole'scher Wert	true oder false verwenden
103.002	Kein gültiger numerischer Wert	Zahl verwenden
103.003	Parameterlänge zu lang oder zu kurz	Parametername steht im Twister-Log
103.500	Ungültige Mandanten-Identifikation	
103.505	Abweichende Währung	
103.506	Ungültige Transaktionsart	
103.507	Ungültige Autorisierung	

Fehlercode	Grund	Abhilfe
103.508	Systemfehler	
103.509	Systemfehler	
103.513	Einlösungsfehler	
103.516	Ungültige Sequenznummer	

8.5.5 Empfangen des Debitorenstatus

Unabhängig von **MultipayInsertDebitRequest** kann sich der Debitorenstatus verändern. Der jeweils aktuelle Debitorenstatus wird (ohne Historie) von Pago via **Debtor-NotifyRequest** bereitgestellt. Dabei werden mit einem Request alle zur Verfügung stehenden Debitorenstatus versendet.



Tag	Daten typ	Feld- länge	Bemerkung
DebtorStatus			
Client			
ClientName	an	20	Name des Mandanten Konstante in Absprache mit Pago
SalesChannel	an	20	Vertriebslinie Konstante in Absprache mit Pago
Branch	an	20	Filiale Konstante in Absprache mit Pago

Tag	Datentyp	Feldlänge	Bemerkung
CustomerId	an	20	Ihre eindeutige Kundenreferenz unter ClientName / SalesChannel / Branch
Customer-Reference	an	20	Ihre eindeutige Vorgangsreferenz unter ClientName / SalesChannel / Branch (siehe auch Abschnitt 8.5) Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
DebtorStatus-Result	an	1	Debitorenstatus N : nicht bezahlt B : bezahlt, Vorgang ist ausgeglichen Z : überzahlt F : überfällig/in Mahnung, Vorgang ist nicht ausgeglichen (Ursache kann eine fehlgeschlagene Zahlung oder eine Rückforderung sein.)
Balance			
Balance	n	15,2	Saldo des Vorgangskontos Negativer Betrag: Guthaben Positiver Betrag: Forderung
Currency	an	3	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest
ReminderLevel	an	1	Aktuelle Mahnstufe 0 : keine Mahnung ... 4 : Inkasso A : Ausgebucht

Tag	Daten typ	Feld- länge	Bemerkung
ReminderCause	an	3	<p>Mahngrund</p> <p>RL0: keine Deckung</p> <p>RL1: Konto erloschen</p> <p>RL2: Konto-Nr. / Name nicht identisch (falsch oder Sparkonto)</p> <p>RL3: kein Abbuchungsauftrag bzw. keine Einzugsermächtigung</p> <p>RL4: Rückruf</p> <p>RL5: Widerspruch</p> <p>RL6: Rückgabe</p> <p>LSV: Zahlart Lastschrift, nicht einziehbar</p> <p>CB: Chargeback</p> <p>KKE: Zahlart Kreditkarte, nicht einziehbar</p>

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
<header>
  <communicationId> 2001-12-31T12:30:30+02:00-001122334455-00601@10.10.10.10
</communicationId>
  <Server>
    PagoServer
  </Server>
  <retry>
    true
  </retry>
</header>
<request>
  <DebtorNotifyRequest>
    <DebtorStatus>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
      <CustomerId>123456789001</CustomerId>
      <CustomerReference>abcdefg2001</CustomerReference>
      <DebtorStatusResult>B</DebtorStatusResult>
      <Balance Currency="EUR">0</Balance>
      <ReminderLevel>0</ReminderLevel>
      <ReminderCause> </ReminderCause>
    </DebtorStatus>
    <DebtorStatus>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
      <CustomerId>123456789002</CustomerId>
      <CustomerReference>abcdefg1999</CustomerReference>
      <DebtorStatusResult>B</DebtorStatusResult>
      <Balance Currency="EUR">0</Balance>
      <ReminderLevel>0</ReminderLevel>
      <ReminderCause> </ReminderCause>
    </DebtorStatus>
  </DebtorStatus>
</request>
```

```
<Client>
  <ClientName>123</ClientName>
  <SalesChannel>123</SalesChannel>
  <Branch>123</Branch>
</Client>
<CustomerId>123456789000</CustomerId>
<CustomerReference>abcdefg2000</CustomerReference>
<DebtorStatusResult>F</DebtorStatusResult>
<Balance Currency="EUR">34,95</Balance>
<ReminderLevel>A</ReminderLevel>
<ReminderCause>LSV</ReminderCause>
</DebtorStatus>
</DebtorNotifyRequest>
</request>
</PAGOxml>
```

Auf **DebtorNotifyRequest** wird von Pago ein **DebtorNotifyResponse** erwartet. Erst mit dem Eingang eines leeren **DebtorNotifyResponse** ohne **errorNotification** gilt die Zustellung des Status als abgeschlossen.

DebtorNotifyResponse

Das Tag **DebtorNotifyResponse** hat keine definierten Elemente.

Tag	Daten typ	Feld-länge	Bemerkung
DebtorNotifyResponse			

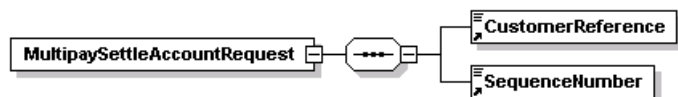
Das folgende Beispiel zeigt einen möglichen **DebtorNotifyResponse**:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml timeStamp="2001-10-30T16:45:25+01:00" version="2.0">
  <header>
    <communicationId>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url>
    <communicationId>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
    </communicationId>
    <retry>false</retry>
    <inReplyTo>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de
    </inReplyTo>
  </header>
  <response>
    <DebtorNotifyResponse/>
  </response>
</PAGOxml>
```

8.5.6 Auszahlung einer Gutschrift

Nach dem Einstellen von einer oder mehreren Gutschriftbuchungen (mit dem Parameter **CollectNow = false**) wird mit **MultipaySettleAccountRequest** die Auszahlung des Forderungssaldos freigegeben. Die Auszahlung bei negativem Forderungssaldo erfolgt in der Regel asynchron. Ergebnisse werden von Pago über **DebtorNotifyRequest** gemeldet.

Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:



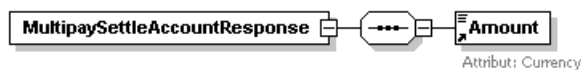
Tag	Daten typ	Feld- länge	Pflicht feld	Bemerkung
Customer- Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz wie in MultipayInitPayment- Request angegeben Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
SequenceNumber	n	6	Ja	Sequenznummer Muss mit „1“ beginnen und wächst mit jedem Folgeaufruf um 1 (siehe auch Abschnitt 8.3.2).

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>false</retry>
  </header>
  <request>
    <MultipaySettleAccountRequest>
      <CustomerReference>gfdgfdgfd54543453</CustomerReference>
      <SequenceNumber>2</SequenceNumber>
    </MultipaySettleAccountRequest>
  </request>
</PAGOxml>
  
```

```
</MultipaySettleAccountRequest>
</request>
</PAGOxml>
```

Zu **MultipaySettleAccountRequest** wird mittels **MultipaySettleAccount-Response** der einzuziehende Betrag zurückgemeldet.



Tag	Daten typ	Feld- länge	Bemerkung
Amount			
Amount	n	10,2	Einzuziehender Betrag
Currency	an	3	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest

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

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>2001-12-31T12:34:30+02:00-001122334459-00601@www.pago.de
    </communicationId>
    <Server> PagoServer </Server>
    <retry> false </retry>
    <inReplyTo>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </inReplyTo>
  </header>
  <response>
    <MultipaySettleAccountResponse>
      <Amount Currency="EUR"> 99.99 </Amount>
    </MultipaySettleAccountResponse>
  </response>
</PAGOxml>
```

9 Multipay mit Modus GUI

In diesem Kapitel wird die Handhabung von Pago Multipay mit Modus GUI (Graphical User Interface) beschrieben. Informationen über die anderen Einsatzmöglichkeiten von Multipay durch den Einsatz anderer Modi finden Sie in Kapitel 2 „Entscheidungshilfen für die angemessene Payment-Methode“.

Kennzeichnend für den Modus GUI sind folgende Merkmale:

- Payment mit Einsatz eines SSL-Form für die Eingabe der Bezahlinformationen durch den Kunden.
- Je nach Mandantenkonfiguration bei der Einrichtung durch Pago wird der Betrag entweder sofort autorisiert und später eingezogen (wie bei Modus 0 „deferred“) oder sofort eingezogen (wie bei Modus 1 „immediate“).
- Alle von Pago unterstützten Zahlarten können genutzt werden.
- Paybox ist immer eine verzögerte Zahlart („deferred“; Modus 0), d. h. im ersten Schritt wird eine Autorisierung durchgeführt, und zu einem späteren Zeitpunkt mit der Forderung die Zahlung ausgelöst.
- Für die Online Überweisung wird ein Java Applet der Firma Stackbox eingesetzt. Dieses Applet wird zur Eingabe der Überweisungsdaten von Pago geladen.

9.1 Anwendungsregeln für den Einsatz von Multipay im Modus GUI

Die gewünschten Zahlarten müssen Sie mit Pago absprechen. Gemäß dieser Absprache richtet Pago Bankkonten, Paybox- und Kreditkarten-Parameter ein und teilt sie, soweit notwendig, dem aufrufenden System mit.

Ein Auftrag wird mit dem Ziel eingestellt, einen Geldbetrag in einer vorher bekannten Maximalhöhe vom Kunden zu erhalten. Beim Anlegen des Auftrags wird dieser Gesamtbetrag an Pago übergeben. Für den Betrag gelten die drei folgenden Regeln:

- (§1) Die Summe aller Einzüge pro Auftrag darf den Gesamtbetrag nicht überschreiten.
- (§2) Der Gesamtbetrag darf nicht nachträglich erhöht werden.
- (§3) Die Summe aller Gutschriftenbeträge darf die Summe aller Forderungsbeträge nicht überschreiten.

Wenn die Forderung erst zu einem späteren Zeitpunkt entsteht, kann die Zahlung zum Zeitpunkt der Auftragseinstellung lediglich autorisiert werden. Bei Online-Zahlarten wird der Einzug durch den Aufruf von **MultipaySettleAccountRequest** zum Zeitpunkt der (Teil-)Lieferung initiiert, und zwar in Höhe der bis zu diesem Zeitpunkt verbuchten noch nicht eingezogenen Forderungen. Wenn für den Vorgang lediglich eine Forderung benötigt wird, kann der Aufruf von **MultipaySettleAccountRequest** entfallen. In diesem Fall muss beim Aufruf von **MultipayInsertDebitRequest** der Parameter **CollectNow = true** verwendet werden.

Nicht nur die Forderungen, die aus der Bestellung des Kunden hervorgehen, können storniert werden, sondern auch Gutschriften, die z. B. bei Beschädigung der Ware oder bei Retoure ausgestellt werden. Darüber hinaus wird zwischen Buchungen und Stornos unterschieden. Forderungsbuchungen und Gutschriftenstornos werden in positives Soll gebucht, Forderungsstornos und Gutschriftenbuchungen in negatives Soll. Dabei gelten folgende Regeln:

(§4) Die Summe der Stornos darf pro Satzart die Summe der Buchungen nicht überschreiten.

(§5) Die Summe aller Forderungsbuchungen darf den Gesamtbetrag nicht überschreiten.

! *Ein einmal stornierter Betrag (Forderungsstorno, Retoure, Versandkostenstorno) kann nicht mehr gefordert werden.
Eine Gutschrift (aus Kulanz) kann nur auf existierende Zahlungseingänge gewährt werden. Eine Gutschrift kann auch wieder storniert werden, die Regeln in §1 bis §5 sind jedoch immer verbindlich.*

! *Je nach Zahlart gibt es weitere Einschränkungen bezüglich Gutschriften und Stornos. Deshalb sollten Sie Forderungen möglichst zu einem Zeitpunkt einstellen, an dem Sie prozessbedingte Rückabwicklungen abschließen können.*

9.2 Umgang mit Retouren und Gutschriften

Mit folgender Vorgehensweise bleiben die Warenbewegungen und Zahlungen je Vorgang bei allen Parteien übersichtlich und nachvollziehbar. Rückbelastungen aufgrund unklarer Abbuchungen des Kunden können reduziert werden.

Situation 1 Dem Kunden wird nach einer Warenretoure Ersatz mit gleichem Wert geliefert:

Diese Situation ist für den Vorgang in Multipay nicht relevant.

Situation 2 Dem Kunden wird nach einer Warenretoure Ersatz mit geringerem Wert geliefert. Zum Ausgleich soll er eine Gutschrift erhalten:

Die Gutschrift über den gewählten Betrag wird mittels **MultipayInsertDebitRequest** angelegt und, je nach Bedarf, entweder sofort oder mit **MultipaySettleAccount** zur Auszahlung gebracht.

Situation 3 Dem Kunden wird nach einer Warenretoure der gezahlte Betrag erstattet. Zu einem späteren Zeitpunkt wird erneut eine Ware an ihn gesandt:

Mit der Warenretoure sollte der zugehörige Vorgang geschlossen werden. Dazu wird mit **MultipayInsertDebitRequest** eine Retoure eingestellt. Der Gutschriftsbetrag wird nach Bedarf sofort oder mit dem nächsten

MultipaySettleAccount ausgezahlt.

Für die nächste Bestellung wird mit **MultipayInitPayment** ein neuer Vorgang angelegt.

9.3 Transaktionssicherheit

9.3.1 Aufrufwiederholung bei Verbindungsabbruch

In der Verbindung zwischen dem Mandantensystem und Pago kann es vorkommen, dass der Aufrufende kein Ergebnis erhält. Das kann u. a. aus zwei Gründen der Fall sein:

- Der Empfänger hat den Aufruf nicht erhalten.
- Der Empfänger hat den Aufruf erhalten und verarbeitet, aber die Rückmeldung ist nicht beim Absender angekommen.

Um solche Vorfälle zu vermeiden, besitzt die Anbindung über LocalServer und XML einen speziellen Retry-Mechanismus. Dieser Mechanismus identifiziert jede Kommunikation einer Transaktion eindeutig mit Hilfe der **communicationId**.

Wird der Response nicht innerhalb einer definierten Zeitspanne empfangen, die über die Konfigurationsvariable **Timeout** definiert wird, sollte der Request mit **retry = true** und der gleichen **communicationId** erneut gesendet werden.

Mit dem Retry können drei Situationen eintreten:

- Wenn Pago den Request vor dem Retry nicht erhalten hat, wird der Request nach dem Retry von Pago verarbeitet und der entsprechende Response zurückgesendet.
- Wenn Pago den Request vor dem Retry erhalten und auch verarbeitet hat, wird nach dem Retry das Ergebnis dieser Verarbeitung im Response zurückgesendet.
- Wenn Pago den Request vor dem Retry erhalten hat und die Verarbeitung noch andauert, wird nach dem Retry im Response die Meldung zurückgesendet: **this transaction is still in progress**. Dann muss der Request mit einem weiteren Retry wiederholt werden.

Das folgende Beispiel zeigt den Response im Falle der noch andauernden Verarbeitung. Nähere Informationen hierzu finden Sie in Abschnitt 5.2 „Header-Block“.

```
</header>
<response>
  <errorNotification type="recoverable" issuer="PagoServer"
    service="MultipayInitGuiPayment">
    <errorText> <![CDATA[this transaction is still in progress]]> </errorText>
    <errorNumber> 30007 </errorNumber>
  </errorNotification>
</response>
</PAGOxml>
```


Wenn das Ergebnis der Verarbeitung vorliegt, kann es in eine der folgenden Klassen eingestuft werden:

- 1) Erfolgreiche Durchführung
- 2) Die Transaktion wurde mit nicht behebbaren Fehlern durchgeführt (**fatal**). Eine Wiederholung wird kein anderes Ergebnis hervorbringen. Daher sollten keine Wiederholungsaufrufe durchgeführt werden.
- 3) Die Transaktion wurde mit einem Fehler durchgeführt, der zu einem späterem Zeitpunkt möglicherweise nicht mehr auftritt. Zur Kennzeichnung des möglichen späteren Wiederholungsversuchs wird der Fehler mit **recoverable** klassifiziert.
Wenn in einem Fehlerszenario die Bank bei einer Autorisierung nicht erreicht wird, kann der Betrag auch nicht bei der betreffenden Kreditkarte reserviert werden. Die Initialisierung ist nicht abgeschlossen. Die Transaktion wird mit negativen Ergebnis **recoverable** abgeschlossen.
Die Autorisierung kann zu einem späteren Zeitpunkt erneut ausgeführt werden. Der erneute Autorisierungsversuch wird mit alter **communicationId** und gleicher Fremdbelegnummer (**CustomerReference**) initialisiert.

Die Aufrufwiederholung ist mehrfach ausführbar. Wenn Sie also nach einem Aufruf kein Ergebnis zurückerhalten haben, müssen Sie im Zweifelsfall den Aufruf solange wiederholen, bis Sie Klarheit über das Ergebnis haben. Wählen Sie dabei bitte einen sinnvollen Zeitabstand zwischen den Wiederholungsanfragen. Einen Funktionsaufruf, der wegen Timeout nicht abgeschlossen werden konnte, können Sie sofort wiederholen. Wenn eines der angebundenen Systeme (z. B. ein Banksystem) nicht zur Verfügung steht, wiederholen Sie den Aufruf nach einigen Minuten.

9.3.2 Handhabung der Sequenznummer

Die Sequenznummer dient der Einhaltung der Verarbeitungsreihenfolge. Pro Fremdbelegnummer muss die Sequenznummer aufsteigend und lückenlos nummeriert werden. Die Auftragsinitialisierung trägt die implizite Sequenznummer „0“. Verwenden Sie deshalb bei der als nächstes aufgerufenen Methode die Sequenznummer „1“. Eine falsche Sequenznummer wird mit einem Fehler abgelehnt. Für weitere Informationen zu Fehlermeldungen siehe auch Abschnitt 5.5 „Fehlermanagement“.

9.3.3 Verhalten der Methoden bei Wiederholungsaufrufen

Nach einem Wiederholungsaufruf mit gesetztem Retry-Flag wird das Ergebnis der ersten erfolgreichen bzw. fatalen Verarbeitung im Response zurückgeliefert. Die folgende Angabe gilt für Transaktionen, die durch das Flag **retryable** wiederholbar sind:

```
<errorNotification issuer="PagoServer"
  service="MultipayInitPayment" type="retryable">
```

Die Funktionen aus Pago Multipay verhalten sich folgendermaßen:

MultipayInitGuiPaymentRequest wird identifiziert durch

- **ClientName**
- **SalesChannel**
- **Branch**
- **CustomerReference.**

Wiederholungsanfragen zu fehlgeschlagenen oder abgelehnten Zahlungen werden erneut durchgeführt. Wiederholungsanfragen zu akzeptierten Zahlungen liefern das vorher erhaltene Ergebnis zurück, wenn sie in den anderen Parametern übereinstimmen. Andernfalls werden sie abgelehnt.

MultipayInsertDebitRequest, **MultipaySettleAccountRequest** und **MultipayReduceAmountRequest** werden identifiziert durch

- **ClientName**
- **SalesChannel**
- **Branch**
- **CustomerReference**
- **SequenceNumber.**

Wiederholungsanfragen liefern das vorher erhaltene Ergebnis zurück. Anfragen mit falscher Sequenznummer werden abgelehnt.

9.4 Pago Risk Management Services in Multipay GUI

Im Modus GUI sind die Dienste der Pago Risk Management Services (Pago RMS) integriert.

Durch Pago Basischeck wird eine noch nicht geprüfte Adresse mit dem Schalter **InAddressOk = false** gekennzeichnet, bei der Initialisierung geprüft und gegebenenfalls korrigiert. Bei Korrekturen muss der Protokollstring ausgewertet werden, um zu entscheiden, ob die Adresse verwendet werden soll oder eine erneute Eingabe durch den Kunden erforderlich ist.

Die RMS-Typen Pago Basisscore, Pago Bonicheck und Pago Boniscore führen auf Basis der korrekten Adresse eine Risikobewertung des Kunden durch. Die korrekte Adresse kann entweder von vornherein bekannt sein oder durch Pago Basischeck ermittelt worden sein. Aufgrund der Risikobewertung bieten die genannten RMS-Typen dem Kunden eine Auswahl an Zahlarten an.

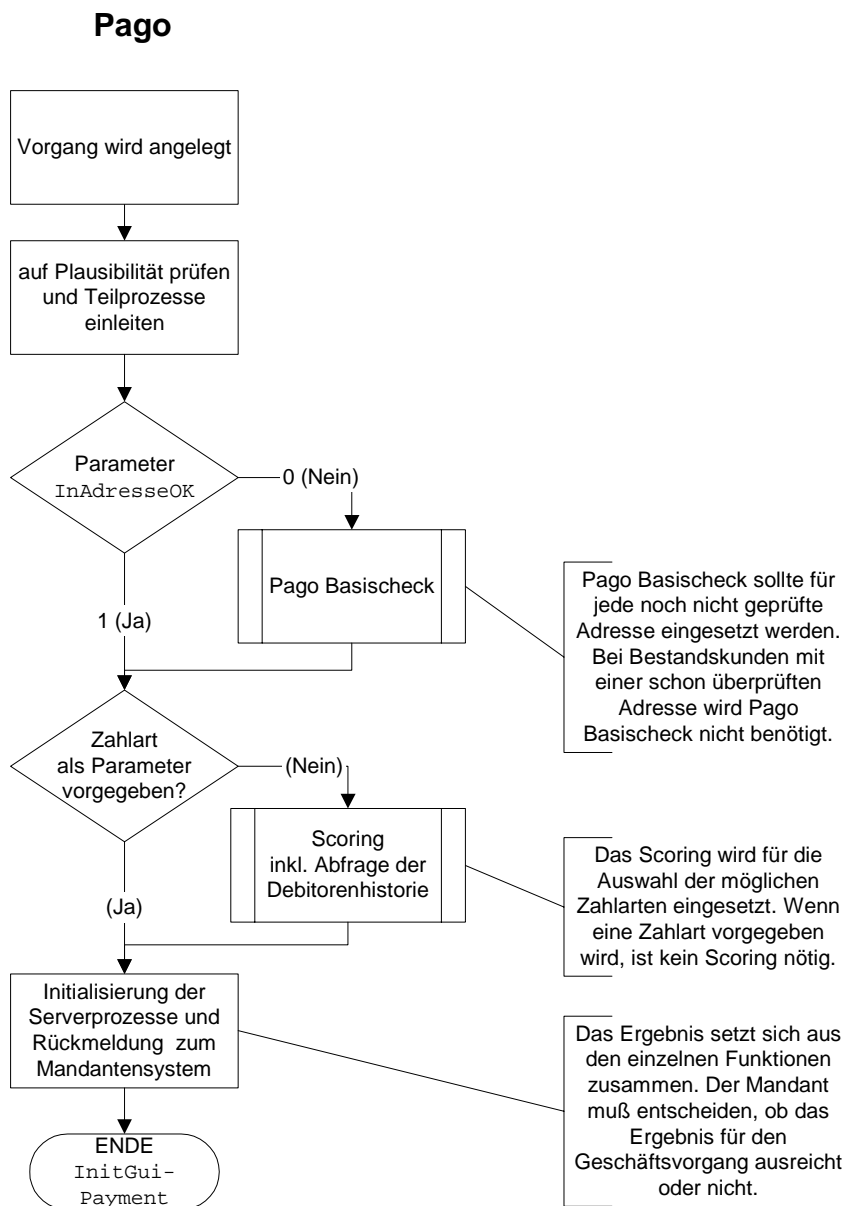
Mit Multipay sind verschiedene RMS-Typen verfügbar. Der Typ Pago Basisscore z. B. wird über das Tag **ScoreType = 1** verwendet.

Wenn dem Kunden unabhängig von einer Risikoprüfung nur eine bestimmte Zahlart angeboten werden soll (Kreditkarte, Lastschrift, etc.), führt Pago kein Scoring durch und erzeugt keine Zahlartenauswahl. Statt **ScoreType** gibt Pago RMS in diesem Fall **PaymentMethod** an. Hinter der zurückgegebenen URL befindet sich direkt die Zahlseite.

Mit **MultipayInitGuiPaymentRequest** werden zunächst die Pago Risk Management Services ausgeführt, es sei denn einer von beiden Diensten oder alle beide wurden durch Konfiguration und Parameter ausgeschaltet. Im Anschluss wird auf dem Server der Zahlungsprozess vollständig initialisiert.

Mit den Rückgabeparametern kann Ihr System entscheiden, ob die Kriterien für den Verkauf ausreichen (Ergebnis der Adressprüfung, Scorewert, etc.) und wenn ja, den Kunden auf die Pago-URL leiten. Anschließend folgt das eigentliche Payment.

Die folgende Schemazeichnung veranschaulicht den beschriebenen Ablauf:



! Beim Rückgabewert **false** von **outAddressOK** kann es der Fall sein, dass umfangreiche Korrekturen an der Schreibweise der eingegebenen Adresse vorgenommen wurden. Deshalb muss auf jeden Fall der Protokollstring ausgewertet werden, um die zurückgelieferten Felder einer Eindeutigkeitsprüfung zu unterziehen.

9.4.1 Der Protokollstring aus Pago Basischeck

Nach dem Prüfprozess wird ein achtstelliger Protokollstring an das Mandantensystem zurückgegeben. Jede der acht Stellen steht für einen bestimmten Aufrufparameter. Jedem Aufrufparameter ist ein Wert zugeordnet, der darüber Aufschluss gibt, was die Prüfung des jeweiligen Parameters ergeben hat.

! Wenn ein oder mehrere Aufrufparameter einen anderen Wert als „0“ haben, sollten Sie den Protokollstring unbedingt auswerten.

Die folgende Tabelle stellt den Aufbau des Protokollstrings dar:

	Reihenfolge der Aufrufparameter im String							
Aufrufparameter	Land	PLZ	Orts-name	Orts-teil	Str-name	Haus-nr.	Post-fach	Groß-empf.-Name

Jedem Aufrufparameter wird einer der folgenden Werte zugeordnet:

Wert	Bedeutung
-	Inhalt der Eingabe <i>nicht analysiert</i> bzw. nicht im Datenbestand verfügbar.
0	Eingabe <i>in Ordnung</i> , unverändert in Ausgabe übernommen.
1	Geringe Abweichungen in der Schreibweise festgestellt (z. B. Groß-/Kleinschreibung, Umlaute); Schreibweise <i>normiert</i> .
2	Größere Abweichungen festgestellt; Angabe <i>korrigiert</i> .
3	Eingabe als <i>Archiveintrag</i> identifiziert und in Ausgabe durch gültige Bezeichnung ersetzt.
4	Eingabe konnte nicht eindeutig identifiziert werden: <i>mehrdeutig</i> (mögliche Ursachen: Schreibfehler/Abkürzung oder fehlende/falsche Daten in abhängigen Feldern).
5	Eingabe <i>falsch</i> , nicht ermittelbar oder nicht korrigierbar (z. B. Postfach-Nr.).
6	Eingabe fehlte, konnte aber <i>ermittelt</i> werden.

Wert	Bedeutung
7	Eingabe fehlte und konnte auch <i>nicht ermittelt</i> werden.
8	Postfach abgetrennt.
9	Nummer war falsch.
A	Feld war zu kurz.
B	Hausnummer bei Archivorten nicht umsetzbar.
C	Normiert, nicht analysiert.
D	STAB ermittelt.
E	STAB unbekannt.
F	STAB Tab. fehlt.
G	Ort bzw. Straße wurde ersatzlos gestrichen.



Die Rückgabewerte A bis G sind interne Meldungen des Systems und müssen nicht ausgewertet werden.

9.4.2 Beispielergebnisse der Adressprüfung

Die Erfahrung mit Pago Basischeck hat gezeigt, dass es im wesentlichen drei Ergebnisgruppen der Adressprüfung gibt. Diese sind:

- Adresse ist in Ordnung und wurde ggf. normiert. Das kann bei folgenden Werten der Fall sein:

Aufrufparameter	Land	PLZ	Ort	Orts- teil	Straße	Haus- nr.	Post- fach	GE
Mögliche Werte:	0,1	0,1	0,1	0,1,6	0,1	-, 0, 1	-, 0	-, 0

Damit ist die Adressprüfung abgeschlossen. Korrekturen durch den Kunden sind nicht notwendig.

- Adresse wurde eindeutig identifiziert und entsprechend korrigiert. Das kann bei folgenden Werten der Fall sein:

Aufrufparameter	Land	PLZ	Ort	Orts- teil	Straße	Haus- nr.	Post- fach	GE
Mögliche Werte:	0, 1	0-3, 6	0-3, 6	0-3, 6	0-3	-, 0, 1	-, 0	-, 0

Bei diesen Werten sollten Sie sich die vorgeschlagenen Änderungen vom Kunden bestätigen lassen.

- Adresse ist nicht in Ordnung (keine Vorschläge). Das kann bei folgenden Werten der Fall sein:

Aufrufparameter	Land	PLZ	Ort	Orts- teil	Straße	Haus- nr.	Post- fach	GE
Mögliche Werte:	2-9, A-G	4, 5, 7, 8, 9, A-G	4, 5, 7, 8, 9, A-G	4, 5, 7, 8, 9, A-G	4 bis 9, A-G	2 bis 9, A-G	1 bis 9, A-G	1 bis 9, A-G

Bei diesen Werten sollten Sie den Kunden konkret auf fehlende oder falsche Angaben hinweisen und um Klärung bitten.



Wenn ein Straßename oder eine Hausnummer nicht im Datenbestand vorhanden ist, was z. Zt. bei einigen Orten in der Schweiz und in Österreich der Fall ist, wird der Wert „nicht analysiert“ zurückgeliefert. In diesem Fall können die Angaben übernommen werden, da eine erneute Überprüfung der Angaben durch den Kunden mit anschließender Analyse das gleiche Ergebnis liefern würde.



Außer dem Flag **OutAddressOK** sollten Sie immer auch den Protokollstring auswerten. Nur so können Sie aus der Anwendung eine Fallunterscheidung für stark veränderte Adressen vornehmen.

9.4.3 Kurzbeschreibung der RMS-Typen

Im Modus GUI können ausschließlich die RMS-Typen genutzt werden, die einen Scorewert zurückliefern. Das sind die Typen Basisscore, Boniscore, Bonikombi und Basiskombi.

RMS-Typ	Prüfmethoden	Ergebnis
1) Basisscore	Scoring mit generischen Antragsdaten, anonymisierten soziodemografischen Daten und Verhaltensdaten*	Scorewert
2) Boniscore	Scoring mit generischen Antragsdaten, anonymisierten soziodemografischen Daten und Verhaltensdaten*; Prüfung auf Negativmerkmale	Scorewert, Ampelwert

RMS-Typ	Prüfmethoden	Ergebnis
6) Bonikombi	Scoring mit generischen Antragsdaten, anonymisierten soziodemografischen Daten und Verhaltensdaten*; Prüfung auf Negativmerkmale; Adressverifikation	Scorewert, Ampelwert, Konsenswert
7) Basiskombi	Scoring mit generischen Antragsdaten, anonymisierten soziodemografischen Daten und Verhaltensdaten*; Adressverifikation	Scorewert, Konsenswert
* Verhaltensdaten nur bei Beauftragung des Service Level Receivables oder Accounting der Pago Financial Services		

Nachdem Sie über die von Ihnen gewünschten RMS-Typen einen Vertrag mit Pago geschlossen haben, konfiguriert Pago die RMS-Typen für Sie. Für die Typen Pago Bonicheck, Pago Boniscore und Pago Namecheck wird ein zusätzlicher Vertrag mit den Datenlieferanten geschlossen (siehe Leistungsbeschreibung Pago Risk Management Services, Release 1.0).

9.4.4 Ein- und Rückgabewerte der RMS-Typen

9.4.4.1 Typ 1: Pago Basisscore

Eingabewerte: Adresse, Geschlecht, Geburtsdatum, Kaufbetrag, Kunden-ID (für die Zuordnung der Debitoren-Historie)

Rückgabewert: Scorewert (numerisch)

9.4.4.2 Typ 2: Pago Boniscore

Eingabewerte: Name, Vorname, Adresse, Geburtsdatum, Kaufbetrag

Rückgabewerte: • Scorewert (numerisch)
 • Ampelwert (alphanumerisch: „r“, „y“ oder „g“)

9.4.4.3 Typ 6: Pago Bonikombi (Pago Boniscore und Pago Namecheck)

Eingabewerte: Name, Vorname, Adresse, Geschlecht, Geburtsdatum, Kaufbetrag

Rückgabewerte: • Ampelwert (alphanumerisch): „r“, „y“ oder „g“
 • Scorewert (numerisch)
 • Konsenswert Nachname (= Wert der Übereinstimmung des Nachnamens (Haushalt) mit der Adresse; numerisch): „1“, „2“ oder „3“
 • Konsenswert Vorname (= Wert der Übereinstimmung des Vornamens mit der Adresse; numerisch): „1“, „2“ oder „3“



Der Konsenswert für den Vornamen wird nur geprüft, wenn der Konsenswert für den Nachnamen den Wert 1 oder 2 ergeben hat.

9.4.4.4 Typ 7: Pago Basiskombi (Pago Basisscore und Pago Namecheck)

Eingabewerte: Name, Vorname, Adresse, Geschlecht, Geburtsdatum, Kaufbetrag

Rückgabewerte:

- Scorewert (numerisch)
- Konsenswert Nachname (= Wert der Übereinstimmung des Nachnamens (Haushalt) mit der Adresse; numerisch): „1“, „2“ oder „3“
- Konsenswert Vorname (= Wert der Übereinstimmung des Vornamens mit der Adresse; numerisch): „1“, „2“ oder „3“



Der Konsenswert für den Vornamen wird nur geprüft, wenn der Konsenswert für den Nachnamen den Wert 1 oder 2 ergeben hat.

9.4.5 Verwendung des Scorewertes

Je höher der zurückgelieferte Scorewert, umso geringer ist das Ausfallrisiko hinsichtlich der Zahlung. Deshalb können Sie Kunden mit hohen Scorewerten mehr Zahlungsarten anbieten als Kunden mit niedrigen Scorewerten.

Als Richtlinie für die Zuordnung von Zahlarten zu Scorewerten kann die folgende Aufstellung dienen. Sie beruht auf Erfahrungswerten im Konsumgüterbereich und wird regelmäßig überprüft und aktualisiert.

<i>Scorewert:</i>	<i>Zahlarten:</i>
Unter 890:	Stackbox, Vorkasse
mindestens 890:	Stackbox, Vorkasse, Nachnahme
mindestens 940:	Stackbox, Vorkasse, Nachnahme, Kreditkarteneinzug, Paybox
mindestens 970:	Stackbox, Vorkasse, Nachnahme, Kreditkarteneinzug, Paybox, Lastschrifteinzug
mindestens 1000:	Stackbox, Vorkasse, Nachnahme, Kreditkarteneinzug, Paybox, Lastschrifteinzug, Rechnung

Bei risikobehafteten Warengruppen empfehlen wir, sich für eine vorsichtigeren Zuordnung von Scorewerten zu Zahlarten zu entscheiden. Entsprechend toleranter können Sie bei Warengruppen mit einem geringerem Risikopotential verfahren.

9.5 Beispiele für den Einsatz von Multipay mit GUI

In den folgenden Abschnitten stellen wir Ihnen die elementaren Anwendungsfälle für Multipay mit GUI beispielhaft vor. Die Beispiele veranschaulichen Ihnen Ihre Möglichkeiten und bieten außerdem eine Diskussionsgrundlage für die Unterstützung durch Pago Consulting.

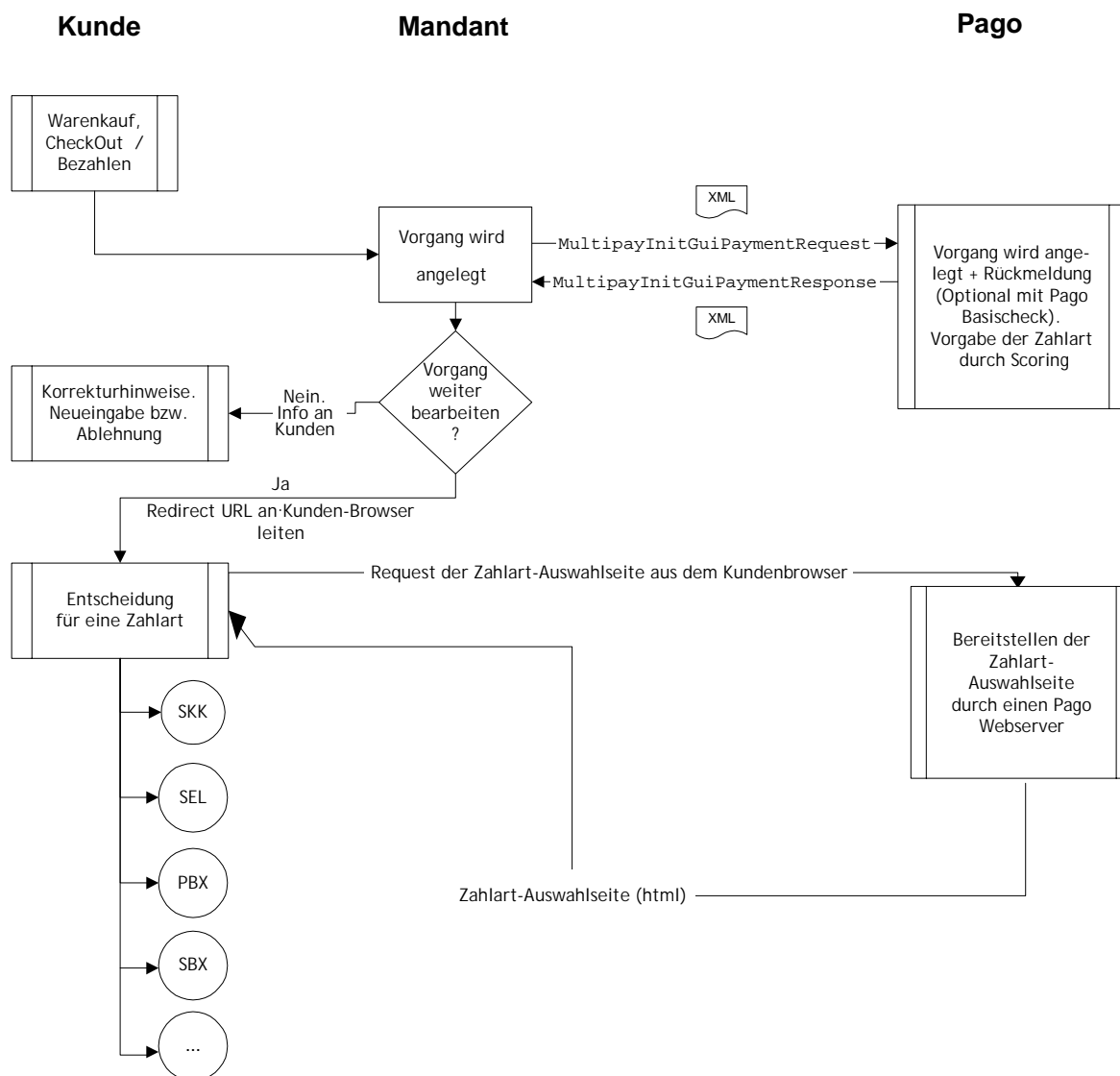
9.5.1 GUI-Payment mit Scoring und Zahlarten-Auswahl

Der Kunde beendet seinen Kaufprozess mit dem Gang an die virtuelle Kasse (Check Out). Der Zahlvorgang wird durch **MultipayInitGuiPaymentRequest** angelegt. Der Kunde wird auf den Webserver mit der Zahlartauswahlseite umgeleitet. Anschließend wird die Zahlung durch Interaktion zwischen Kunden und Pago autorisiert.

Mit der Initialisierung wird, je nach Einstellung, ein Scoring-Prozess und evtl. vorher eine Adressprüfung ausgeführt. Das Ergebnis des Scoring wird für die Zahlartenauswahl verwendet.

Auf der Zahlarten-Auswahlseite entscheidet sich der Kunde für seine gewünschte Zahlart.

In der folgenden Grafik wird der gesamte Ablauf beispielhaft dargestellt:

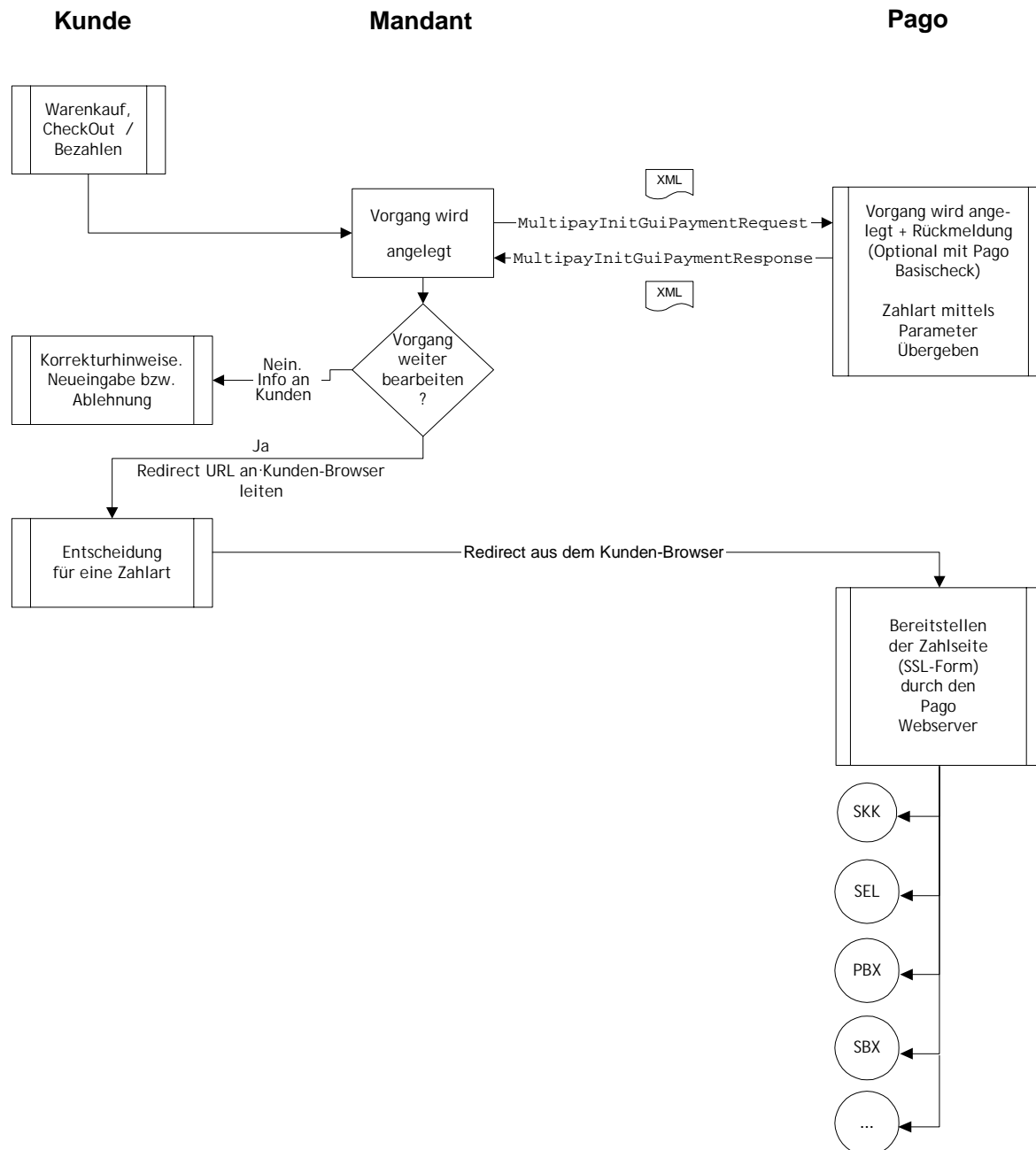


Zum besseren Verständnis des Initialisierungsprozesses siehe auch die Grafik in Abschnitt 9.4 „Pago Risk Management Services in Multipay GUI“.

9.5.2 GUI-Payment ohne Zahlarten-Auswahl

Alternativ zur Zahlartenauswahl mit Hilfe von Pago RMS kann mit der Initialisierung im Anschluss an den Kaufprozess beim Checkout (Gang an die Kasse) auch die Zahlart direkt mit **MultipayInitGuiPaymentRequest** übergeben werden. Die Autorisierung erfolgt im nächsten Schritt durch Interaktion zwischen Kunde und Pago. Dabei erhält der Kunde keine Zahlarten-Auswahlseite.

In der folgenden Grafik wird ein möglicher Ablauf dargestellt (Initialisierung des Vorgangs ohne Zahlarten-Auswahlseite und mit SSL-Form, Autorisierung und Einzug des Betrags).



9.5.3 GUI-Payment mit oder ohne Zahlarten-Auswahl, mit SSL-Form für Zahlung mit Kreditkarte

Die vom Kunden eingegebenen Zahlungsinformationen werden vor der Durchführung auf Plausibilität überprüft. Über das Payment-System von Pago werden dabei folgende Angaben bei Kreditkartenbezahlung überprüft:

- die Kreditkarten-Anfangsziffern der unterstützten Kreditkartengesellschaften
- die Prüzfiffer der Kreditkartennummer
- das Verfalldatum der Kreditkarte.

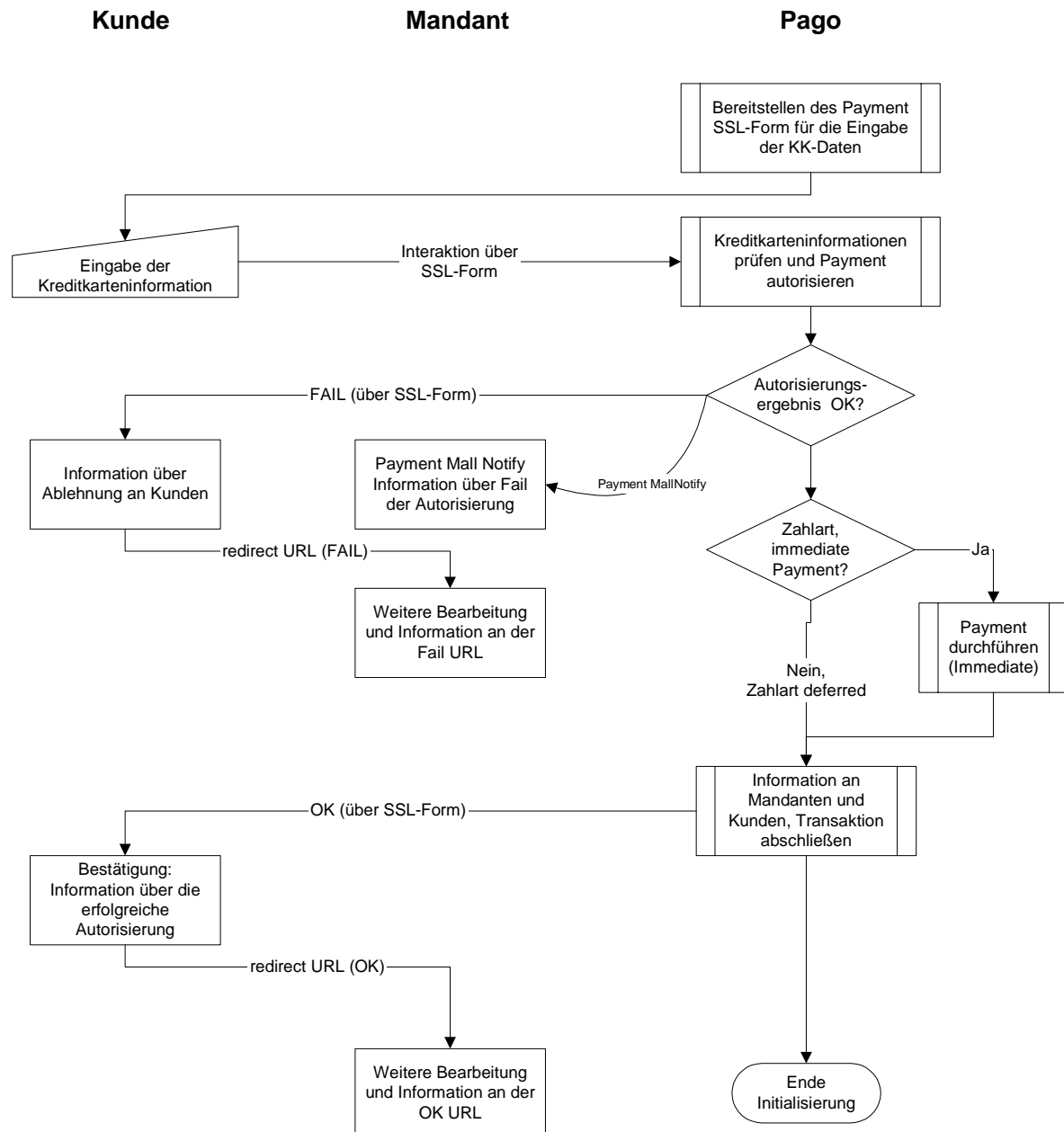
Die Prüfroutinen bieten eine einfache Kontrolle gegen absichtliche oder unbeabsichtigte Fehleingabe des Kunden.

Wenn alle Angaben logisch korrekt sind, wird die direkte Belastung der Kreditkarte an das zuständige Kreditkarteninstitut in Auftrag gegeben, bzw. die elektronische Lastschrift an die Bank weitergegeben.

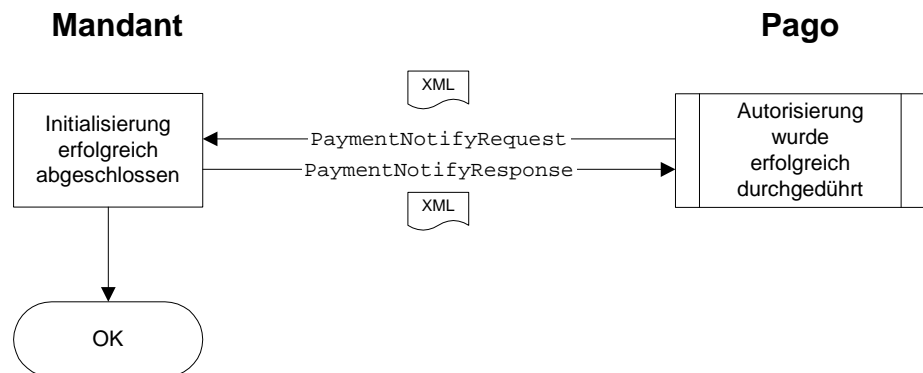
Sollte die Kreditkartenbelastung durch das Kreditkarteninstitut nicht durchgeführt werden können, wird eine entsprechende Fehlermeldung an den Kunden und das Mandantensystem zurückgemeldet.

Optional kann vor der Durchführung der Zahlungsanweisung eine Adressprüfung und eine Risikobewertung über die Pago Risk Management Services durchgeführt werden.

Bei der Entscheidung für Kreditkarte wird das für die Kreditkartenzahlung benötigte SSL-Form vom Kunden-Browser angefordert. Der detaillierte Prozess stellt sich wie folgt dar:



Zeitgleich mit dem Redirect des Kunden auf den Webserver des Mandanten wird eine XML-Benachrichtigung generiert und in Form des **PaymentNotifyRequest** an das Mandantensystem gesendet.



In dem Beispiel wird von einer erfolgreichen Autorisierung ausgegangen.

9.5.4 GUI-Payment mit oder ohne Zahlarten-Auswahl, mit SSL-Form für Zahlung mit Kreditkarte und Teilnahme am 3-D-Secure-Verfahren

Beim 3-D-Secure Verfahren, das von VISA entwickelt wurde, wird vor der eigentlichen Betragsreservierung oder Betragsbuchung eine zusätzliche Authentifizierung des Karteninhabers durchgeführt. Ist der Händler bei VISA zur Teilnahme an diesem Verfahren angemeldet, werden alle Zahlungen mit VISA-Kreditkarten zusätzlich für den Händler abgesichert. Nähere Informationen zur erhöhten Absicherung für den Händler werden in der Dokumentation zu 3-D Secure zur Verfügung gestellt.

Für alle VISA-Transaktionen wird nun geprüft, ob die Kreditkarte ebenfalls für das Verfahren angemeldet ist. Für angemeldete Karten wird nun die Eingabe des durch den Karteninhaber konfigurierten Passworts benötigt.

Wie bei jeder Zahlung mit Kreditkarte überprüft Pago auch bei 3-D-Secure vor der eigentlichen Authentifizierung oder Autorisierung über das Payment-System folgende Angaben:

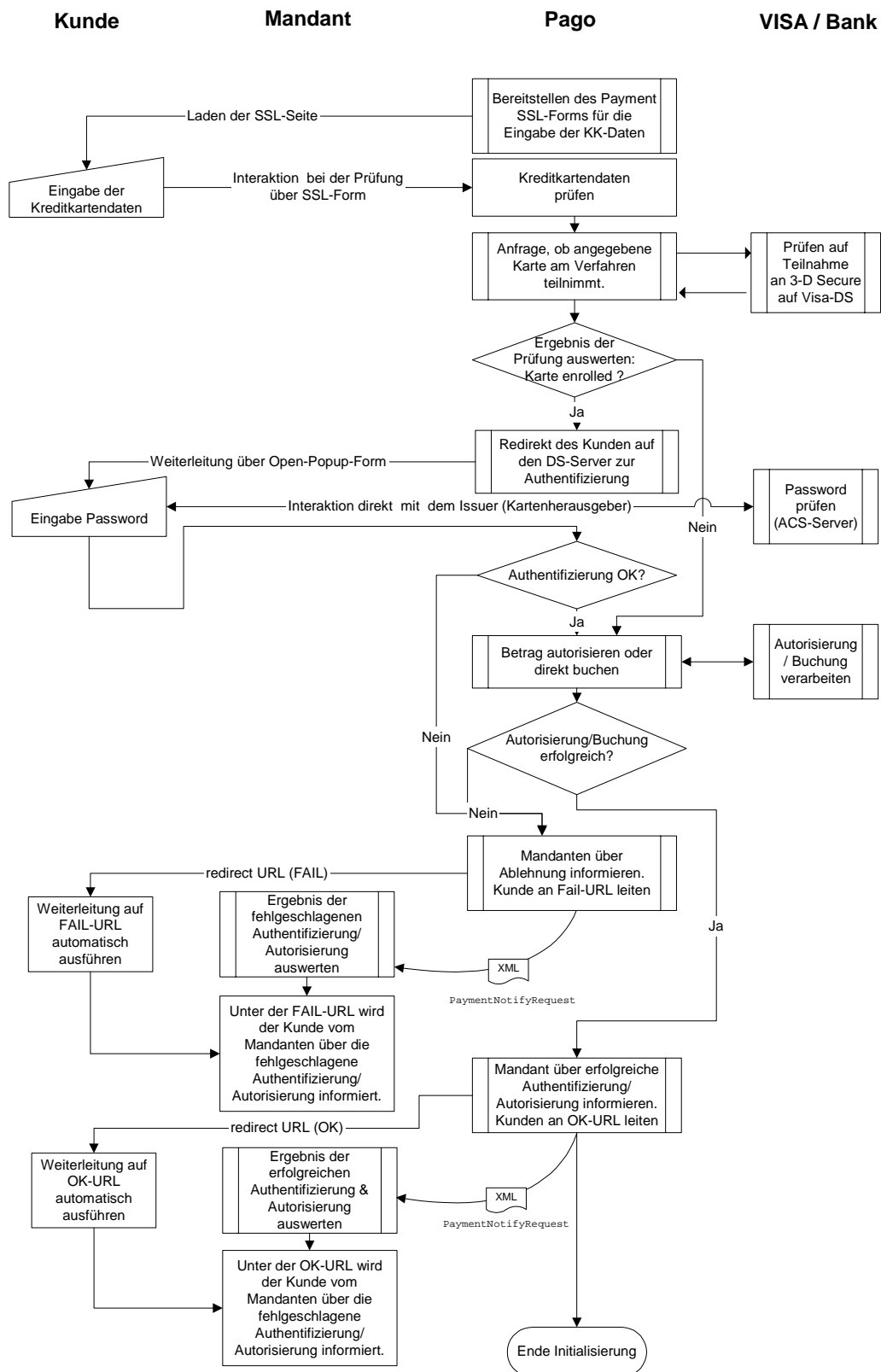
- die Kreditkarten-Anfangsziffern der unterstützten Kreditkartengesellschaften
- die Prüfziffer der Kreditkartennummer
- das Verfalldatum der Kreditkarte.

Die Prüfroutinen bieten eine einfache, zusätzliche Kontrolle gegen absichtliche oder unbeabsichtigte Fehleingabe des Kunden.

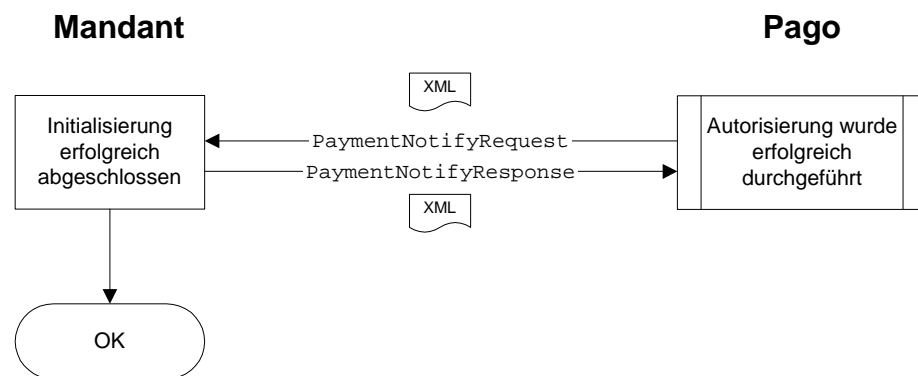
Wenn alle Angaben logisch korrekt sind, wird die direkte Belastung der Kreditkarte an das zuständige Kreditkarteninstitut in Auftrag gegeben. Sollte die Kreditkartenbelastung durch das Kreditkarteninstitut nicht durchgeführt werden können, wird eine entsprechende Fehlermeldung an den Kunden und an das Mandantensystem zurückgemeldet.

Optional kann auch für 3-D-Secure vor der Durchführung der Zahlungsanweisung eine Adressprüfung und eine Risikobewertung über die Pago Risk Management Services durchgeführt werden.

Bei der Entscheidung für Kreditkarte wird das für die Kreditkartenzahlung benötigte SSL-Form vom Kunden-Browser angefordert. Der detaillierte Prozess stellt sich wie folgt dar:



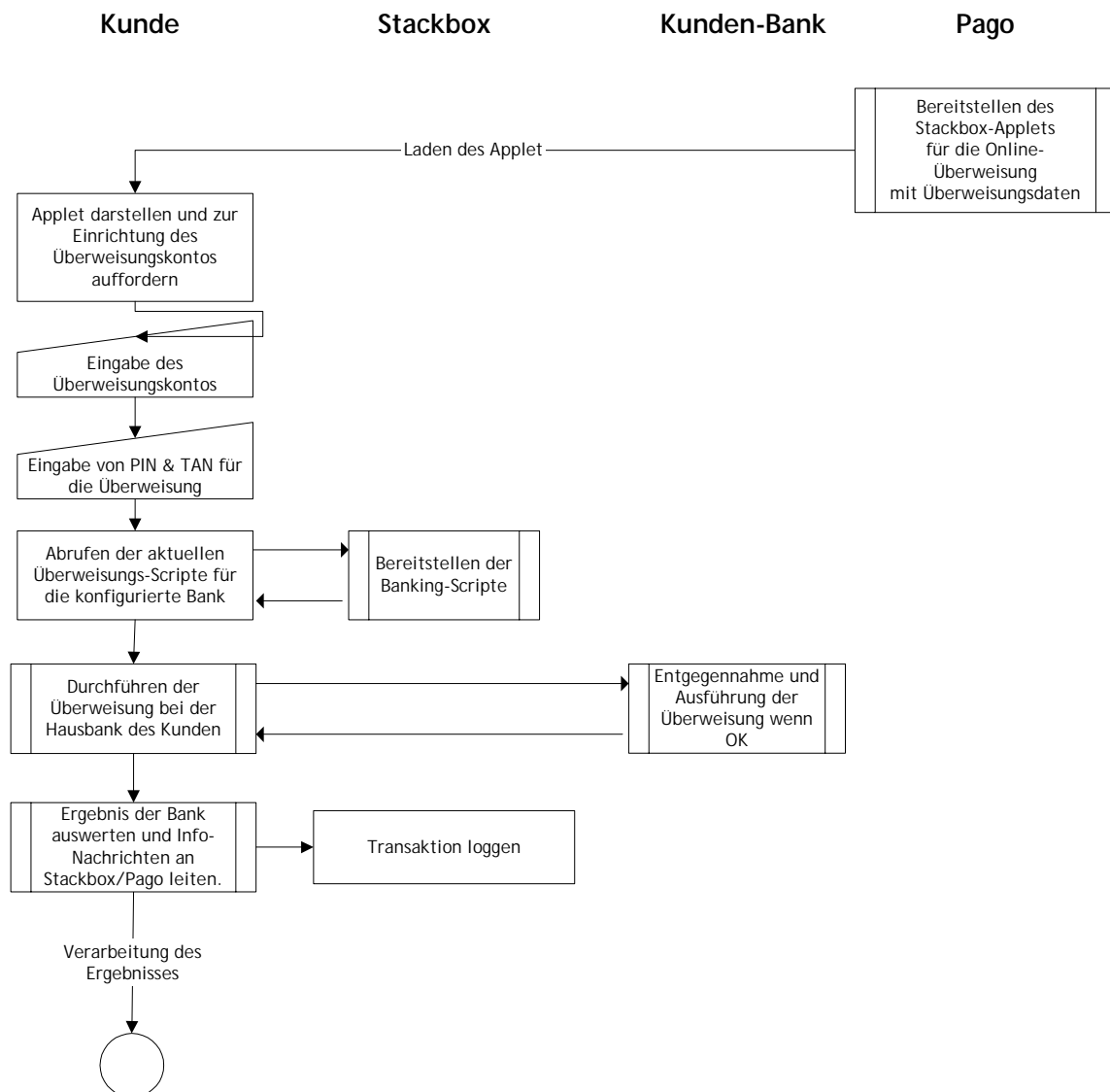
Zeitgleich mit dem Redirect des Kunden auf den Webserver des Mandanten wird eine XML-Benachrichtigung generiert und in Form des **PaymentNotifyRequest** an das Mandantensystem gesendet.



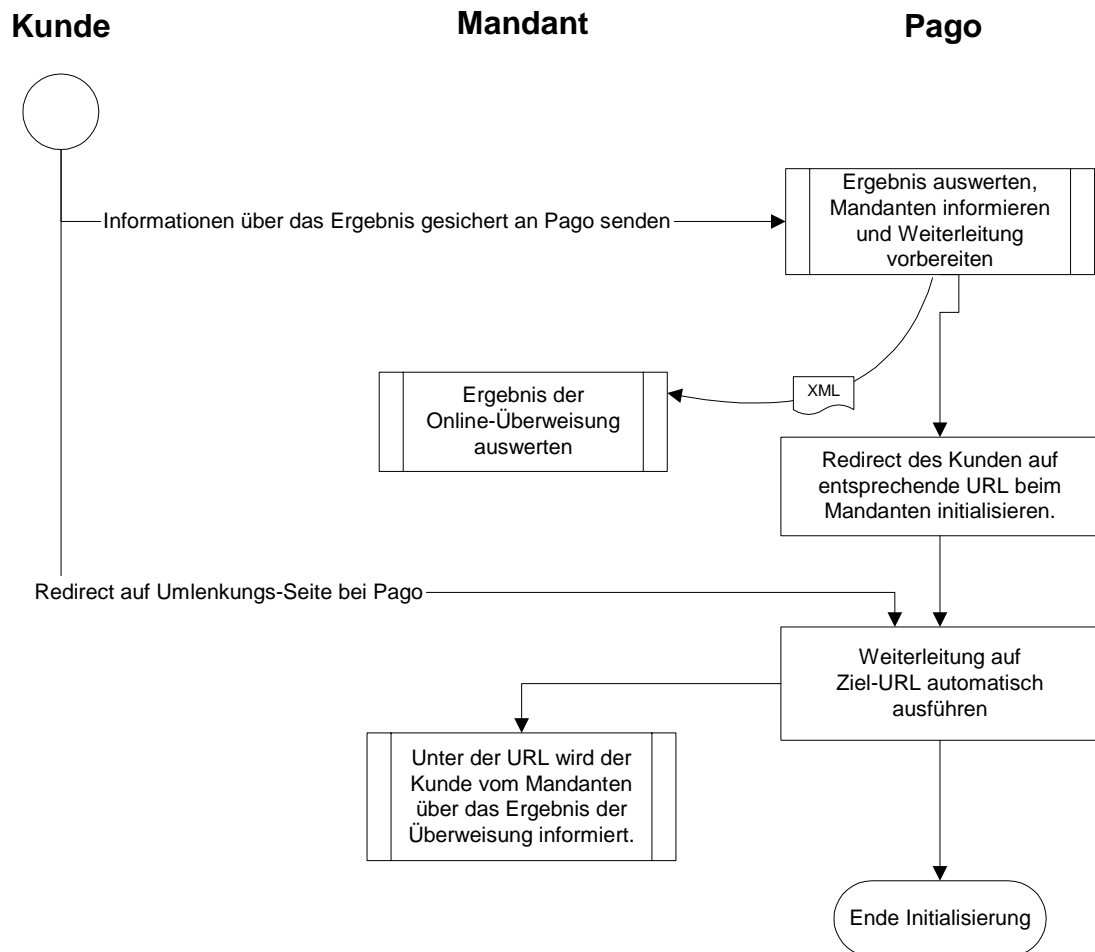
In dem Beispiel wird von einer erfolgreichen Autorisierung ausgegangen.

9.5.5 GUI-Payment mit oder ohne Zahlarten-Auswahl, Online-Überweisung mit Stackbox-Applet

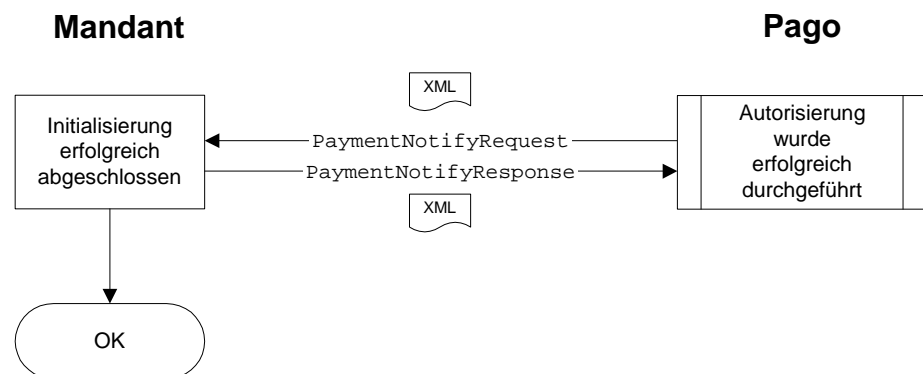
Unter Verwendung eines Java-Applet von der Firma Stackbox kann nun die Online-Überweisung als Zahlart verwendet werden. Die Initialisierung des Applet erfolgt auf der Basis der konfigurierten Stammdaten (Name und Kontoverbindung des Empfängers) und der Informationen aus dem Vorgang (Preis, Referenznummer). Die Information und Einrichtung der Bankverbindung des Kunden wird nach dem Download des Applet vom Kunden vorgenommen. Mit Eingabe von PIN und TAN kann dann die Überweisung durchgeführt werden.



Zeitgleich mit dem Redirect des Kunden auf den Webserver des Mandanten wird eine XML-Benachrichtigung generiert und in Form des **PaymentNotifyRequest** an das Mandantensystem gesendet.



Nach Veränderung des Payment-Status wird dieser von Pago über **PaymentNotifyRequest** übermittelt.



In dem Beispiel wird von einer erfolgreichen Autorisierung ausgegangen.

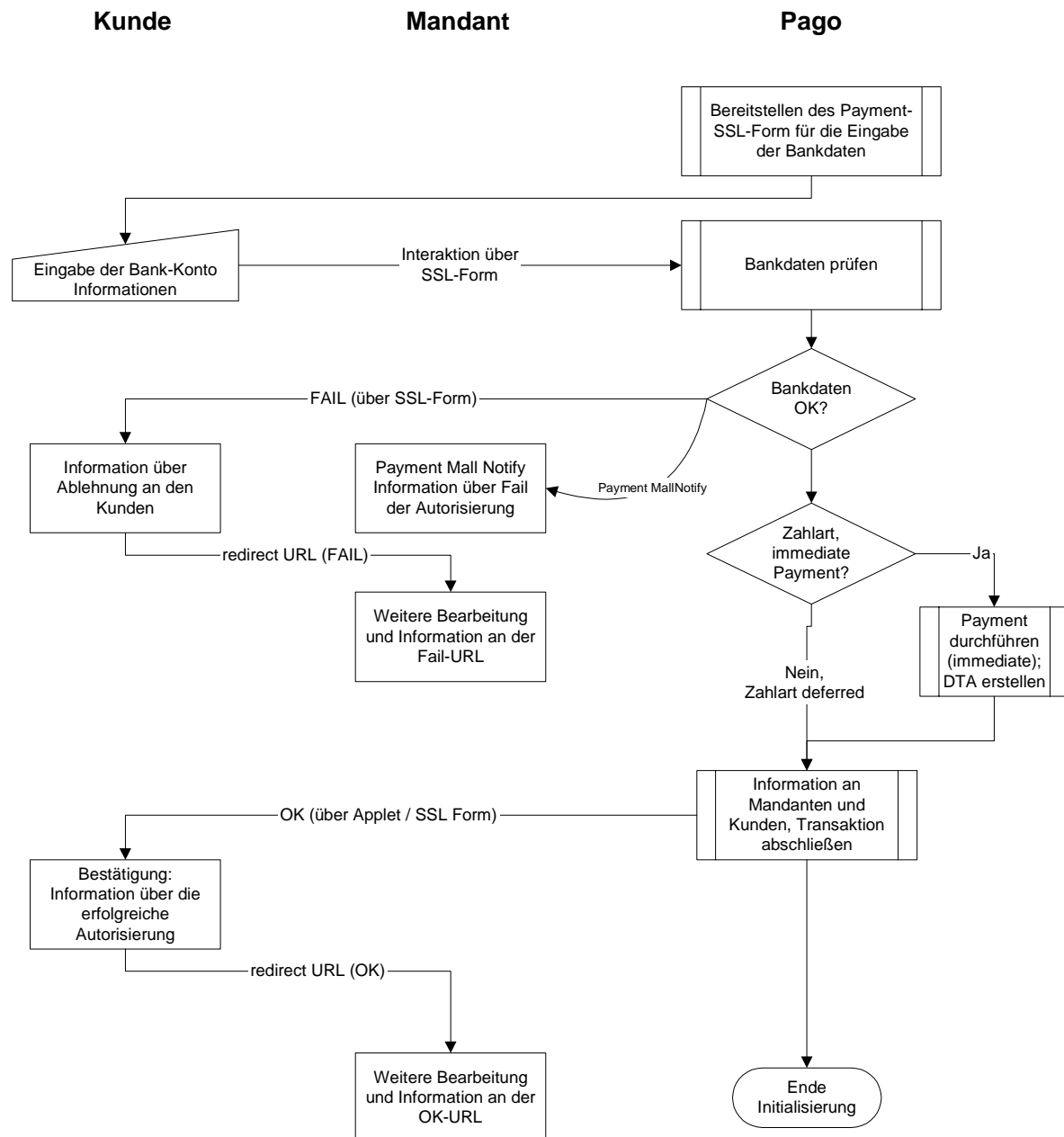
9.5.6 GUI-Payment mit oder ohne Zahlarten-Auswahl, mit SSL-Form für Zahlung per Lastschrift (ELV)

Die vom Kunden eingegebenen Zahlungsinformationen werden vor der Durchführung auf Plausibilität überprüft. Über das Payment-System von Pago werden dabei folgende Angaben bei Lastschriftbezahlung überprüft:

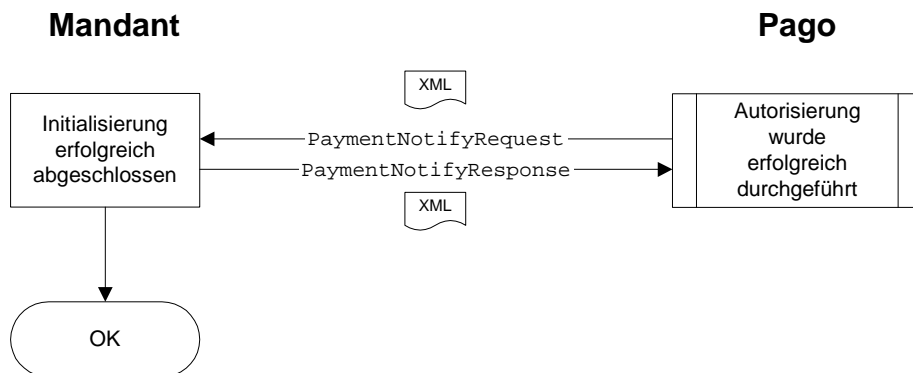
- Bankleitzahl mit der angegebenen Kontonummer anhand des entsprechenden Prüfziffernverfahrens

Die Prüfroutinen bieten eine einfache Kontrolle gegen absichtliche oder unbeabsichtigte Fehleingabe des Kunden.

Bei der Entscheidung für Lastschrift wird das für Lastschrift benötigte SSL-Form vom Kunden-Browser angefordert. Der detaillierte Prozess stellt sich wie folgt dar:



Zeitgleich mit dem Redirect des Kunden auf den Webserver des Mandanten wird eine XML-Benachrichtigung generiert und in Form des **PaymentNotifyRequest** an das Mandantensystem gesendet.

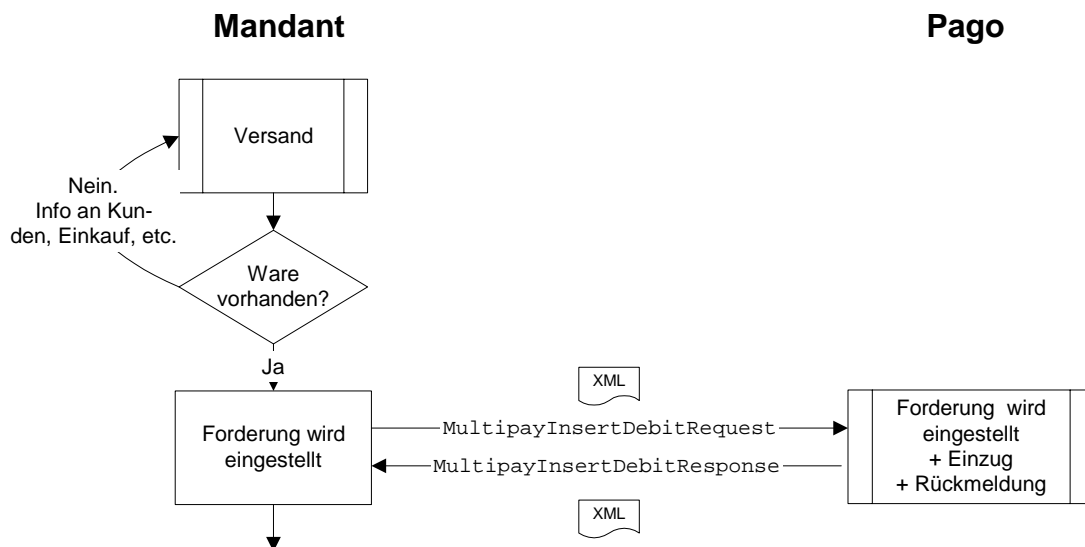


In dem Beispiel wird von einer erfolgreichen Autorisierung ausgegangen.

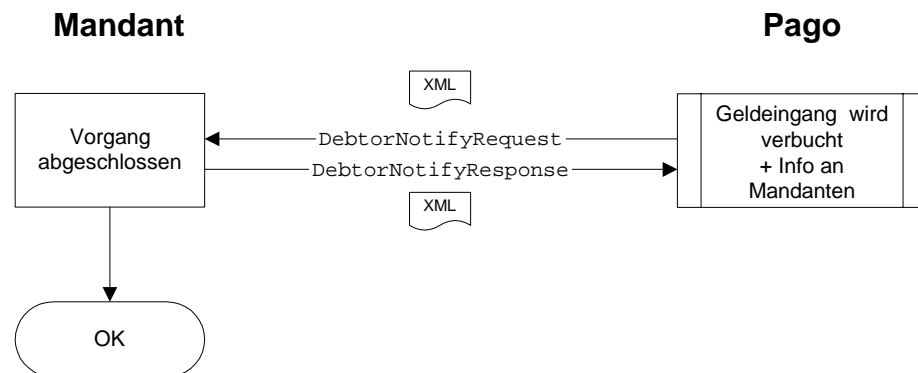
9.5.7 GUI-Payment mit oder ohne Zahlarten-Auswahl, Einstellen einer Forderung, alles einziehen

Nachdem sichergestellt ist, dass die Ware am Lager ist, wird eine Forderung eingestellt. Die Ware wird versendet und der Forderungsbetrag eingelöst.

(Der Parameter **CollectNow** = **true** bedeutet, dass der Betrag eingezogen wird.)



Der Betrag geht bei Pago ein, und die Forderung wird ausgeglichen. Der Mandant (z. B. Online Shop) wird über den veränderten Debitorenstatus mittels **DebtorNotifyRequest**, den Pago versendet, informiert.



Weitere Informationen zur asynchronen Rückmeldung entnehmen Sie bitte der Dokumentation zur asynchronen Rückmeldung.

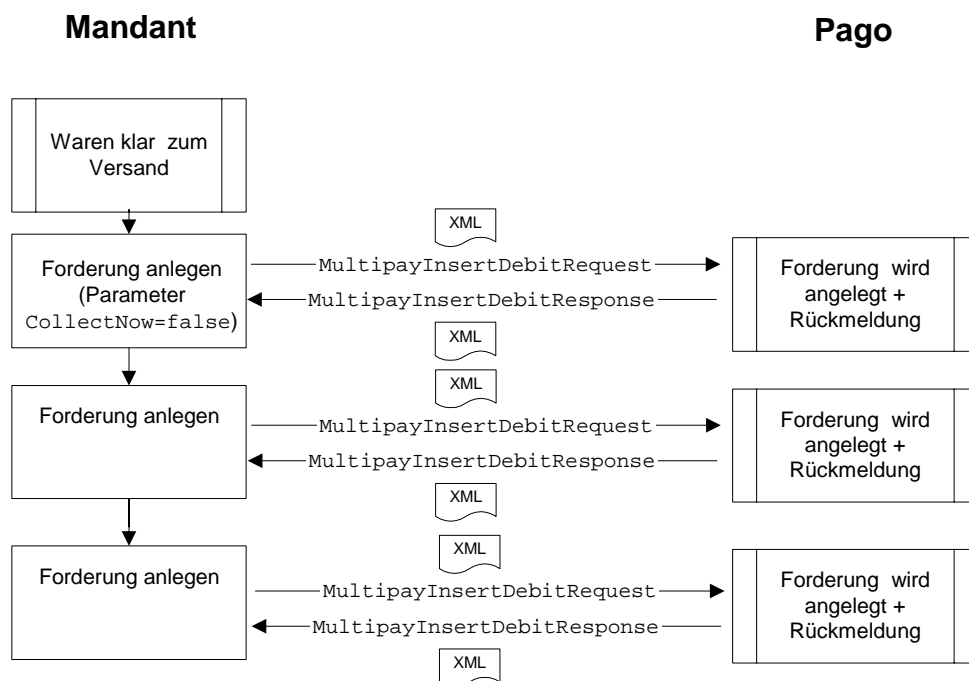
9.5.8 GUI-Payment mit oder ohne Zahlarten-Auswahl, Einstellen mehrerer Forderungen bei mehreren Positionen, späterer Einzug

Wenn mehrere Artikel versendet werden können, müssen Sie für jeden Artikel eine Forderung einstellen.

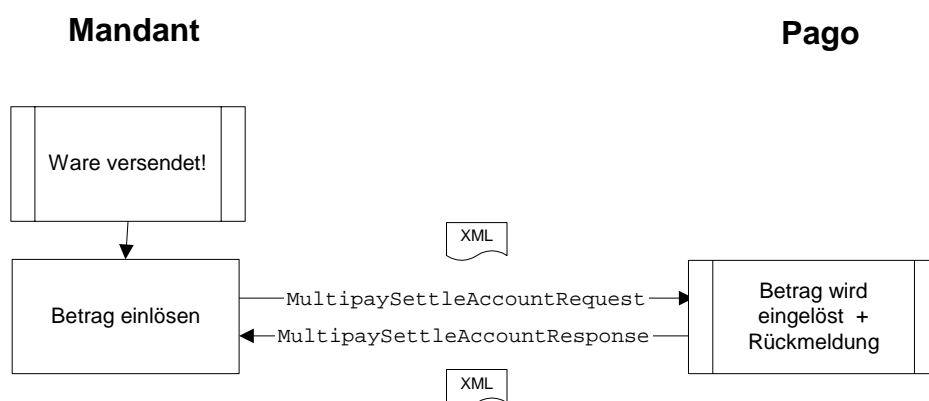
Durch Warenwert, Versandkosten, MwSt, etc. ergeben sich mehrere Forderungen, die über einzelne **MultipayInsertDebitRequest** angelegt werden. Auch Forderungen mit unterschiedlichen MwSt-Sätzen können so eingestellt und in den Reports aus dem Debitorenmanagement einschließlich der MwSt-Sätze dargestellt werden.

Ein anderer Grund für das Einstellen einzelner Positionen ist die unterschiedliche Verfügbarkeit der Waren in einem oder mehreren Lagern.

In diesem Beispiel wird gezeigt, wie mit mehreren Forderungssätzen für Warenwert, MwSt und Versandkosten gearbeitet werden kann.



Der Saldo der offenen Forderungen wird nun eingelöst. Der Einzug wird über die Funktion **MultipaySettleAccountRequest** ausgelöst, da vorher die Forderungen mit dem Parameter **CollectNow = false** eingestellt wurden. Durch die Funktion **MultipaySettleAccountRequest** werden die Forderungen für einen Einzug zusammengefasst. Alternativ hätte auch mit der letzten Forderung der Parameter **CollectNow = true** eingestellt werden können.

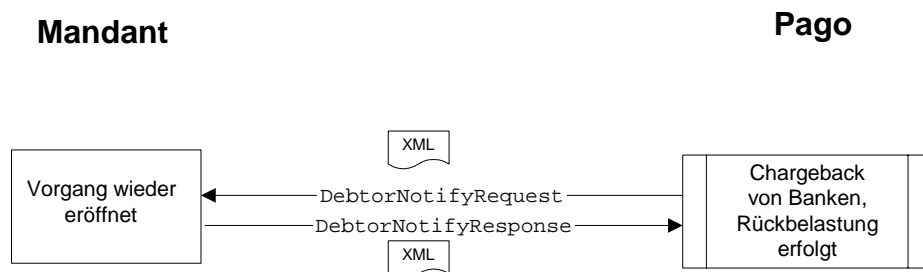


Der Debitorenstatus gibt Auskunft über den Geldeingang.

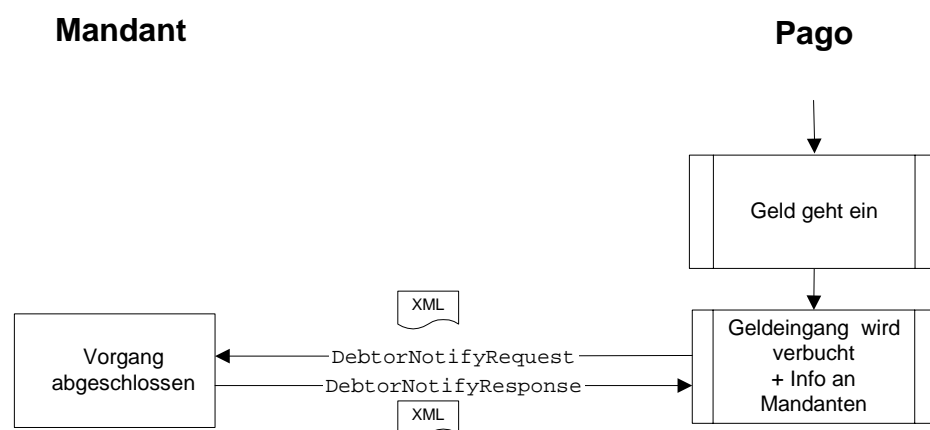


9.5.9 GUI-Payment mit oder ohne Zahlarten-Auswahl, Zahlungsrückläufe und Gutschriften; Meldung nach Chargeback

Statusinformationen aus dem Debitorenmanagement können auch nach dem Abschluss eines Vorgangs auftreten. In diesem Beispiel nehmen wir ein Chargeback auf eine Kreditkartenzahlung oder eine Rücklastschrift auf eine Einzugsermächtigung an. Der Debitorenstatus wird mittels **DebtorNotifyRequest** von Pago bereitgestellt.



Das Mahnwesen wird in diesem Fall über Pago abgewickelt. Der Kunde reagiert auf die Mahnung und zahlt. Der Geldeingang wird über den Debitorenstatus mitgeteilt.

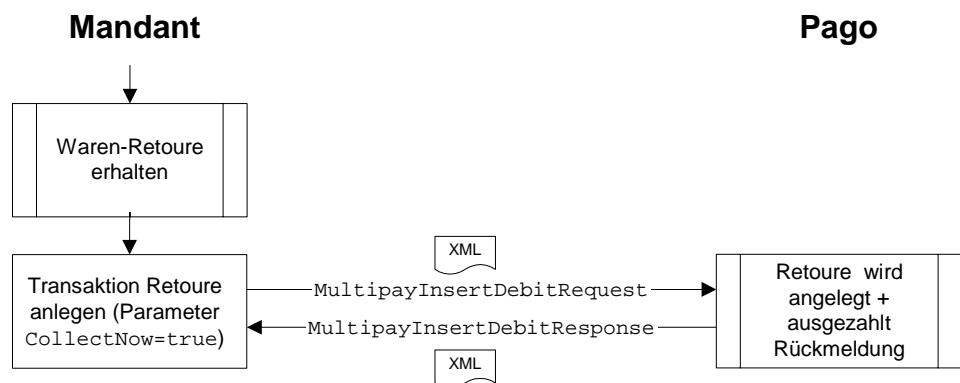


9.5.10 GUI-Payment mit oder ohne Zahlarten-Auswahl, Zahlungsrückläufe und Gutschriften, Gutschrift

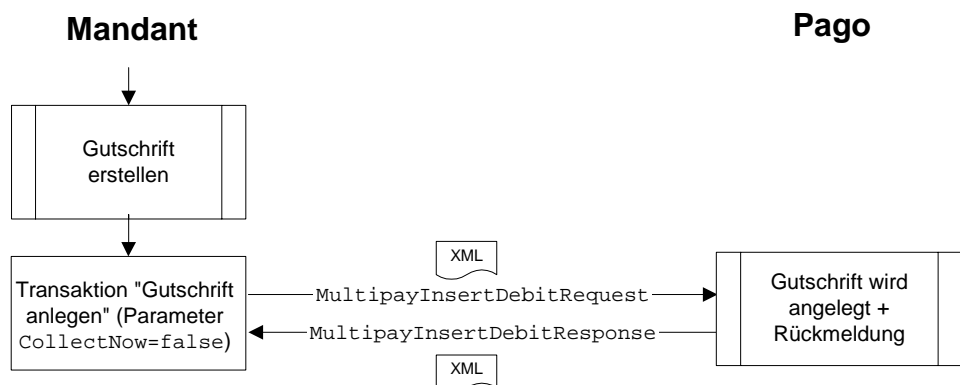
Gutschriften werden mit **MultipayInsertDebitRequest** eingestellt. (Achten Sie hierbei auf die Regeln aus Abschnitt 9.1 „Anwendungsregeln für den Einsatz von Multipay im Modus GUI“).

Dieses Beispiel zeigt den möglichen Ablauf für die Verarbeitung einer Retoure und einer Gutschrift.

Die Retoure wird mit sofortiger Auszahlung angelegt.

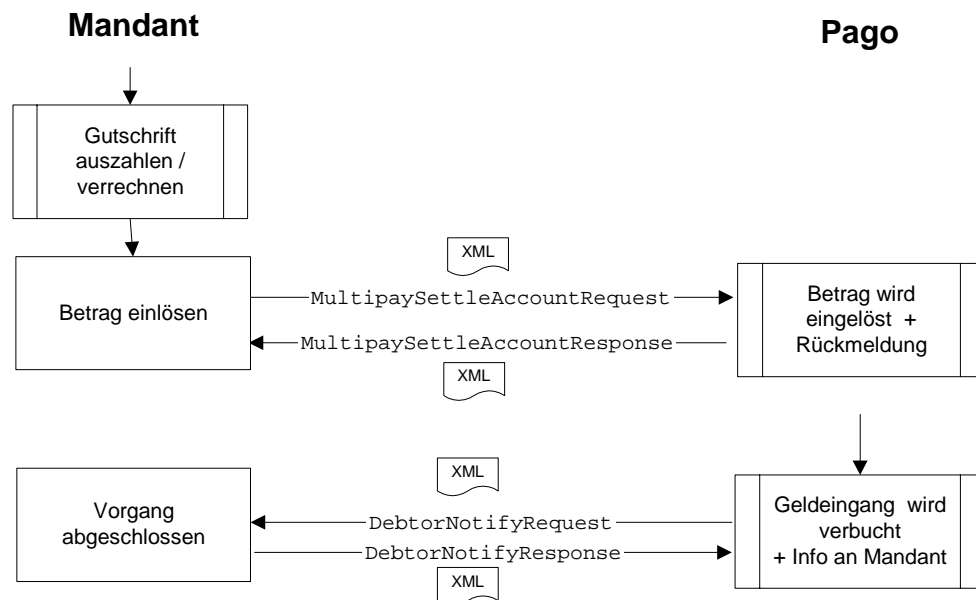


Wenn eine Gutschrift nicht ausgezahlt, sondern mit Forderungen verrechnet werden soll, verwenden Sie den Parameter **CollectNow = false**. Die Verrechnung erfolgt mit dem nächsten **MultipaySettleAccountRequest** oder dem nächsten **MultipayInsertDebitRequest**, wenn der Parameter **CollectNow = true** gesetzt ist.



Nach dem Aufruf der Funktion **MultipaySettleAccountRequest** wird die Gutschrift ausgeführt. Wenn in der Zwischenzeit eine neue Forderung eingestellt wird, wird diese mit der Gutschrift verrechnet und nur noch der Saldo ausgezahlt.

! Durch eine Gutschrift wird eine Forderung nicht aufgehoben. Die Summe der Forderungen muss weiterhin kleiner oder gleich dem Betrag des **MultipayInitPaymentRequest** sein.



9.6 Einbindung des SSL-Form bzw. Applet in das Mandantensystem

Der Zahlungsprozess wird, je nach Zahlart, Einrichtung und Initialisierung - wie oben beschrieben - über verschiedene modulare Webseiten abgebildet. Die Kommunikation erfolgt zwischen Mandant und Pago-System. Damit lädt der Mandant die Webseiten zur Zahlung direkt vom Pago-Webserver.

Für jede Zahlart wird eine Template-Seite benötigt, in die das SSL-Form oder das Applet eingebunden wird. Vor Inbetriebnahme von Pago Multipay im Modus GUI sind die jeweiligen Template-Seiten für die Einbindung der SSL-Forms an das Layout Ihres Systems anzupassen.

Für die Zahlarten Kreditkarte, Elektronisches Lastschriftverfahren und Paybox wird ein SSL-Form eingebunden. Das Template muss an entsprechender Stelle folgenden Tag enthalten, der bei der dynamischen Generierung durch das SSL-Form ersetzt wird:

```
<! SSLFORM>
```

Für die Zahlarten Kreditkarte, Elektronisches Lastschriftverfahren und Paybox werden unterschiedliche SSL-Forms verwendet, die folgende Größe aufweisen:

Kreditkarte: width=500, height=255

Elektronisches Lastschriftverfahren: width=500, height=195

Paybox: width=500, height=205

Für die Zahlart Stackbox wird ein Java-Applet eingebunden. Das Template muss an entsprechender Stelle folgenden Tag enthalten, der bei der dynamischen Generierung durch das Java-Applet ersetzt wird:

```
</APPLET>
```

Das Java-Applet hat folgende Größe: width=600, height=400

Die Zahlseite setzt sich durch das Layout des Mandanten und die entsprechenden SSL-Forms zusammen. Die einzelnen Forms sind im Layout neutral gehalten:

Template:

SSL-Forms / Applet:



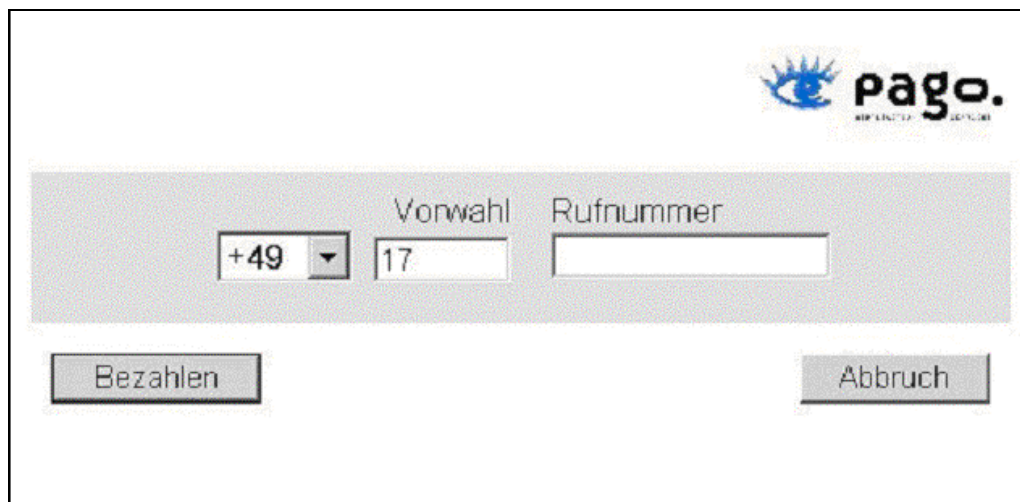
Das SSL-Form für die Eingabe von Kreditkartendaten enthält Felder für die Kreditkartennummer, den CVV bzw. CVC und das Verfalldatum. Das Verfalldatum wurde auf zwei Felder aufgeteilt, um versehentliche Fehleingaben zu reduzieren.

The screenshot shows a web form for credit card payment. At the top right is the 'pago.' logo with a blue eye icon. Below the logo, there are four input fields arranged in a 2x2 grid. The top row contains 'Kreditkartennummer:' followed by a long text box, and 'Prüfnummer:' followed by a short text box. The bottom row contains 'Gültig bis (MM/JJJJ):' followed by two short text boxes separated by a slash, with the second box containing the number '20'. To the right of these fields is a link that says 'Wo finde ich die Prüfnummer meiner Kreditkarte?'. At the bottom of the form are two buttons: 'Bezahlen' on the left and 'Abbruch' on the right.

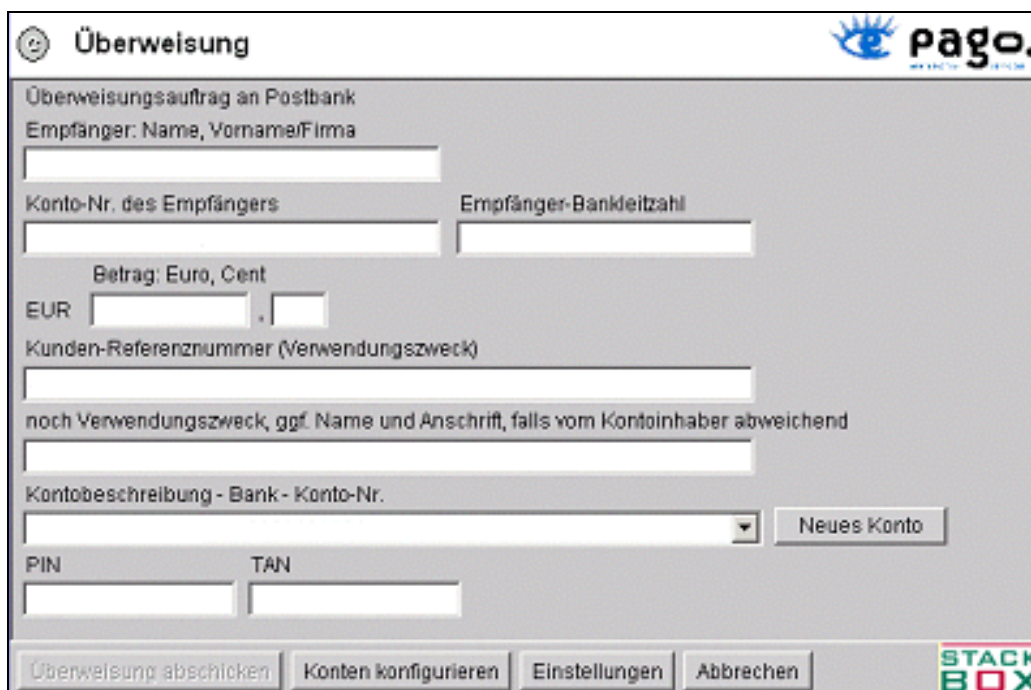
Das SSL-Form für die Eingabe der Bankverbindung zum Lastschriftinzug enthält die Felder „Kontonummer“ und „Bankleitzahl“.

The screenshot shows a web form for bank transfer payment. At the top right is the 'pago.' logo with a blue eye icon. Below the logo, there are two input fields: 'Kontonummer:' followed by a long text box, and 'Bankleitzahl:' followed by a short text box. At the bottom of the form are two buttons: 'Bezahlen' on the left and 'Abbruch' on the right.

Das SSL-Form für die Zahlung mit Paybox enthält ein Feld für die Telefonnummer. Sie wird über die Landesvorwahl, die Vorwahl und die Rufnummer abgebildet.



Das Java-Applet von Stackbox für die Online-Überweisung enthält alle Felder eines Überweisungsauftrags und zusätzlich Felder für PIN und TAN. Das Applet wird von Pago mit den Daten des Vorgangs (Kontonummer und Bankleitzahl des Empfängers, Betrag in Euro und Verwendungszweck) initialisiert. Der Kunde muss sein Online-Konto konfigurieren und auswählen. Ferner hat er die Möglichkeit, bis zu 27 Zeichen Freitext in die zweite Zeile des Verwendungszwecks einzutragen. Für die Durchführung der Transaktion ist dann nur noch das Ausfüllen der Felder PIN und TAN nötig. Je nach persönlicher Einstellung des Kontos durch den Kunden kann die PIN gespeichert werden. Die TAN muss jeweils vor der Transaktion eingegeben werden. Ferner werden PIN und TAN nur an das Bankensystem des Kunden übertragen.



Für die Anpassung der SSL-verschlüsselten Eingabeseiten, die im Zahlungsprozess vom Pago-Webserver geladen werden, sind folgende Komponenten nötig:

- angepasstes Template für die Zahlartauswahl
- angepasstes Template je Zahlart
- alle Grafiken und Logos.

Diese Dateien sollten Sie grafisch an Ihr Systemlayout anpassen.

Die fertiggestellte HTML-Seite (**.tpl**, template) senden Sie per E-Mail an Pago zurück. Sie wird auf dem Pago-Test- bzw. Produktivrechner abgelegt. Wenn der Kunde einen Zahlungsvorgang einleitet, wird die HTML-Seite auf dem Pago-Server aufgerufen (Redirect URL/Folge-URL).

9.7 Technische Schnittstellenbeschreibung

Der Service Pago Multipay wird integriert durch das Senden und Empfangen von XML-Dokumenten via LocalServer. Der Aufbau der benötigten XML-Dokumente wird nachfolgend beschrieben.

Die Pago-Multipay-Services im Modus GUI werden über folgende XML-Requests bereitgestellt:

- **MultipayInitGuiPaymentRequest** zum Initiieren von Zahlungen
- **MultipayInsertDebitRequest** zum Einstellen von Forderungsbuchungen
- **MultipaySettleAccountRequest** zum Anstoßen des Einzugs
- **MultipayReduceAmountRequest** zur Reduzierung des reservierten Betrags.

Die Fremdbelegnummer bezieht sich auf einen Vorgang im Mandantensystem, der über Pago Multipay abgewickelt wird (z. B. ein Kauf). Sie muss dem entsprechenden Vorgang beim Mandanten eindeutig zugeordnet werden können. Für die Fremdbelegnummer sollten ausschließlich Ziffern und Großbuchstaben verwendet werden, da das System Kleinbuchstaben automatisch in Großbuchstaben umsetzt. Falls zwei Fremdbelegnummern angelegt werden, die sich nur durch Groß- und Kleinschreibung voneinander unterscheiden, können sie vom System nicht mehr eindeutig zugeordnet werden.

! *Halten Sie sich bei Datumseingaben genau an das vorgegebene Format (YYYYMMDD oder DD.MM.YYYY). Lassen Sie keine Stelle aus, auch nicht die führende „0“ (z. B.: „01.11.2000“).*

Legende

Die Parameter des Request und des Response sind in Tabellen dargestellt. Für die Tabellen gilt folgende Legende:

Tag: Feldbezeichnung

Datentyp: „an“ = alphanumerisch / „n“ = numerisch
 Feldlänge: Anzahl der Zeichen
 Pflichtfeld: Angabe notwendig (Ja/Nein)



Bei numerischen Feldern, in denen eine Zahl mit Nachkommastellen eingetragen werden kann, sind unter „Feldlänge“ zwei Zahlen angegeben. Die Zahl vor dem Komma gibt die maximale Anzahl der gesamten Stellen an, die Zahl hinter dem Komma die maximale Anzahl der Nachkommastellen.

Beispiel: **10,2** --> max. 10 Stellen, davon max. 2 Nachkommastellen.

9.7.1 Initiieren von Zahlungen

Zahlungen werden mit **MultipayInitGuiPaymentRequest** initiiert. Der Vorgang wird durch eine Fremdbelegnummer eindeutig gekennzeichnet, die dem Vorgang zugeordnet werden muss.

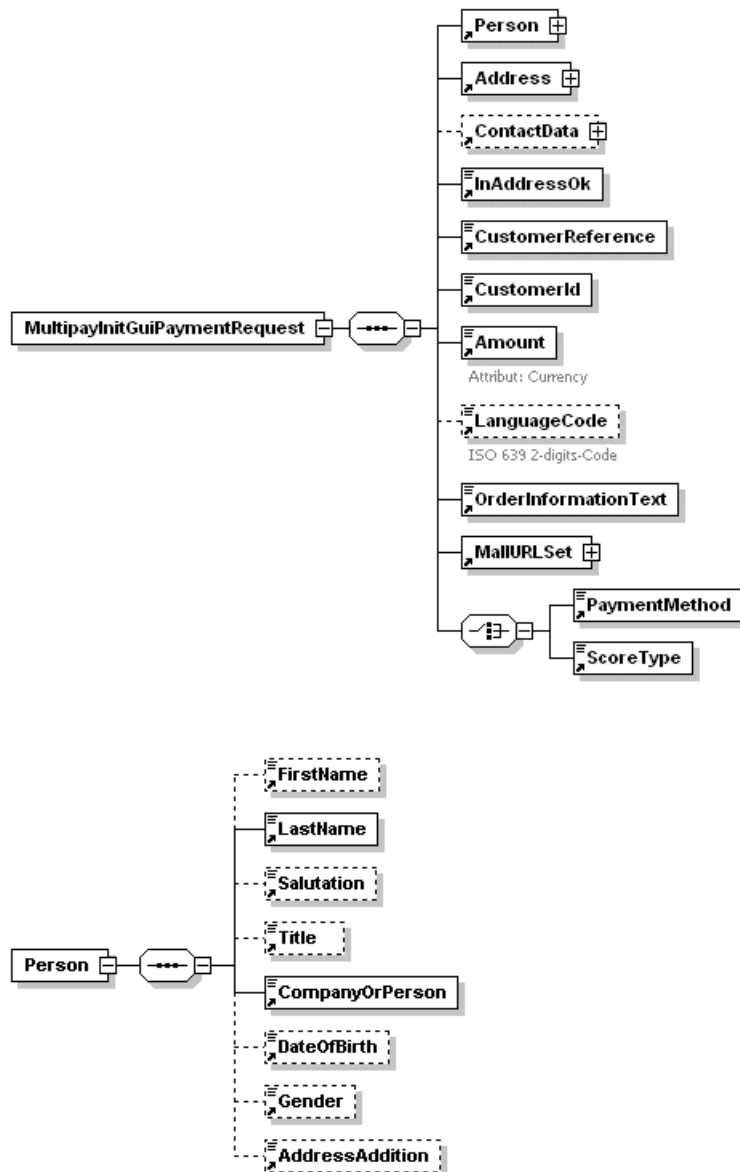


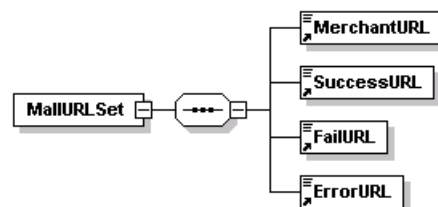
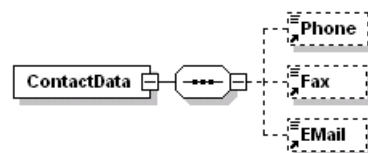
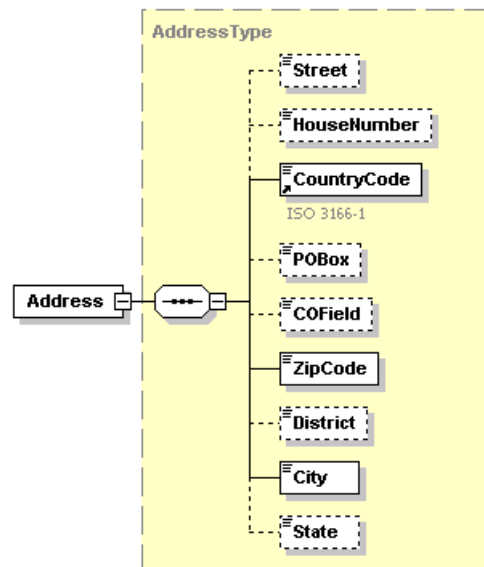
Bei wiederholten Aufrufen zur selben Fremdbelegnummer wird immer nur das Ergebnis zurückgeliefert, das zum ursprünglichen Vorgang dieser Fremdbelegnummer gehört (Returncode, Message).

Wenn die Zahlart von vornherein feststeht, werden die Pago Risk Management Services nicht benötigt. Dann wird die Zahlart direkt mit **PaymentMethod** angegeben.

Wenn dem Kunden Zahlarten zur Auswahl gestellt werden sollen, werden die Pago Risk Management Services und eine gültige Adresse benötigt. Über **ScoreType** wird der RMS-Typ definiert, und über **InAddressOk** wird eingestellt, ob es sich um eine schon zuvor geprüfte und gültige Adresse handelt. Über den Scorewert und die zur Verfügung stehenden Zahlarten wird eine Auswahl von Zahlarten generiert und dem Kunden auf der Zahlartauswahlseite (HTML) angezeigt.

Im Folgenden wird die Struktur von **MultipayInitGuiPaymentRequest** beschrieben:





Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
Person				
FirstName	an	30	Nein	Vorname
LastName	an	30	Ja	Nachname oder Firmenname
Salutation	an	10	Nein	Anrede, z. B. „Herr“, „Frau“, „Firma“
Title	an	20	Nein	Titel, z. B. „Dr.“, „Prof.“
CompanyOr- Person	char	1	Ja	F (Firma) oder P (Person)

Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
DateOfBirth	an	10	Nein	Geburtsdatum Format: YYYYMMDD oder DD.MM.YYYY
Gender	char	1	Nein	Geschlecht Fixwert: „“ (Leerstring)
Address- Addition	an	60	Nein	Adresszusatz z. B. „7. Stock“, „bei Maier“, „Firma Müller“, z. H. Herrn Schmidt“
Adress				
Street	an	46	Ja	Straßenname
HouseNumber	an	35	Ja	Hausnummer
CountryCode	an	3	Ja	gültiges Länderkennzeichen nach ISO 3166 (siehe Kapitel 10)
POBox	n	35	Nein	Postfach-Nummer
COField	an	40	Nein	Großempfänger-Name
ZipCode	an	9	Ja	Postleitzahl
District	an	40	Nein	Ortsteil
City	an	40	Ja	Ortsname
State	an	40	Nein	Staat (US-Bundesstaat)
ContactData				
Phone	an	20	Nein	Telefonnummer laut Pago Bill Muss mit „+“ und Ländervorwahl beginnen (z. B. „+49“).
Fax	an	20	Nein	Faxnummer laut Pago Bill Muss mit „+“ und Ländervorwahl beginnen (z. B. „+49“).
EMail	an	80	Nein	E-Mail-Adresse
InAddressOK	an	5	Ja	Liegt eine geprüfte, gültige Adresse vor? true oder false

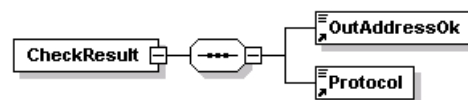
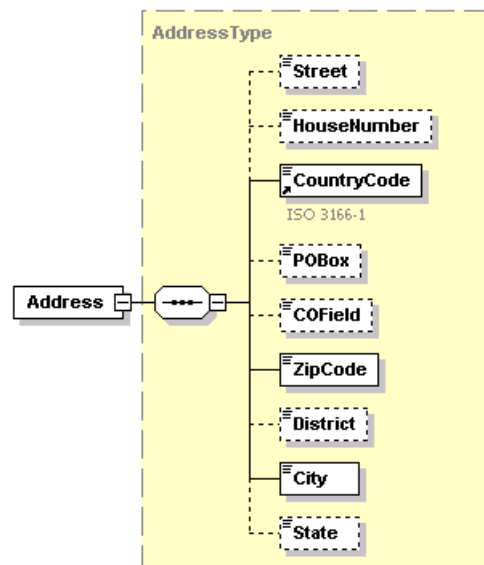
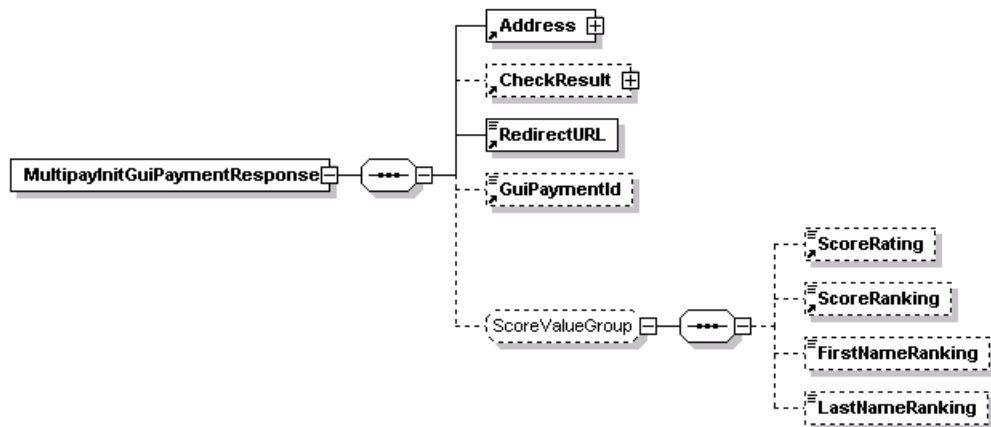
Tag	Daten- typ	Feld- länge	Pflicht- feld	Bemerkung
Customer- Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz unter ClientName / SalesChannel / Branch Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
CustomerId	an	20	Ja	Ihre eindeutige Kundenreferenz unter ClientName / SalesChannel / Branch
Amount				
Amount	n	10,2	Ja	positiver Bruttobetrag inkl. MwSt
Currency	an	3	Ja	Währung gemäß ISO 4217 (z. B. EUR, USD) Currency ist ein Attribut von Amount
LanguageCode	an	3	Nein	Sprachkennzeichen des Kunden Verwendung zukünftig im Mahnwesen
Order- Information	an	2000	Ja	Anzeigetext im Applet (z. B. für Bestellungen)
MallUrlSet				
MerchantUrl	an	512	Ja	URL für „Back-Button“
OkURL	an	512	Ja	URL für „Vorgang OK“
FailURL	an	512	Ja	URL für „Vorgang fehlgeschlagen“
ErrorURL	an	512	Ja	URL für „bei Systemfehler“
ScoreType oder PaymentMethod				
ScoreType	n	2	Ja	RMS-Typ 1, 2, 6 oder 7
PaymentMethod	an	3	Ja	Mit Pago abgesprochene Zahlart, z. B.: SEL Lastschriftverfahren (DTA) SKK Kreditkarteneinzug PBX Paybox SBX Stackbox NN Nachnahme REC Rechnungskauf VOR Vorkasse

Details über die Verwendung der Adressfelder bei Rechnungsdruck und Mahnwesen finden Sie in dem Dokument „PrintLayout“.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://kenny.dmz.pago.de:7777/xml/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-11-16T10:00:01+02:00">
<header>
  <communicationId>2001-11-16T10:00:01+01:00-1001200001-000001@bmp.multipay.gui</communicationId>
  <Client>
    <ClientName>123</ClientName>
    <SalesChannel>123</SalesChannel>
    <Branch>123</Branch>
  </Client>
  <retry>false</retry>
</header>
<request>
  <MultipayInitGuiPaymentRequest>
    <Person>
      <FirstName>Hans</FirstName>
      <LastName>Debtor</LastName>
      <Salutation>Herr</Salutation>
      <Title>Dr.</Title>
      <CompanyOrPerson>P</CompanyOrPerson>
      <DateOfBirth>31.12.1968</DateOfBirth>
      <Gender>M</Gender>
      <AddressAddition>App. 4</AddressAddition>
    </Person>
    <Address>
      <Street>reperbaan</Street>
      <HouseNumber>1</HouseNumber>
      <CountryCode>DE</CountryCode>
      <POBox></POBox>
      <COField></COField>
      <ZipCode>2000</ZipCode>
      <District></District>
      <City>Hamburg</City>
      <State></State>
    </Address>
    <InAddressOk>false</InAddressOk>
    <CustomerReference>987654323</CustomerReference>
    <CustomerId>13447</CustomerId>
    <Amount Currency="EUR">195.95</Amount>
    <LanguageCode>DE</LanguageCode>
    <OrderInformationText>EineOase</OrderInformationText>
    <MallURLSet>
      <MerchantURL>http://www.merchant.de</MerchantURL>
      <SuccessURL>http://www.merchant.de/Success.html</SuccessURL>
      <FailURL>http://www.merchant.de/Fail.html</FailURL>
      <ErrorURL>http://www.merchant.de/Error.html</ErrorURL>
    </MallURLSet>
    <ScoreType>1</ScoreType>
  </MultipayInitGuiPaymentRequest>
</request>
</PAGOxml>
```

Das Ergebnis der Initialisierung wird mittels **MultipayInitGuiPaymentResponse** von Pago geliefert. Es enthält die ggf. korrigierte Adresse, eine Transaktions-ID und die Pago-URL, auf die der Kunde zur Übernahme der Mandantensession umgeleitet wird.



Tag	Daten typ	Feld-länge	Bemerkung
Address			
Street	an	46	Straßenname
HouseNumber	an	35	Hausnummer
CountryCode	an	3	gültiges Länderkennzeichen nach ISO 3166 (siehe Kapitel 10)
POBox	n	35	Postfach-Nummer

Tag	Datentyp	Feldlänge	Bemerkung
COField	an	40	Großempfänger-Name
ZipCode	an	9	Postleitzahl
District	an	40	Ortsteil
City	an	40	Ortsname
State	an	40	Staat (US Bundesstaat)
CheckResult			
OutAddressOK	an	5	Angabe, ob die zurückgegebene Adresse korrekt ist: true : Adresse ist nicht korrekt false : Adresse ist korrekt
Protocol	an	8	detaillierte Angabe des Prüfergebnisses (siehe Abschnitt 9.4.1)
RedirectURL	an	512	Pago-URL zur Übernahme der Mandantensession
GuiPaymentID	n	20	zurückübermittelte eindeutige Vorgangs-ID
ScoreRating	n	5	Ermittelter Scorewert (bei RMS-Typen 1, 2, 6 oder 7)
ScoreRanking	an	1	Ermittelter Ampelwert (bei RMS-Typen 2 oder 6) „R“: red „Y“: yellow „G“: green
FirstNameRanking	n	3	Vornametrefter (nur bei RMS-Typ 6)
LastNameRanking	n	3	Nachnametrefter (nur bei RMS-Typ 6)

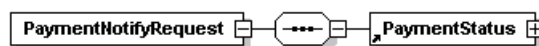
Zu den Rückgabewerten, die mit dem Aufruf bereitgestellt werden, gehört die asynchrone Rückmeldung **PaymentNotifyRequest**. Mit **PaymentNotifyRequest** wird der Zahlstatus aus der Verarbeitung im Applet bzw. im SSL-Form übermittelt.

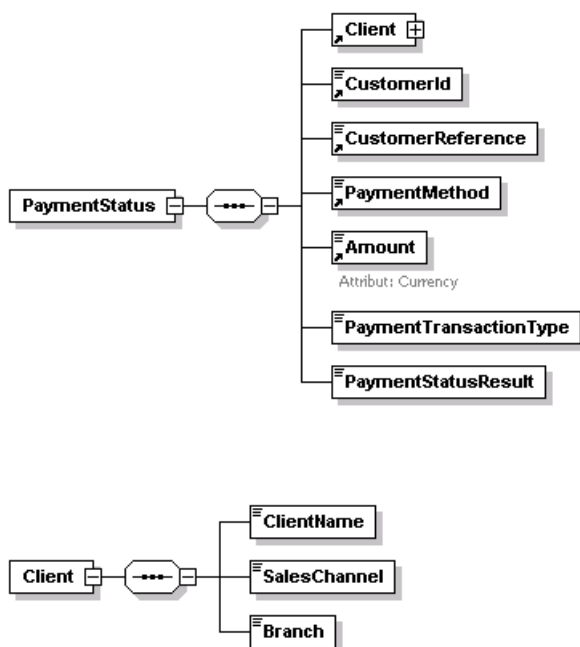
Fehlercode	Grund	Abhilfe
102.001	Nicht-boole'scher Wert	true oder false verwenden
102.002	kein gültiger numerischer Wert und nicht leer	leer lassen oder Zahl verwenden
102.003	weder ScoreType noch PaymentMethod angegeben	eins von beiden muss gefüllt sein

Fehlercode	Grund	Abhilfe
102.004	Currency ist für diesen Dienst Pflichtfeld	Currency muss gefüllt sein
102.500	Ungültige Mandantenstruktur	
102.501	Ungültige Personendaten	
102.502	Ungültige Adressdaten	
102.503	Ungültige URLs	
102.504	Ungültiger Language Code	
102.505	Ungültiger RMS-Typ	
102.506	Interner Fehler (z. B. Kundenkreis fehlt / Transaktionsart fehlt / Status unbekannt / Doublettenprüfung)	
102.507	Ungültiger Mandant	
102.508	Ungültige Filiale	
102.509	Ungültige Kundendaten	
102.510	Transaktion wurde schon autorisiert	
102.511	Autorisierung in Bearbeitung	
102.513	Ungültiger Betrag	
102.514	Sequenznummer ist ungültig	
102.516	Das Format des Länderkennzeichens ist nicht korrekt oder wurde nicht mit angegeben.	

9.7.2 Rückmeldung des Payment-Status

Wenn die Autorisierung oder Zahlung mit Multipay GUI durchgeführt wurde, entnimmt Pago das Ergebnis der Autorisierung bzw. der Zahlung aus der Rückmeldung der Kreditkartengesellschaft bzw. der Bank und teilt Ihnen dieses Ergebnis über **PaymentNotifyRequest** im Payment-Status mit. **PaymentNotifyRequest** (asynchrone Rückmeldung) wird unmittelbar im Anschluss an die Autorisierung / Zahlung erstellt und versendet. Der Vorgang ist über **CustomerReference** definiert. Ergebnisse aus später durchgeführten Captures, Gutschriften, etc. werden nach der Batch-Verarbeitung bei Pago erstellt und versendet.





Tag	Daten typ	Feld-länge	Bemerkung
PaymentStatus			
Client			
ClientName	an	20	Name des Mandanten Konstante in Absprache mit Pago
SalesChannel	an	20	Vertriebslinie Konstante in Absprache mit Pago
Branch	an	20	Filiale Konstante in Absprache mit Pago
CustomerId	an	20	Ihre eindeutige Kundenreferenz unter ClientName / SalesChannel / Branch
Customer-Reference	an	20	Ihre eindeutige Vorgangsreferenz unter ClientName / SalesChannel / Branch (siehe auch Abschnitt 9.7) Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).

Tag	Daten typ	Feldlänge	Bemerkung
PaymentMethod	an	3	Mit Pago abgesprochene Zahlart, z. B.: SEL : Lastschriftverfahren (DTA) SKK : Kreditkarteneinzug PBX : Paybox SBX : Stackbox NN : Nachnahme REC : Rechnungskauf VOR : Vorkasse
Amount			
Amount	n	10,2	Saldo des Vorgangskontos negativer Betrag: Guthaben positiver Betrag: Forderung
Currency	an	3	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest
Payment-Transaction-Type	an	25	Transaktionsart Authorization : Betragsautorisierung Capture : Einlösen auf eine Autorisierung Credit : Auszahlung Gutschrift Exception : Verarbeitung wurde unterbrochen; neuen Status abwarten PBXCapture : Einlösen auf eine Paybox-Autorisierung PBXCaptureCancel : Einlösen widerrufen und damit auszahlen Payment : Sofortbelastung des Betrags auf der Kreditkarte PaymentAfterFailedCapture : Sofortbelastung des Betrags nach fehlgeschlagenem Einlösen auf eine Autorisierung
PaymentStatus-Result	an	5	Ergebnis der Verarbeitung OK : Verarbeitung erfolgreich durchgeführt Failed : Verarbeitung fehlgeschlagen Error : Fehler während der Ausführung

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:33+02:00">
<header>
  <communicationId>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de>
  <communicationId>
  <Server>PagoServer</Server>
  <retry>false</retry>
</header>
<request>
  <PaymentNotifyRequest>
    <PaymentStatus>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
      <CustomerId>123456789001</CustomerId>
      <CustomerReference>abcdefg2001</CustomerReference>
      <PaymentMethod>ELV</PaymentMethod>
      <Amount Currency="EUR">195.95</Amount>
      <PaymentTransactionType>Payment</PaymentTransactionType>
      <PaymentStatusResult>OK</PaymentStatusResult>
    </PaymentStatus>
  </PaymentNotifyRequest>
</request>
</PAGOxml>
```

Auf **PaymentNotifyRequest** wird von Pago ein **PaymentNotifyResponse** erwartet. Erst mit dem Eingang eines leeren **PaymentNotifyResponse** ohne **error-Notification** gilt die Zustellung des Status als abgeschlossen.

PaymentNotifyResponse

Damit hat das Tag **PaymentNotifyResponse** keine definierten Elemente.

Tag	Daten typ	Feld- länge	Bemerkung
PaymentNotifyResponse			

Das folgende Beispiel zeigt einen möglichen **PaymentNotifyResponse**.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml timeStamp="2001-10-30T16:45:25+01:00" version="2.0">
  <header>
    <communicationId>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url>
    <communicationId>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
    <retry>false</retry>
    <inReplyTo>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de
```

```

</inReplyTo>
</header>
<response>
  <PaymentNotifyResponse/>
</response>
</PAGOxml>

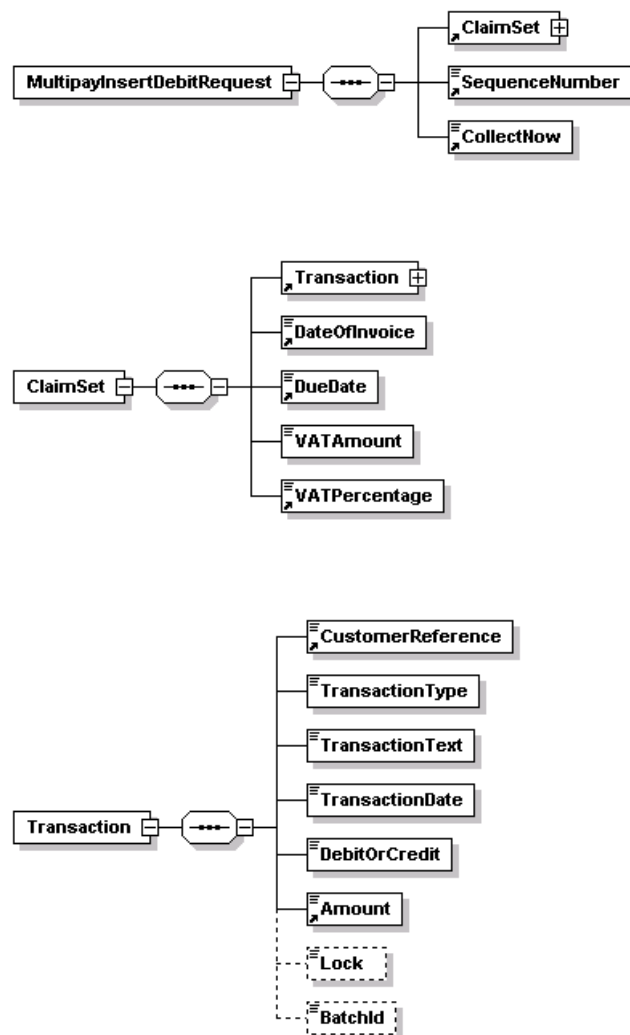
```

9.7.3 Einstellen von Forderungsbuchungen

Eine Forderungsbuchung stellen Sie mit **MultipayInsertDebitRequest** ein.



Achten Sie darauf, dass die Summe der eingestellten Forderungen nicht höher ist als der bei **MultipayInitPaymentRequest** angegebene Betrag.



Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:

Tag	Datentyp	Feldlänge	Pflichtfeld	Bemerkung
ClaimSet				
Transaction				
Customer-Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz wie in MultipayInit-PaymentRequest angegeben Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
Transaction-Type	an	6	Ja	Transaktionsart FD = Forderung Ware / Dienstleistung VD = Versandkosten GS = Gutschrift RT = Retoure VO = Voucher ! Die Transaktionsart bestimmt nicht, ob der Betrag im Soll oder Haben gebucht wird.
Transaction-Text	an	30	Ja	Transaktionstext für Buchhaltung / Kontoauszug
Transaction-Date	an	10	Ja	Transaktionsdatum (Kaufdatum) Format: YYYYMMDD oder DD.MM.YYYY
DebitOrCredit	an	1	Ja	Soll oder Haben S : Soll (debit)(FD/VD/GS/RT) Für alle Buchungen ist S zu verwenden. H : Haben (credit) nur intern bei Pago
Amount				
Amount	n	10,2	Ja	Bruttobetrag inkl. MwSt + : Forderung oder Stornierung einer Gutschrift - : Gutschrift oder Stornierung einer Forderung

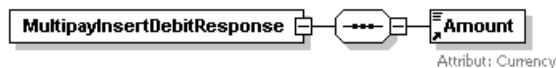
Tag			Daten typ	Feld- länge	Pflicht- feld	Bemerkung
		Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPayment- Request
		Lock	-	-	Nein	Reserviert
		BatchId	an	30	Nein	Batch-ID Dient bei Bedarf zur Gruppierung von Sätzen (nur nach Absprache)
		DateOfInvoice	an	10	Ja	Rechnungsdatum Format: YYYYMMDD oder DD.MM.YYYY
		DueDate	an	10	Ja	Fälligkeitsdatum Format: YYYYMMDD oder DD.MM.YYYY
		VATAmount	n	10,2	Ja	Mehrwertsteuerbetrag oder -1 Bei -1 keine Angabe. Wird von Pago berechnet, wenn ein MwSt-Satz angegeben ist.
		VATPercentage	n	4,2	Ja	Mehrwertsteuer in Prozent -1: keine Angabe (Pago bucht keine MwSt im Debitorensystem)
		SequenceNumber	n	6	Ja	Sequenznummer Muss mit „1“ beginnen und wächst mit jedem Folgeaufruf um 1 (siehe auch Abschnitt 9.3.2).
		CollectNow	n	1	Ja	Einzug der bisher über Multi- payInsertDebitRequest eingestellten Sätze. Je MultipayInitPayment- Request sind mehrere Einzüge möglich. Jeder Einzug kann mehrere Debit-Sätze enthalten. true = jetzt einziehen false = nicht einziehen

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="Thu May 17 13:42:15 2001">
```

```
<header>
  <communicationId>
    2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
  </communicationId>
  <Client>
    <ClientName>123</ClientName>
    <SalesChannel>123</SalesChannel>
    <Branch>123</Branch>
  </Client>
  <retry>false</retry>
</header>
<request>
  <MultipayInsertDebitRequest>
    <ClaimSet>
      <Transaction>
        <CustomerReference>abcdefg2001</CustomerReference>
        <TransactionType>FD</TransactionType>
        <TransactionText>Beispiel Forderung</TransactionText>
        <TransactionDate>30.12.2002</TransactionDate>
        <DebitOrCredit>S</DebitOrCredit>
        <Amount Currency="EUR" >99.99</Amount>
        <BatchId></BatchId>
      </Transaction>
      <DateOfInvoice>16.07.2001</DateOfInvoice>
      <DueDate>16.10.2001</DueDate>
      <VATAmount>13.79</VATAmount>
      <VATPercentage>16.0</VATPercentage>
    </ClaimSet>
    <SequenceNumber>1</SequenceNumber>
    <CollectNow>false</CollectNow>
  </MultipayInsertDebitRequest>
</request>
</PAGOxml>
```

Als Rückgabewert auf **MultipayInsertDebitRequest** wird neben dem Ergebnis der einzuziehende Betrag gemeldet („0“ bei **CollectNow** = **false**).



Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
Amount				
Amount	n	10,2	Ja	Einzuziehender Betrag
Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPayment- Request

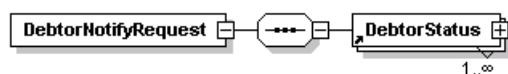
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml timeStamp="2001-10-30T16:45:25+01:00" version="2.0">
```

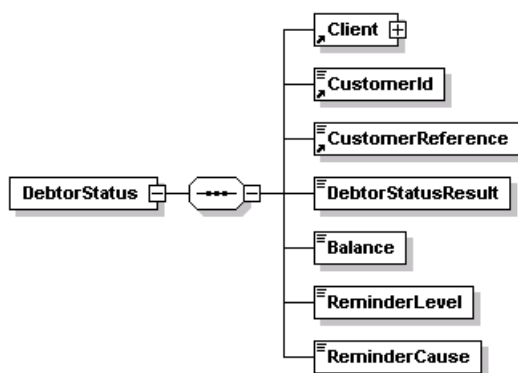
```
<header>
  <communicationId>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de
</communicationId>
  <Server> PagoServer </Server>
  <retry> false </retry>
  <inReplyTo>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
</inReplyTo>
</header>
<response>
  <MultipayInsertDebitResponse>
    <Amount Currency="EUR"> 0 </Amount>
  </MultipayInsertDebitResponse>
</response>
</PAGOXml>
```

Fehlercode	Grund	Abhilfe
103.001	Nicht-boole'scher Wert	true oder false verwenden
103.002	Kein gültiger numerischer Wert	Zahl verwenden
103.003	Parameterlänge zu lang oder zu kurz	Parametername steht im Twister-Log
103.500	Ungültige Mandanten-Identifikation	
103.505	Abweichende Währung	
103.506	Ungültige Transaktionsart	
103.507	Ungültige Autorisierung	
103.508	Systemfehler	
103.509	Systemfehler	
103.513	Einlösungsfehler	
103.516	Ungültige Sequenznummer	

9.7.4 Empfangen des Debitorenstatus

Unabhängig von **MultipayInsertDebitRequest** kann sich der Debitorenstatus verändern. Der jeweils aktuelle Debitorenstatus wird (ohne Historie) von Pago via **Debtor-NotifyRequest** bereitgestellt. Dabei werden mit einem Request alle zur Verfügung stehenden Debitorenstatus versendet.





Tag	Daten typ	Feld-länge	Bemerkung
DebtorStatus			
Client			
ClientName	an	20	Name des Mandanten Konstante in Absprache mit Pago
SalesChannel	an	20	Vertriebslinie Konstante in Absprache mit Pago
Branch	an	20	Filiale Konstante in Absprache mit Pago
CustomerId	an	20	Ihre eindeutige Kundenreferenz unter ClientName / SalesChannel / Branch
Customer-Reference	an	20	Ihre eindeutige Vorgangsreferenz unter ClientName / SalesChannel / Branch (siehe auch Abschnitt 9.7) Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
DebtorStatus-Result	an	1	Debitorenstatus N : nicht bezahlt B : bezahlt, Vorgang ist ausgeglichen Z : überzahlt F : überfällig/in Mahnung, Vorgang ist nicht ausgeglichen (Ursache kann eine fehlgeschlagene Zahlung oder eine Rückforderung sein.)

Tag	Daten typ	Feld- länge	Bemerkung
Balance			
Balance	n	15,2	Saldo des Vorgangskontos Negativer Betrag: Guthaben Positiver Betrag: Forderung
Currency	an	3	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest
ReminderLevel	an	1	Aktuelle Mahnstufe 0: keine Mahnung ... 4: Inkasso A: Ausgebucht
ReminderCause	an	3	Mahngrund RL0: keine Deckung RL1: Konto erloschen RL2: Konto-Nr. / Name nicht identisch (falsch oder Sparkonto) RL3: kein Abbuchungsauftrag bzw. keine Einzugsermächtigung RL4: Rückruf RL5: Widerspruch RL6: Rückgabe LSV: Zahlart Lastschrift, nicht einziehbar CB: Chargeback KKE: Zahlart Kreditkarte, nicht einziehbar

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
<header>
  <communicationId> 2001-12-31T12:30:30+02:00-001122334455-00601@10.10.10.10
</communicationId>
  <Server>
    PagoServer
  </Server>
  <retry>
    true
  </retry>
</header>
<request>
  <DebtorNotifyRequest>
    <DebtorStatus>
      <Client>
        <ClientName>123</ClientName>
        <SalesChannel>123</SalesChannel>
        <Branch>123</Branch>
      </Client>
    </DebtorStatus>
  </DebtorNotifyRequest>
</request>
</PAGOxml>
```

```
<CustomerId>123456789001</CustomerId>
<CustomerReference>abcdefg2001</CustomerReference>
<DebtorStatusResult>B</DebtorStatusResult>
<Balance Currency="EUR">0</Balance>
<ReminderLevel>0</ReminderLevel>
<ReminderCause> </ReminderCause>
</DebtorStatus>
<DebtorStatus>
  <Client>
    <ClientName>123</ClientName>
    <SalesChannel>123</SalesChannel>
    <Branch>123</Branch>
  </Client>
  <CustomerId>123456789002</CustomerId>
  <CustomerReference>abcdefg1999</CustomerReference>
  <DebtorStatusResult>B</DebtorStatusResult>
  <Balance Currency="EUR">0</Balance>
  <ReminderLevel>0</ReminderLevel>
  <ReminderCause> </ReminderCause>
</DebtorStatus>
<DebtorStatus>
  <Client>
    <ClientName>123</ClientName>
    <SalesChannel>123</SalesChannel>
    <Branch>123</Branch>
  </Client>
  <CustomerId>123456789000</CustomerId>
  <CustomerReference>abcdefg2000</CustomerReference>
  <DebtorStatusResult>F</DebtorStatusResult>
  <Balance Currency="EUR">34,95</Balance>
  <ReminderLevel>A</ReminderLevel>
  <ReminderCause>LSV</ReminderCause>
</DebtorStatus>
</DebtorNotifyRequest>
</request>
</PAGOxml>
```

Auf **DebtorNotifyRequest** wird von Pago ein **DebtorNotifyResponse** erwartet. Erst mit dem Eingang eines leeren **DebtorNotifyResponse** ohne **error-Notification** gilt die Zustellung des Status als abgeschlossen.

DebtorNotifyResponse

Damit hat das Tag **DebtorNotifyResponse** keine definierten Elemente.

Tag	Daten typ	Feld- länge	Bemerkung
DebtorNotifyResponse			

Das folgende Beispiel zeigt einen möglichen **DebtorNotifyResponse**:

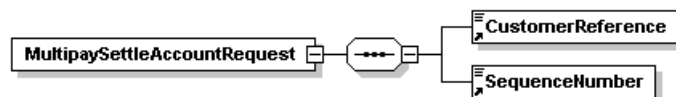
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml timeStamp="2001-10-30T16:45:25+01:00" version="2.0">
  <header>
```

```
<communicationId>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url>
<communicationId>
  <Client>
    <ClientName>123</ClientName>
    <SalesChannel>123</SalesChannel>
    <Branch>123</Branch>
  </Client>
  <retry>false</retry>
  <inReplyTo>2001-12-31T12:30:33+02:00-001122334458-00601@www.pago.de
  </inReplyTo>
</header>
<response>
  <DebtorNotifyResponse/>
</response>
</PAGOfxml>
```

9.7.5 Anstoßen des Einzugs

Nach dem Einstellen von einer oder mehreren Forderungsbuchungen wird mit **MultipaySettleAccountRequest** der Einzug freigegeben. Der Einzug erfolgt in der Regel asynchron. Ergebnisse werden von Pago über **DebtorNotifyRequest** gemeldet.

Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:

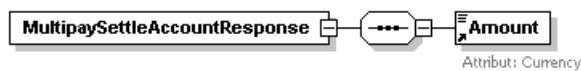


Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
Customer- Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz wie in MultipayInitPaymentRequest angegeben Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
SequenceNumber	n	6	Ja	Sequenznummer Muss mit „1“ beginnen und wächst mit jedem Folgeaufruf um 1 (siehe auch Abschnitt 9.3.2).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>false</retry>
  </header>
  <request>
    <MultipaySettleAccountRequest>
      <CustomerReference>gfdgdfgd54543453</CustomerReference>
      <SequenceNumber>2</SequenceNumber>
    </MultipaySettleAccountRequest>
  </request>
</PAGOxml>
```

Zu **MultipaySettleAccountRequest** wird mittels **MultipaySettleAccount-Response** der einzuziehende Betrag zurückgemeldet.



Tag	Daten typ	Feld- länge	Bemerkung
Amount			
Amount	n	10,2	Einzuziehender Betrag
Currency	an	3	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest

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

<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>2001-12-31T12:34:30+02:00-001122334459-00601@www.pago.de
    </communicationId>
    <Server> PagoServer </Server>
    <retry> false </retry>
    <inReplyTo>2001-12-31T12:02:30+02:00-000000000126-00001@www.client.url
    </inReplyTo>
  </header>
  <response>
    <MultipaySettleAccountResponse>
      <Amount Currency="EUR"> 99.99 </Amount>
    </MultipaySettleAccountResponse>
  </response>
```

</PAG0xml>

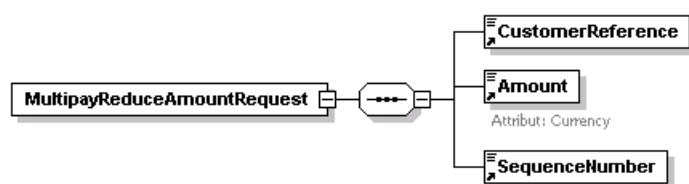
Fehlercode	Grund	Abhilfe
105.002	Kein numerischer Wert	Numerischen Wert einsetzen
105.004	Currency ist Pflicht in diesem Service	Währung einsetzen
105.500	Aus DB: MSA_UNGUELTIGER_KLIENT	
105.501	Aus DB: MSA_INVALID_PARAMETERS	
105.502	Aus DB: MSA_UNKNOWN_TRANSACTION_IDENTIFICATION	
105.503	Aus DB: MSA_INVALID_AUTHORIZATION	
105.505	Aus DB: MSA_INVALID_SEQUENCE_NUMBER	

9.7.6 Reduzierung des reservierten Betrags

Der Request **MultipayReduceAmountRequest** informiert Pago, wenn der bei **MultipayInitPaymentRequest** angegebene Betrag reduziert wurde. Diese Reduzierung wird bei später eingestellten Debit-Sätzen berücksichtigt.

Wenn bei einem Vorgang mit Betragsreservierung (**KKE**, **PBX**) der Betrag in **MultipayReduceAmountRequest** auf „0“ gesetzt wird (Komplettstorno), wird die Reservierung storniert. Bei Teilstornos wird die Betragsreservierung aufrechterhalten.

Folgende Parameter werden in dem Funktionsaufruf an den Pago-Server geleitet:



Tag	Daten typ	Feld- länge	Pflicht- feld	Bemerkung
Customer- Reference	an	20	Ja	Ihre eindeutige Vorgangsreferenz wie in MultipayInitPayment-Request angegeben Die folgenden Zeichen sind erlaubt: <ul style="list-style-type: none"> • Ziffern „0“ bis „9“ • Großbuchstaben „A“ bis „Z“ • Trennzeichen „.“ (Punkt), „-“ (Bindestrich), „+“ (Plus) und „/“ (Schrägstrich).
Amount				
Amount	n	10,2	Ja	Neuer Gesamtbetrag des Vorgangs
Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPayment-Request
SequenceNumber	n	6	Ja	Sequenznummer Muss mit „1“ beginnen und wächst mit jedem Folgeaufruf um 1 (siehe auch Abschnitt 9.3.2).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>
      2001-12-31T12:02:30+02:00-000000000127-00001@www.client.url
    </communicationId>
    <Client>
      <ClientName>123</ClientName>
      <SalesChannel>123</SalesChannel>
      <Branch>123</Branch>
    </Client>
    <retry>false</retry>
  </header>
  <request>
    <MultipayReduceAmountRequest>
      <CustomerReference>gfdgdfgd54543453</CustomerReference>
      <Amount Currency="EUR">99.99</Amount>
      <SequenceNumber>3</SequenceNumber>
    </MultipayReduceAmountRequest>
  </request>
</PAGOxml>
```

Als Rückgabewert auf **MultipayReduceAmountRequest** wird der verbleibende Reservierungsbetrag gemeldet.



Tag	Daten typ	Feld-länge	Pflicht-feld	Bemerkung
Amount				
Amount	n	10,2	Ja	Verbleibender Reservierungsbetrag
Currency	an	3	Ja	Währung gemäß ISO 4217 Muss identisch sein mit Angabe in MultipayInitPaymentRequest

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE PAGOxml SYSTEM "http://xml.pago.de/2.0/PAGOxml.dtd">
<PAGOxml version="2.0" timeStamp="2001-12-31T12:30:30+02:00">
  <header>
    <communicationId>2001-12-31T12:35:30+02:00-001122334460-00601@www.pago.de
    </communicationId>
    <Server> PagoServer </Server>
    <retry> false </retry>
    <inReplyTo>2001-12-31T12:02:30+02:00-000000000127-00001@www.client.url </inReplyTo>
  </header>
  <response>
    <MultipayReduceAmountResponse>
      <Amount Currency="EUR"> 0 </Amount>
    </MultipayReduceAmountResponse>
  </response>
</PAGOxml>
  
```

Fehlercode	Grund	Abhilfe
104.002	Wert muss numerisch sein	Zahl einsetzen
104.004	Currency ist Pflicht für diesen Dienst	Währung angeben
104.500	Aus DB: MPSA_UNKNOWN_CLIENT	
104.504	Währung weicht von InitPayment ab	Gleiche Währung wie bei InitPayment verwenden

Fehlercode	Grund	Abhilfe
104.505	Ungültige Währung	Gleiche Währung wie bei InitPayment verwenden
104.506	Ungültige Autorisierung	InitPayment wurde nicht erfolgreich beendet
104.509	Ungültige Sequenznummer	

10 Die wichtigsten Länderkennzeichen laut ISO 3166

A

Afghanistan: AF; Albania: AL; Algeria: DZ; American Samoa: AS; Andorra: AD; Angola: AO; Anguilla: AI; Antarctica: AQ; Antigua und Barbuda: AG; Argentina: AR; Armenia: AM; Aruba: AW; Australia: AU; Austria: AT; Azerbaijan: AZ

B

Bahamas: BS; Bahrain: BH; Bangladesh: BD; Barbados: BB; Belarus: BY; Belgium: BE; Belize: BZ; Benin: BJ; Bermuda: BM; Bhutan: BT; Bolivia: BO; Bosnia and Herzegowina: BA; Botswana: BW; Bouvet Island: BV; Brazil: BR; British Indian Ocean Territory: IO; Brunei Darussalam: BN; Bulgaria: BG; Burkina Faso: BF; Burundi: BI

C

Cambodia: KH; Cameroon: CM; Canada: CA; Cape Verde: CV; Cayman Islands: KY; Central African Republic: CF; Chad: TD; Chile: CL; China: CN; Christmas Island: CX; Cocos (Keeling) Islands: CC; Colombia: CO; Comoros: KM; Congo: CG; Congo, The Democratic Republic of the: CD; Cook Islands: CK; Costa Rica: CR; Cote D'Ivoire: CI; Croatia (local name: Hrvatska): HR; Cuba: CU; Cyprus: CY; Czech Republic: CZ

D

Denmark: DK; Djibouti: DJ; Dominica: DM; Dominican Republic: DO

E

East Timor: TP; Ecuador: EC; Egypt: EG; El Salvador: SV; Equatorial Guinea: GQ; Eritrea: ER; Estonia: EE; Ethiopia: ET

F

Falkland Islands (Malvinas): FK; Faroe Islands: FO; Fiji: FJ; Finland: FI; France: FR; France, Metropolitan: FX; French Guiana: GF; French Polynesia: PF; French Southern Territories: TF

G

Gabon: GA; Gambia: GM; Georgia: GE; Germany: DE; Ghana: GH; Gibraltar: GI; Greece: GR; Greenland: GL; Grenada: GD; Guadeloupe: GP; Guam: GU; Guinea: GN; Guinea Bissau: GW; Guyana: GY; Guatemala: GT

H

Haiti: HT; Heard and Mc Donald Islands: HM; Holy See (Vatican City State): VA;
Honduras: HN; Hong Kong: HK; Hungary: HU

I

Iceland: IS; India: IN; Indonesia: ID; Iran (Islamic Republic of): IR; Iraq: IQ;
Ireland: IE; Israel: IL; Italy: IT

J

Jamaica: JM; Japan: JP; Jordan: JO

K

Kazakhstan: KZ; Kenya: KE; Kiribati: KI; Korea, Democratic People's Republic of: KP;
Korea, Republic of: KR; Kuwait: KW; Kyrgyzstan: KG

L

Lao People's Democratic Republic: LA; Latvia: LV; Lebanon: LB; Lesotho: LS;
Liberia: LR; Libyan Arab Jamahiriya: LY; Liechtenstein: LI; Lithuania: LT;
Luxembourg: LU

M

Macau: MO; Macedonia, The Former Yugoslav Republic of: MK; Madagascar: MG;
Malawi: MW; Malaysia: MY; Maldives: MV; Mali: ML; Malta: MT; Marshall Islands: MH;
Martinique: MQ; Mauritania: MR; Mauritius: MU; Mayotte: YT; Mexico: MX; Microne-
sia, Federated States of: FM; Moldova, Republic of: MD; Monaco: MC; Mongolia: MN;
Montserrat: MS; Morocco: MA; Mozambique: MZ; Myanmar: MM

N

Namibia: NA; Nauru: NR; Nepal: NP; Netherlands: NL; Netherlands Antilles: AN; New
Caledonia: NC; New Zealand: NZ; Nicaragua: NI; Niger: NE; Nigeria: NG; Niue: NU;
Norfolk Island: NF; Northern Mariana Islands: MP; Norway: NO

O

Oman: OM

P

Pakistan: PK; Palau: PW; Panama: PA; Papua New Guinea: PG; Paraguay: PY;
Peru: PE; Philippines: PH; Pitcairn: PN; Poland: PL; Portugal: PT; Puerto Rico: PR

Q

Qatar: QA

R

Reunion: RE; Romania: RO; Russian Federation: RU; Rwanda: RW

S

Saint Kitts and Nevis: KN; Saint Lucia: LC; Saint Vincent and the Grenadines: VC; Samoa: WS; San Marino: SM; Sao Tome and Principe: ST; Saudi Arabia: SA; Senegal: SN; Seychelles: SC; Sierra Leone: SL; Singapore: SG; Slovakia (Slovak Republic): SK; Slovenia: SI; Solomon Islands: SB; Somalia: SO; South Africa: ZA; South Georgia and the South Sandwich Islands: GS; Spain: ES; Sri Lanka: LK; St. Helena: SH; St. Pierre and Miquelon: PM; Sudan: SD; Suriname: SR; Svalbard and Jan Mayen Islands: SJ; Swaziland: SZ; Sweden: SE; Switzerland: CH; Syrian Arab Republic: SY

T

Taiwan, Province of China: TW; Tajikistan: TJ; Tanzania, United Republic of: TZ; Thailand: TH; Togo: TG; Tokelau: TK; Tonga: TO; Trinidad and Tobago: TT; Tunisia: TN; Turkey: TR; Turkmenistan: TM; Turks and Caicos Islands: TC; Tuvalu: TV

U

Uganda: UG; Ukraine: UA; United Arab Emirates: AE; United Kingdom: GB; United States: US; United States Minor Outlying Islands: UM; Uruguay: UY; Uzbekistan: UZ

V

Vanuatu: VU; Venezuela: VE; Vietnam: VN; Virgin Islands (British): VG; Virgin Islands (U.S.): VI

W

Wallis and Futuna Islands: WF; Western Sahara: EH

Y

Yemen: YE; Yugoslavia: YU

Z

Zambia: ZM; Zimbabwe: ZW

Notizen