



---

**Financial Management Service**  
**Financial Transaction Implementation Guide**

---

**October 2007**

**Release 2.1**

## Table of Contents

1	Introduction.....	4
1.1	Overview.....	4
1.2	Content Model Diagrams.....	4
2	XML Implementation.....	7
2.1	Model Overview.....	7
2.2	Transmission.....	8
2.3	Batch.....	9
2.4	Account Summary.....	10
2.5	Business Transaction.....	11
2.6	Financial Transaction.....	12
2.6.1	Payer and Payee.....	14
2.6.2	Bank Info.....	15
2.6.3	Plastic.....	15
2.6.4	Addendum.....	16
2.7	Account Classification.....	17
2.8	User Defined Data / Program Data.....	18
2.9	Transmission Response Model.....	19
3	Financial Transactions and Payment Medium Examples.....	21
3.1	Automated Clearing House (ACH).....	21
3.2	Check.....	23
3.3	Fedwire.....	24
3.4	Plastic.....	25
3.5	Cash.....	26
4	Examples.....	26
4.1	Multiple Financial Transactions.....	26
4.2	Multiple TAS-BETC or Ckey Name-Value.....	27

## List of Figures

Figure 1-1: Sequence Compositor .....	5
Figure 1-2: Choice Compositor .....	5
Figure 1-3: Mandatory Single Element.....	5
Figure 1-4: Single Optional Element.....	5
Figure 1-5: Mandatory Multiple Element Containing Child Elements .....	6
Figure 1-6: Attributes.....	6
Figure 2-1: Business Transaction Standard Model .....	7
Figure 2-2: Transmission Model.....	8
Figure 2-3: Batch Model.....	9
Figure 2-4: AccountSummary Model .....	10
Figure 2-5: Business Transaction Model .....	12
Figure 2-6: Financial Transaction Model .....	13
Figure 2-7: Payer and Payee Model.....	14
Figure 2-8: BankInfo Model .....	15
Figure 2-9: Plastic Model.....	16
Figure 2-10: Addendum Model .....	16
Figure 2-11: Accounting Classification Model.....	17
Figure 2-12: Program Data Model.....	18
Figure 2-13: TransmissionRs Model .....	19
Figure 2-14: Success Model.....	19
Figure 2-15: Error and Warning Model.....	20
Figure 3-1: ACH Transaction (www.nacha.org) .....	21

## 1 Introduction

### 1.1 Overview

The purpose of this document is to describe how to implement Financial Management Service's (FMS) Financial Transaction Implementation using W3C XML Schemas. This document is intended for data architects, requirements analysts, project managers and software developers of financial management systems that interface with the operation of Financial Management Service.

This Implementation Guide is divided into the following four sections:

**Section 1: Document Conventions** – This is an introduction on how to view the diagrams and models that are described in this document.

**Section 2: XML Implementation** – This section covers the different models used within the Financial Transaction Standard in order for the audience to understand the composition of the models, mandatory and optional elements, cardinality, attributes and specific implementation details for the elements.

**Section 3: Financial Transaction Payment Medium Examples** – The purpose of this section is to cover examples of each payment medium used for financial transactions. This supports developers who may need to implement one type of financial transaction.

**Section 4: Examples** – This section describes unique examples that developers may need to implement that may not be intuitive from the previous sections. This section is also intended to clarify specific cases from developer comments and situations.

### 1.2 Content Model Diagrams

A content model is a description of the structure and contents of an element. All content model diagrams in this document have been generated with Altova XML Spy. This description of the content model is adapted from XML Spy's user documentation.

#### Compositor

The compositor specifies either a sequential (*sequence*) or a disjunctive (*choice*) interpretation of the elements. This in turn determines whether the element's children must be in order or correspond to one choice of the specified options.

## Sequence

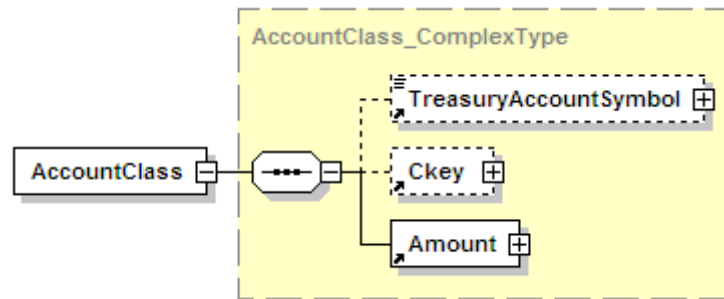


Figure 1-1: Sequence Compositor

The sequence compositor is used when the elements must be specified in the order of the definition. In this case `TreasuryAccountSymbol` must be specified before `CKey`.

## Choice

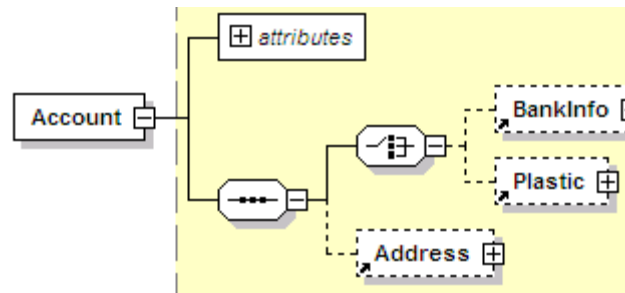


Figure 1-2: Choice Compositor

The choice compositor is used when the disjunctive of the elements is used. In the Account example in figure 1-2 there can be either `BankInfo` or `Plastic` but not both.

## Mandatory Single Element

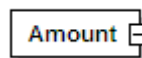


Figure 1-3: Mandatory Single Element

The rectangle indicates an element and the solid border indicates that the element is required. The absence of a number range indicates a single element (i.e. `minOcc=1` and `maxOcc=1`). The name of the element is `Amount`.

## Single Optional Element

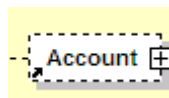
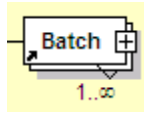


Figure 1-4: Single Optional Element

The rectangle indicates an element and the dashed border means the element is optional. The absence of a number range indicates a single element (i.e. `minOcc=0` and `maxOcc=1`). The element name is `Account`.

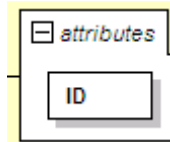
### Mandatory Multiple Elements Containing Child Elements



**Figure 1-5: Mandatory Multiple Element Containing Child Elements**

The rectangle indicates an element and the solid border indicates that the element is required. The number range `1..∞` means that `minOcc=1` and `maxOcc=unbounded`. The plus sign means complex content (i.e. at least one element or attribute child). The element name is `Batch`.

### Mandatory Attribute



**Figure 1-6: Attributes**

Each attribute is shown in a rectangle with either a solid line which means it is required or a dotted line which means it is optional. The attribute name is `ID`.

### Optional Attribute



**Figure 1-7: Attributes**

This is an example of an optional attribute `TotalCount`.

## 2 XML Implementation

### 2.1 Model Overview

The heart of the Financial Transaction Standard is the `BusinessTransaction` element. This represents a single unit of work for example a \$200 student loan payment.

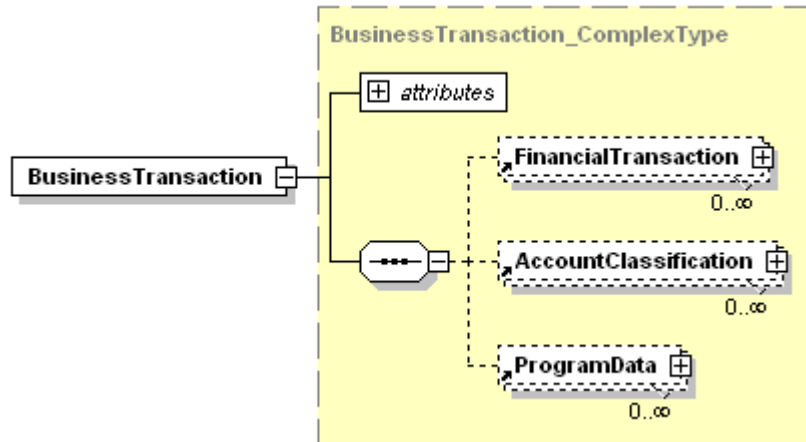


Figure 2-1: Business Transaction Standard Model

A Business Transaction can have multiple `FinancialTransaction` elements using different payment mediums. An example is an individual who may make a student loan payment with a check for \$110 and a plastic transaction of \$90.

Each Business Transaction element may also have multiple `AccountClassification` elements for storing federal accounting related information. The methods for categorizing accounting related information are stored in either Treasury Account Symbol – Business Event Type Codes (TAS-BETCs) or Classification Key Name and Values. An example of this is a student loan payment of \$200 that has \$30 categorized as interest and \$170 categorized as principle.

Business Transactions may also contain information that does not pertain to financial transactions or accounting categorization such a loan account number or an individual's Social Security Number. This information should be specified between the customer interface system and the agency and is stored in `ProgramData`.

## 2.2 Transmission

The `Transmission` element is the root element of the Financial Transaction XML instance. This element consists of attributes to describe transmission files, and it is used as a container for `Batch` elements.

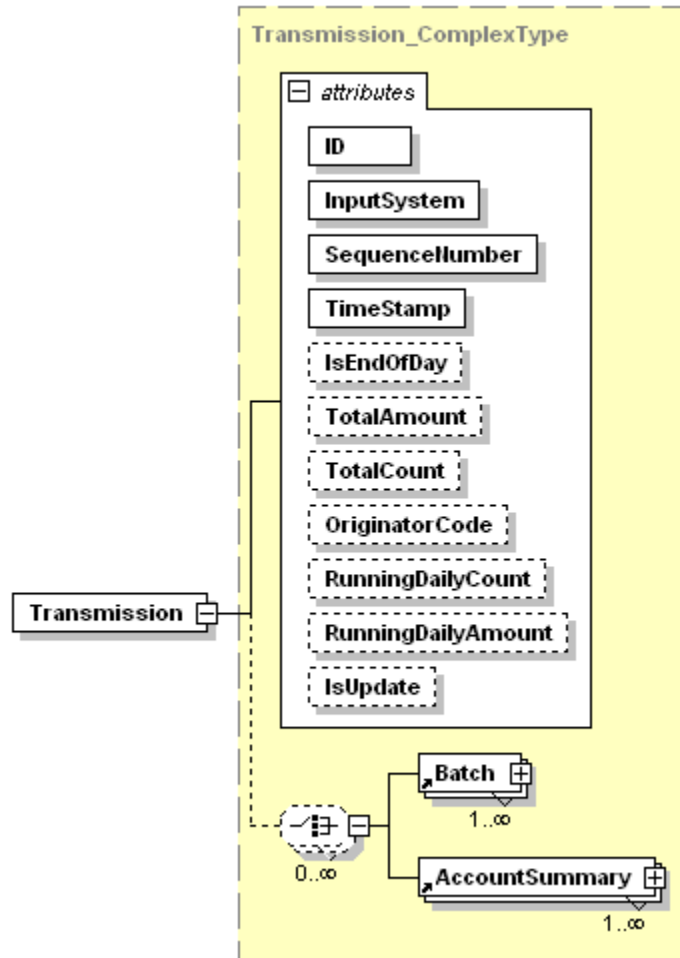


Figure 2-2: Transmission Model

### Transmission Header Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<Transmission
  xsi:schemaLocation=
    "urn:us:gov:treas:fms:BusinessTransaction:v2.1 CommonSchema.xsd"
  xmlns="urn:us:gov:treas:fms:BusinessTransaction:v2.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  InputSystem="EFTPS"

```



```

RunningDailyAmount="1234.12"
RunningDailyCount="1"
ID="aaaa"
OriginatorCode="aa"
IsEndOfDay="1"
TimeStamp="2001-12-17T09:30:47.0Z"
TotalCount="1"
IsUpdate="1"
SequenceNumber="0000"
TotalAmount="12345678912.12" >

```

## 2.3 Batch

The Batch element is a container for BusinessTransaction elements. The purpose of the Batch element is to allow the rest of the Transmission to be processed if there is an error in a single batch. It is also used to group Business Transactions for performance.

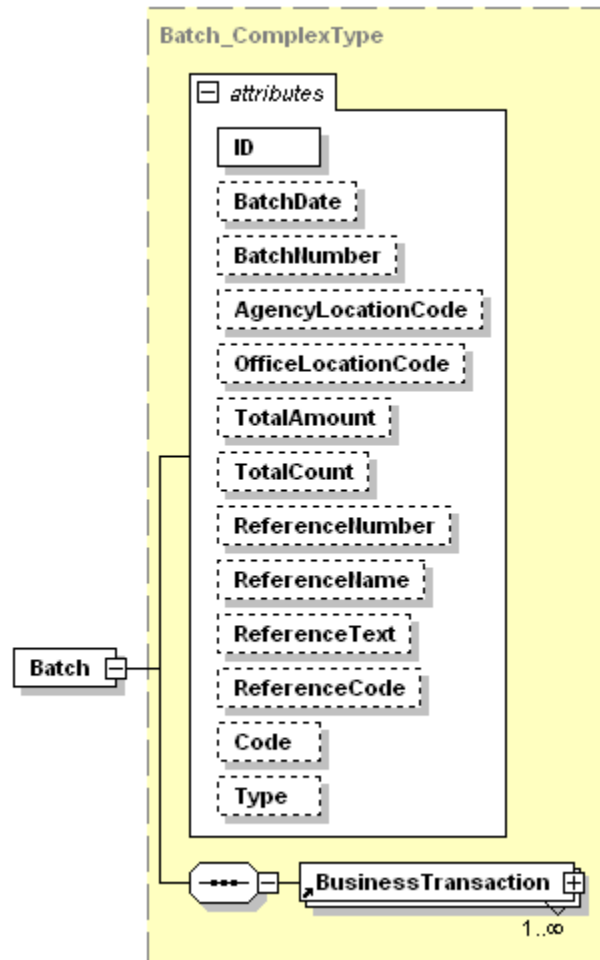


Figure 2-3: Batch Model

## 2.4 Account Summary

The AccountSummary element contains information on debit vouchers or deposit tickets. This is part of the plan to phase out the Standard Form 215 and Standard Form 5515 by Federal Finance for summary deposits.

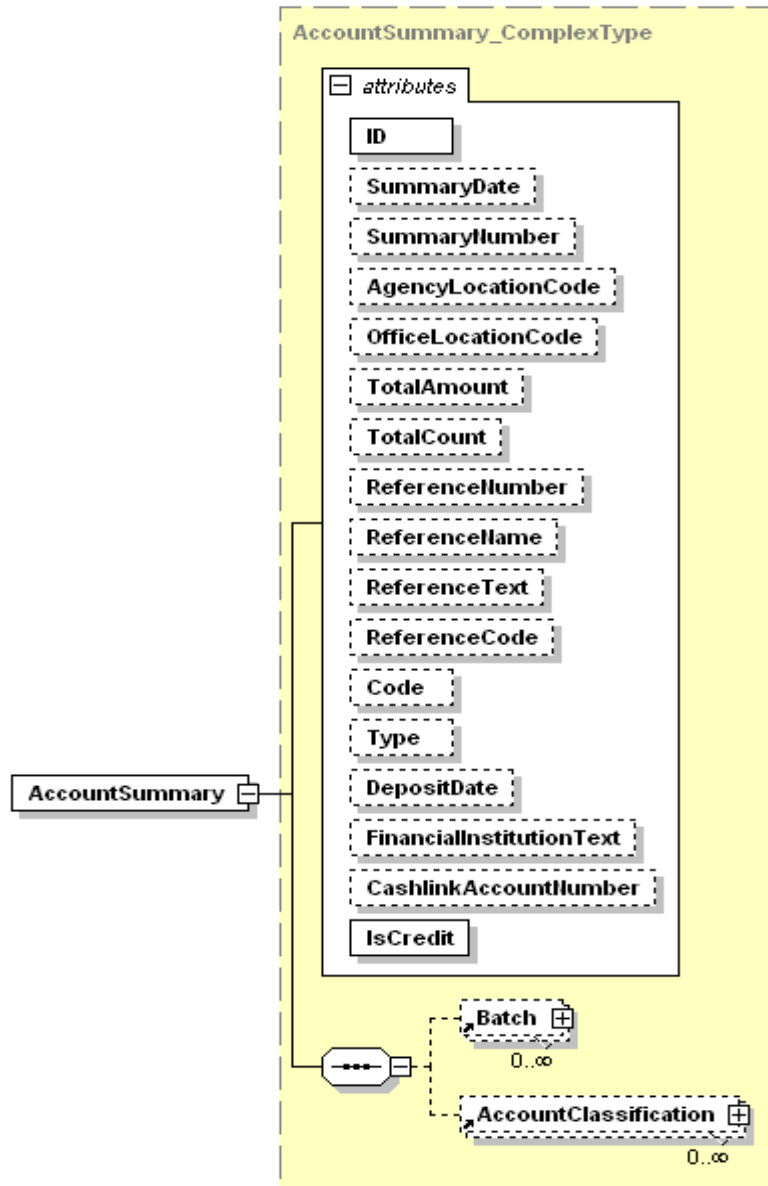


Figure 2-4: AccountSummary Model

An example mapping of the Standard Form 215 is as follows:

<b>SF 215 Element</b>	<b>Financial Transaction Schema Element</b>
(1) Deposit Number	SummaryNumber
(2) Date Presented or Mailed to Bank	SummaryDate
(3) 8-Digit or 4-Digit Agency Location Code (ALC)	AgencyLocationCode
(4) Amount	TotalAmount

An example mapping to Standard Form 5515 is as follows:

<b>SF 5515 Element</b>	<b>Financial Transaction Schema Element</b>
(1) Voucher Number	SummaryNumber
(2) Date of Debit to U.S. Treasury's Account	SummaryDate
(3) 8-Digit or 4-Digit Agency Location Code (ALC)	AgencyLocationCode
(4) Amount	TotalAmount

## **2.5 Business Transaction**

A Business Transaction represents a single unit of work that can contain many Financial Transactions, Account Classifications and Program Data.

Each `BusinessTransaction` element may contain many `FinancialTransaction`, `AccountClassification` and `ProgramData` elements within the `BusinessTransaction` element.

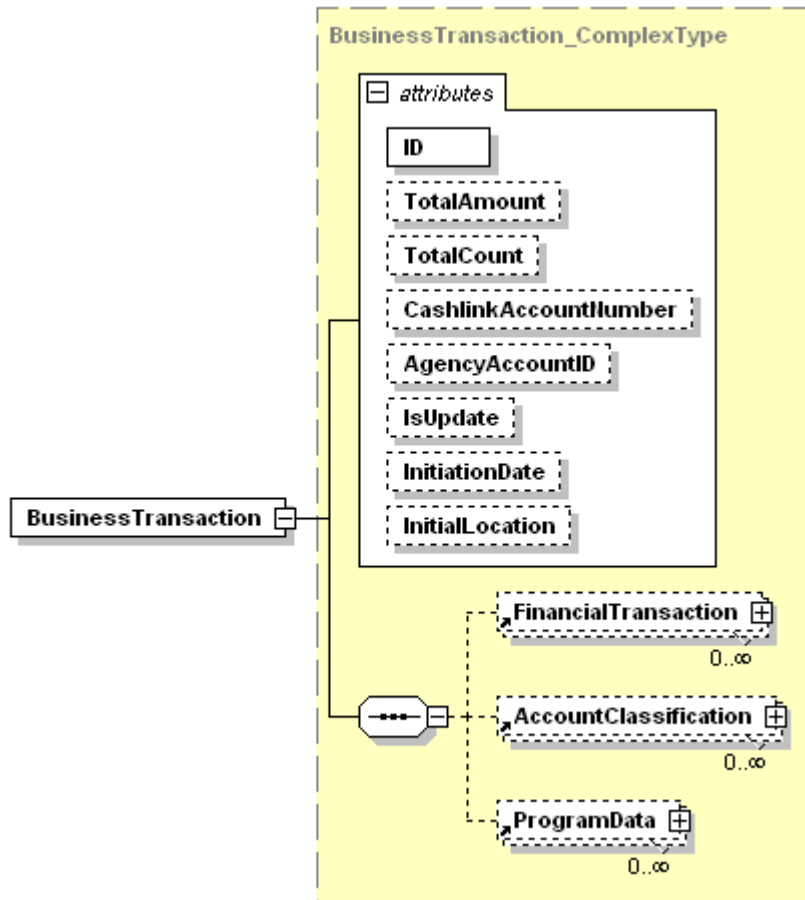
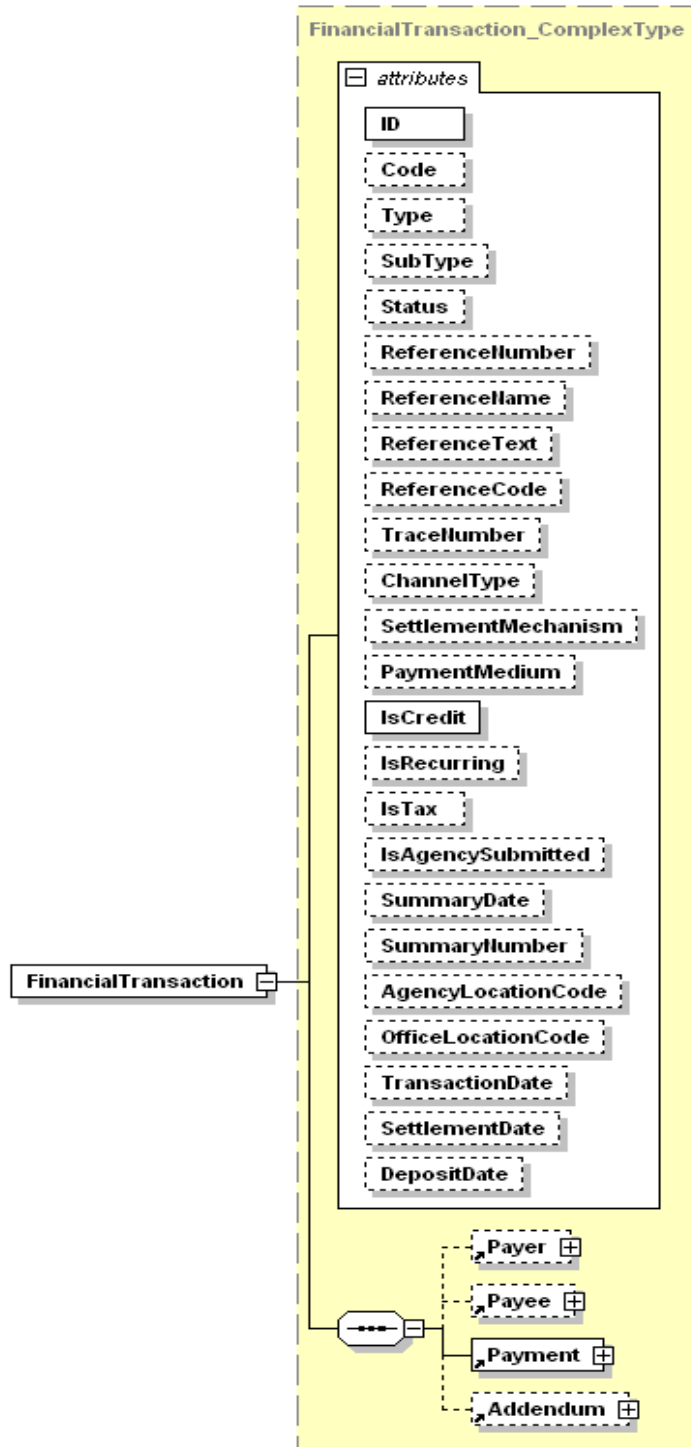


Figure 2-5: Business Transaction Model

## 2.6 Financial Transaction

The **FinancialTransaction** element is the model that the Business Transaction standard uses to represent a single financial transaction such as ACH, Cash, Check, Fedwire or Plastic.

**Figure 2-6: Financial Transaction Model**

Each Financial Transaction must have a Payment. Payer and Payee are optional elements described in section 2.6.1. The Addendum is also an optional element used for ACH financial transactions.

## 2.6.1 Payer and Payee

The `Payer` element represents the individual, organization or agency that submits payment and the `Payee` element represents the individual, organization or agency that collects payment. `Payer` and `Payee` both use `TransactionParty_ComplexType` to define their models. For federal collection systems; the `Payee` is always the `Agency`.

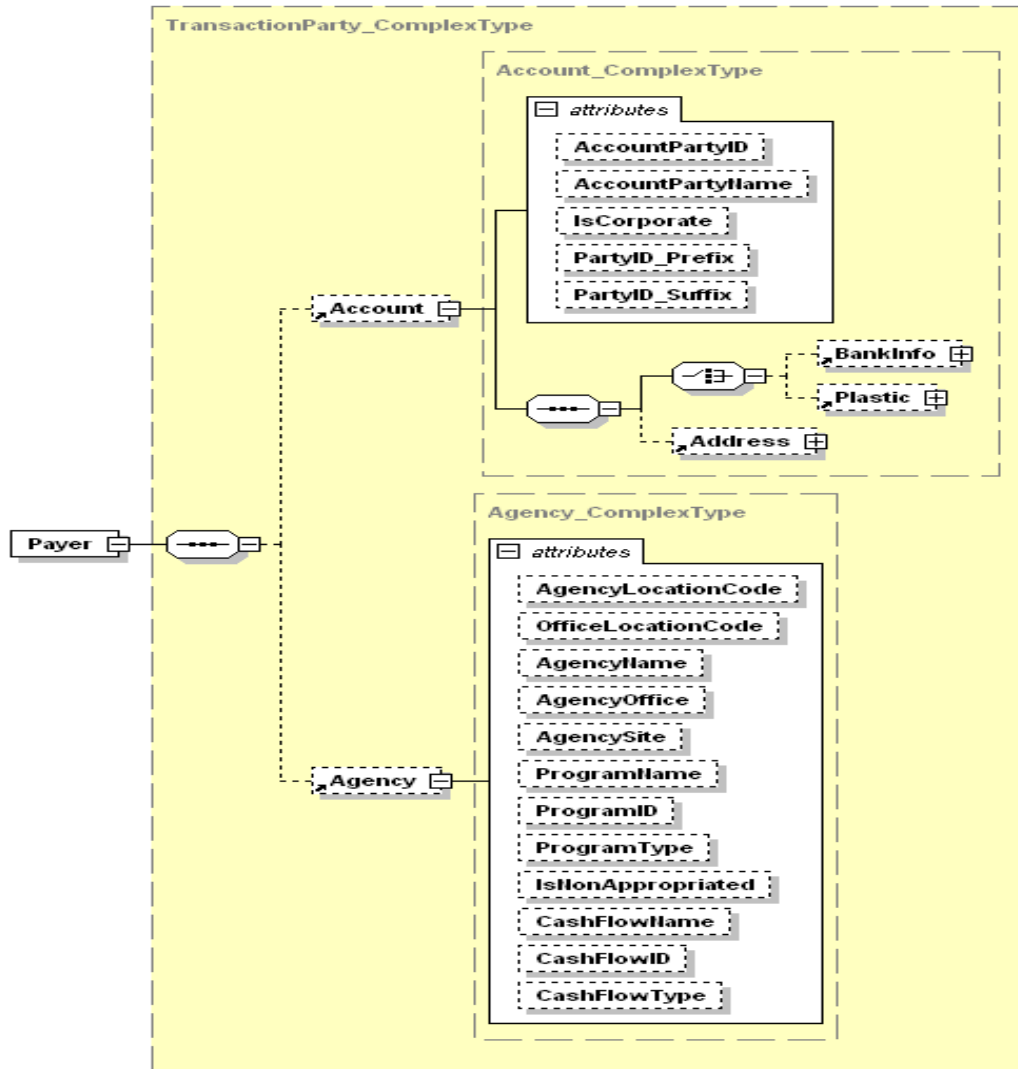


Figure 2-7: Payer and Payee Model

In a collection system the `Payer` element is always the customer who **pays** the government funds. This can be an individual or an organization which is identified in the `Account` element. The data for `Account` should always identify the person or organization that completes the Financial Transaction which could be different than the Business Transaction. If the person that pays is different than the person who owes the government funds, this should be contained in `ProgramData`. `BankInfo` and `Plastic` elements are defined in more detail in sections 2.6.2 and 2.6.3.

## 2.6.2 Bank Info

The `BankInfo` element is used to store information about a bank such as an ABA number. The element along with Account attributes and Address stores information about a check.

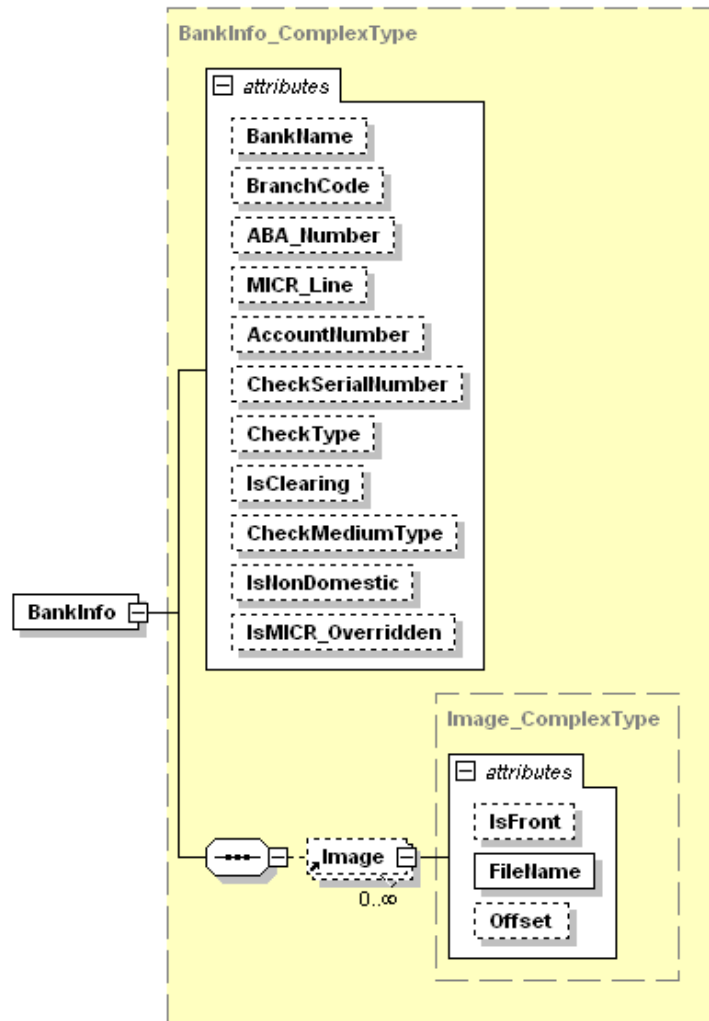


Figure 2-8: BankInfo Model

## 2.6.3 Plastic

The `Plastic` element is used to store detailed information about a credit card or a debit card. This element should only be used when the financial transaction payment medium is a plastic card transaction.

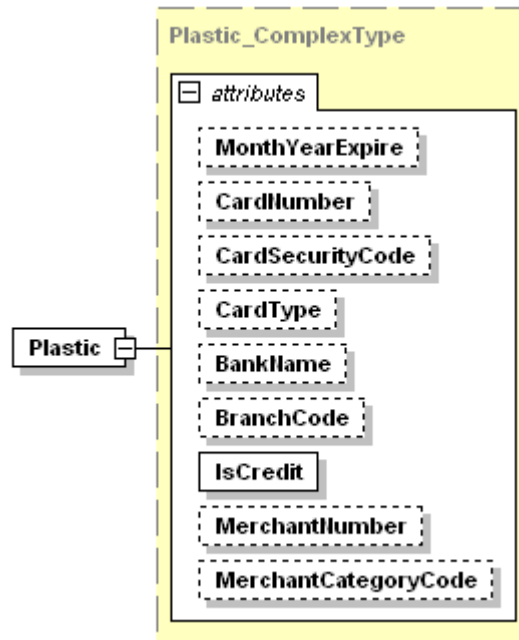


Figure 2-9: Plastic Model

## 2.6.4 Addendum

The Addendum data element is used for additional data of an ACH transaction. The Addendum element can contain many Addenda for the ACH Transaction.

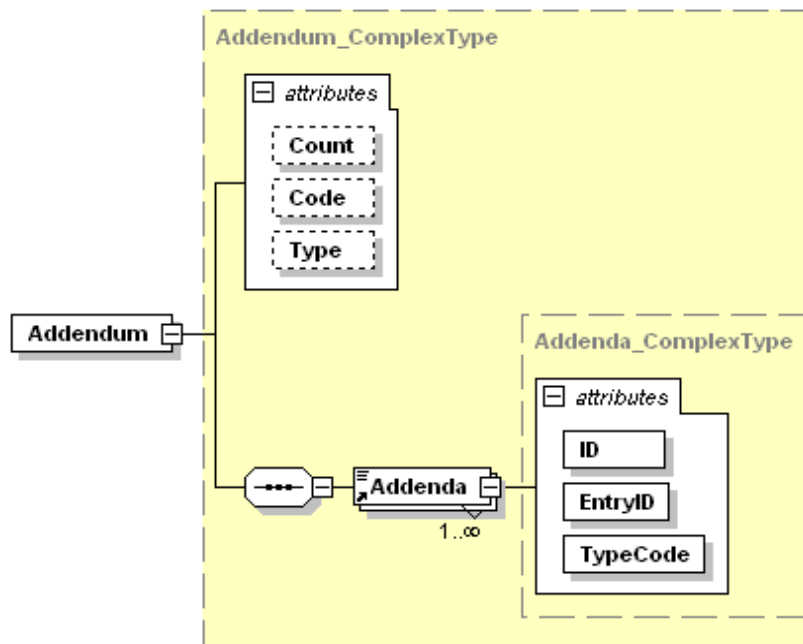


Figure 2-10: Addendum Model



## 2.7 Account Classification

The `AccountClassification` element is used to classify the Business Transaction's total amount into accounting categories for Government Wide Accounting and the US Standard General Ledger. GWA categorizes the amount either by the Treasury Account Symbol (TAS) or the Classification Key (Ckey) Name and Value pair. A Business Event Type Code (BETC) further defines a TAS. Each TAS-BETC or Ckey can have their own amounts associated with them.

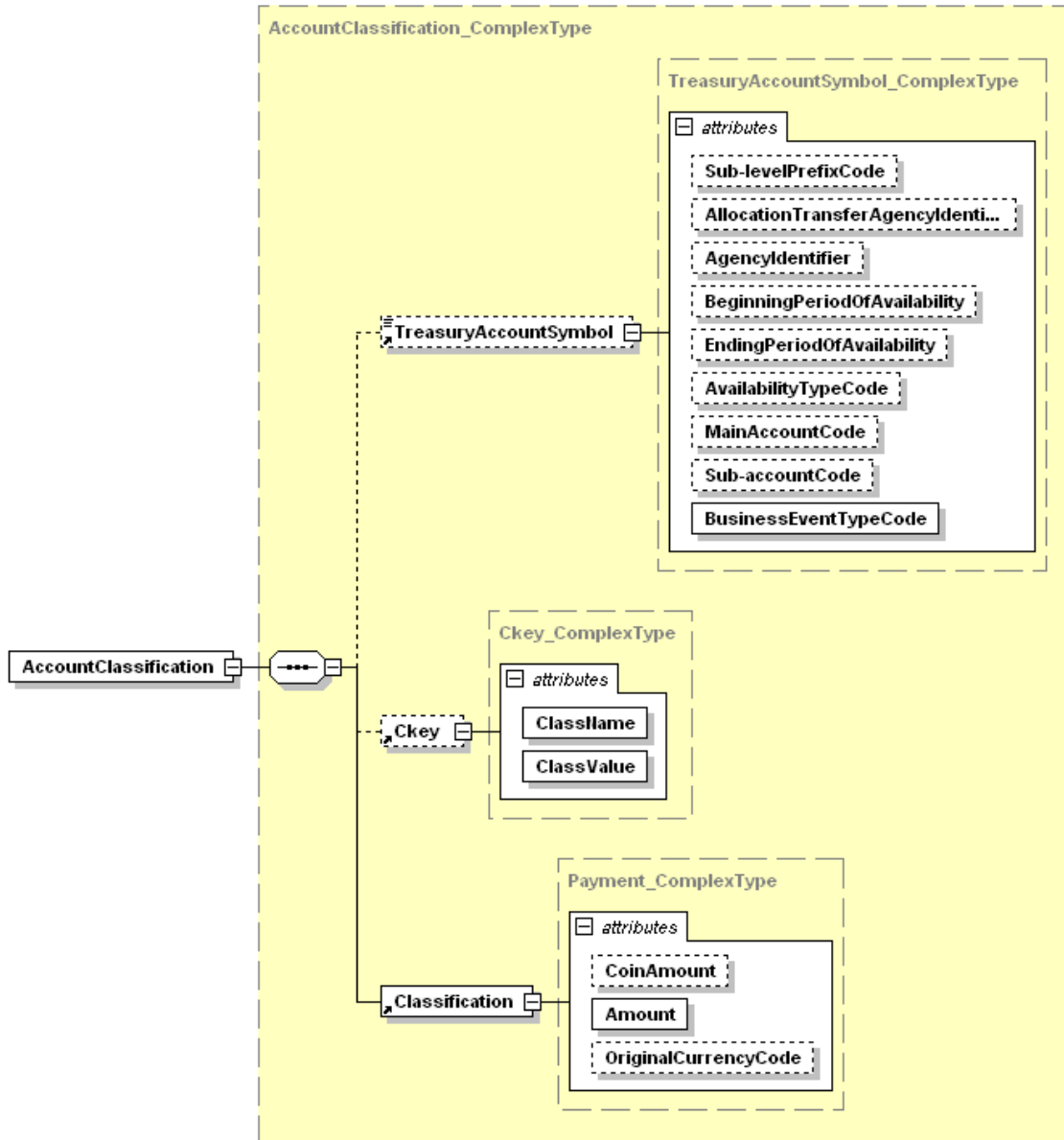


Figure 2-11: Accounting Classification Model

The `TreasuryAccountSymbol` element is an identification code assigned by Treasury, in collaboration with OMB and the owner agency, to an individual appropriation, receipt, or other fund account. All financial transactions of the Federal

government are classified by TAS for reporting to Treasury and OMB. The TAS consists of eight components that are represented in the schema by `TreasuryAccountSymbol`'s attributes.

The Ckey is a unique agency accounting code assigned to a transaction. Agencies will establish CKeys in SAM that will be used to derive the appropriate values of TAS-BETCs.

## 2.8 User Defined Data / Program Data

The `ProgramData` element is used to capture Agency data and user defined data that is not needed to process a Financial Transaction. This data is considered 'pass-through' data because Financial Management Service Systems do not store this information. This is based on the requirement that FMS is a single touch point for standard trading partners and Federal Program Agencies.

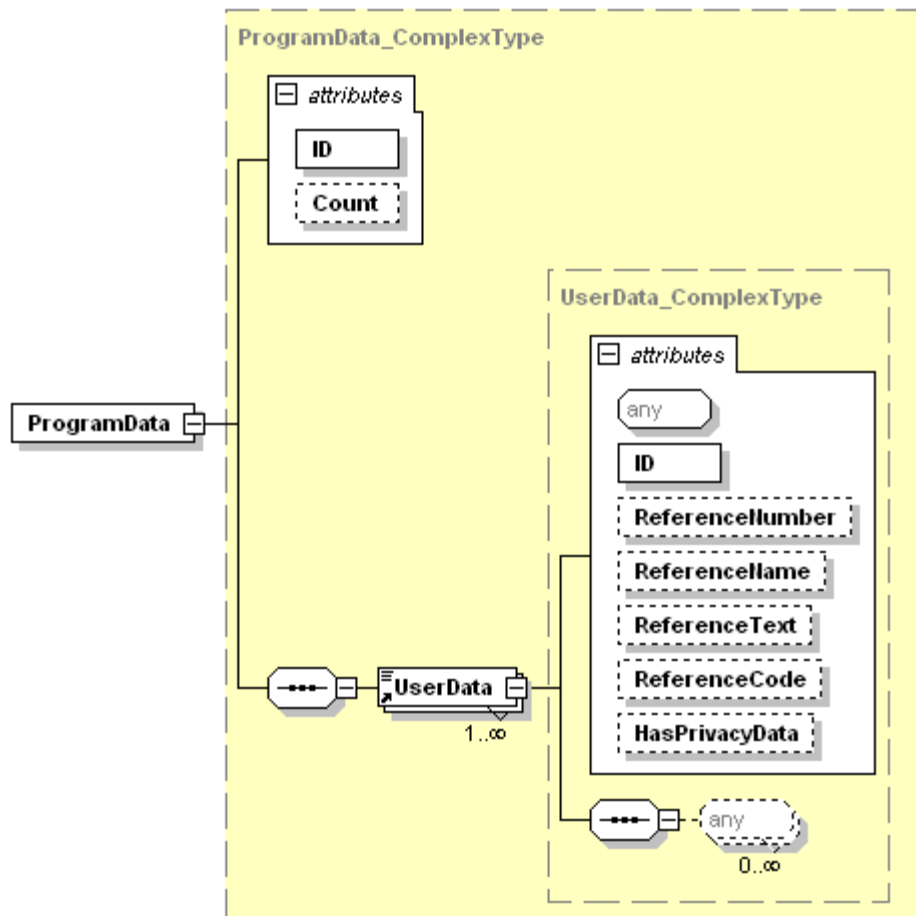


Figure 2-12: Program Data Model

## 2.9 Transmission Response Model

The Transmission Response XML format is a model designed to acknowledge the receipt of a financial transaction transmission. After receiving the file a system can respond with success, warning and errors.

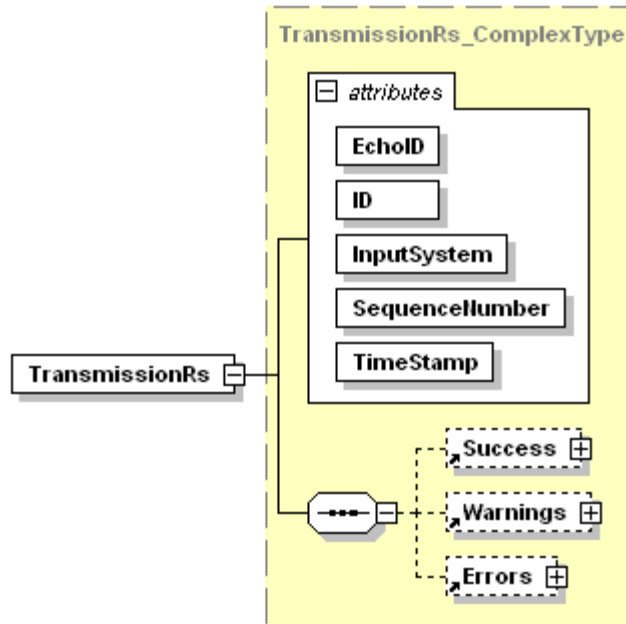


Figure 2-13: TransmissionRs Model

The root element is **TransmissionRs**. The attributes for **TransmissionRs** are **EchoID**, **ID**, **InputSystem**, **SequenceNumber** and **TimeStamp**. The attribute **EchoID** identifies the original unique identifier of the original transmission. The **InputSystem** attribute describes the initial sending partner. The **SequenceNumber** is the sequence number of the original transmission sent for that day, and the **TimeStamp** is the time stamp of the response transmission. The children for **TransmissionRs** are **Success**, **Warnings** and **Errors** which are all optional meaning it is possible to have a successful transmission with **Warnings** and **Errors**.

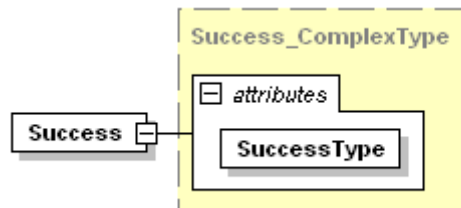


Figure 2-14: Success Model

The `Success` element has the attribute `SuccessType`. This should be set to either `Received` or `Validated`. `Received` means that the system has received the file but not validated to the schema. `Validated` means that the transmission was received and validated to the schema.

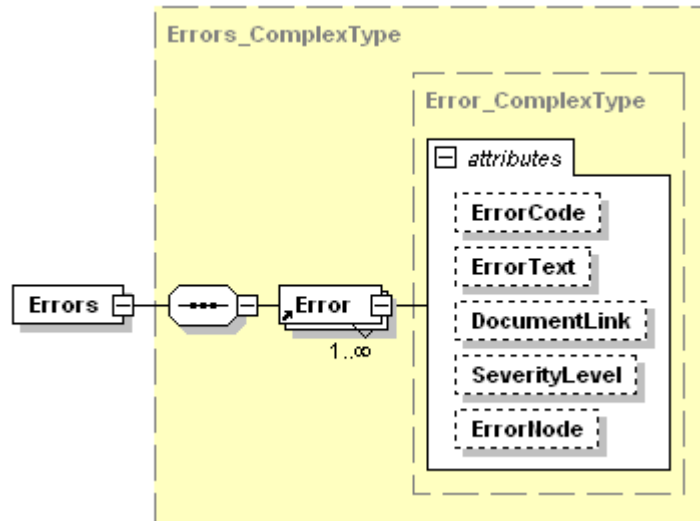


Figure 2-15: Error and Warning Model

A warning is defined as an issue in a `BusinessTransaction` that still allows for the `BusinessTransaction` to be processed. The warning is meant as instructional information for a systems administrator that may need to be looked into further. An Error message means that the `BusinessTransaction` was unable to be inserted into the database.

The `Errors` and `Warnings` elements contain the same attributes, and the difference between the uses of these elements is the severity of the issue. The attributes are `ErrorCode`, `ErrorText`, `DocumentLink`, `SeverityLevel`, and `ErrorNode`. `ErrorCode` is a predetermined controlled vocabulary between the sending systems and `ErrorText` is a short description of the error or warning. The `DocumentLink` attribute should be set to a URI that resolves to a location with further information on the Error or Warning. `SeverityLevel` should be set to an integer 1 through 9 where 1 is highest level of severity and nine is the lowest.

For the attribute `ErrorCode`, this should be a XPATH string expression for the XML node that is identified by the Warning or Error. Examples include:

*/Transmission/Batch/BusinessTransaction/@TotalAmount*  
*Batch[5]/BusinessTransaction/FinancialTransaction[6]/@Status*

For a full description of XPath specification please see the W3C specification at: <http://www.w3.org/TR/xpath>.

### 3 Financial Transactions and Payment Medium Examples

#### 3.1 Automated Clearing House (ACH)

The ACH Network is a highly reliable and efficient nationwide batch-oriented electronic funds transfer system governed by the NACHA operating rules which provide for the interbank clearing of electronic payments for participating depository financial institutions. The Federal Reserve and Electronic Payments Network act as ACH Operators, central clearing facilities through which financial institutions transmit or receive ACH entries. ([www.nacha.org](http://www.nacha.org))

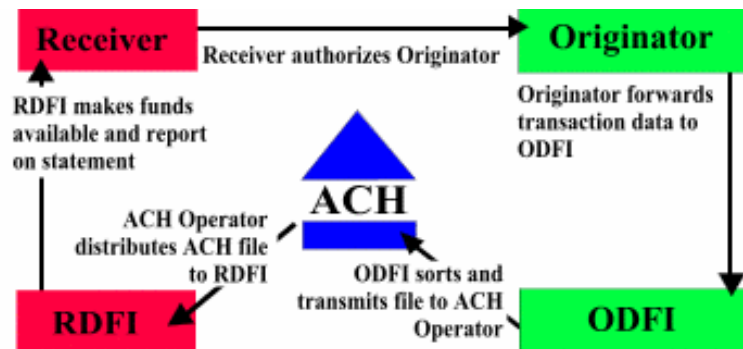


Figure 3-1: ACH Transaction ([www.nacha.org](http://www.nacha.org))

Figure 3-1 is a description of the ACH network from the NACHA Rules Handbook. The originator is the organization that initiates entries into the ACH Network. This can be an organization that offers direct payment to consumers that owe a recurring or one time consolidate. For government collections the Originator is always the Agency collecting funds. This equates to the `Payee` data element in the XML Schema.

The Originating Depository Financial Institution (ODFI) is the institution that has made an arrangement with an Originator to transmit entries into the ACH system on behalf of the Originator.

A Receiving Depository Financial Institution (RDFI) is a participating depository financial institution that receives entries directly or indirectly from its Automated Clearing House Operator for debit or credit to the accounts of its customers. For a collection system this is always the customer who makes a payment to the government. In the XML schema, this equates to the `Payer` data element

This XML example of an ACH transaction contains the following given information:

<b>NACHA Data Element</b>	<b>Value</b>
Record Type Code	DHDR
Transaction Code	4534
Receiving DFI Identification	01100020
Check Digit	1234
DFI Account Number	
Total Amount	8746.14
Identification Number	090909
Number of Addenda Records	2
Receiving Company Name/ID Number	DAVID JONES CO
Reserved	
Discretionary Data	8
Addenda Record Indicator	
Trace Number	22222
Record Type Code	
Addenda Type Code	1. 080 2. 999
Payment Related Information	1. \$3.16 interest for 20 days 2. Credit Memo Adjustment
Addenda Sequence Number	1. 489 2. 343
Entry Detail Sequence Number	1. 37487 2. 33345
ALC	01500000

XML fragment sample (only showing the FinancialTransaction element and child nodes):

```
<FinancialTransaction ID="1" IsCredit="0" Code="4534"
TraceNumber="22222">
  <Payer>
    <Account AccountPartyName="David Joe Comp">
      <BankInfo ABA_Number="01100200" AccountNumber="090909"
CheckSerialNumber="1234"/>
    </Account>
  </Payer>
  <Payee>
    <Agency AgencyLocationCode="01500000"/>
  </Payee>
  <Payment Amount="8746.14"/>
  <Addendum Count="2" Code="8">
    <Addenda ID="489" TypeCode="98" EntryID="37487">
      $3.16 interest for 20 Days
    </Addenda>
    <Addenda ID="343" TypeCode="99" EntryID="33345">
      Credit Memo Adjustment
    </Addenda>
  </Addendum>
</FinancialTransaction>
```

### 3.2 Check

Data Element	Value
Name	Jane Doe
Routing Number	255071981
Check Serial Number	12345
Account Number	111111
Date	2007-03-20
Amount	275.00
ALC	01500000

XML fragment sample (only showing the BusinessTransaction element and child nodes):

```
<BusinessTransaction ID="342134343">
  <FinancialTransaction ID="" IsCredit="0" SettlementDate="2007-
03-20" PaymentMedium="Check" TraceNumber="AL23C6S4HH1"
ChannelType="POS-OTC">
    <Payer>
```

```

        <Account AccountPartyName="Jane Doe">
          <BankInfo ABA_Number="255071981" AccountNumber="111111"
CheckSerialNumber="12345"/>
        </Account>
      </Payer>
      <Payee>
        <Agency AgencyLocationCode="01500000"/>
      </Payee>
      <Payment Amount="275.00"/>
    </FinancialTransaction>
  </BusinessTransaction>

```

### 3.3 Fedwire

The Fedwire Funds Service provides a real-time gross settlement system in which more than 9,500 participants initiate funds transfers that are immediate, final, and irrevocable when processed. Participants that maintain a reserve or clearing account with a Federal Reserve Bank may use Fedwire to send payments to, or receive payments from, other account holders directly. Participants use Fedwire to handle large-value, time-critical payments, such as payments for the settlement of interbank purchases and sales of federal funds; the purchase, sale, and financing of securities transactions; the disbursement or repayment of loans; and the settlement of real estate transactions.

Data Element	Value
1510	22/32 (Type/Subtype)
2000	300.00
3100	255071434
3320	7867
3400	Payee ABA
4200 (ALC)	01500000
6000	\$3.16 interest for 20 days
IMAD	1234
OMAD	4347

XML fragment sample (only showing the FinancialTransaction element and child nodes):

```

<FinancialTransaction ID="1234" IsCredit="0" ReferenceNumber="4347"
TraceNumber="7867">
  <Payer>
    <Account AccountPartyName="John Doe">
      <BankInfo ABA_Number="255071434"/>
    </Account>

```



```

    </Payer>
    <Payee>
      <Agency AgencyLocationCode="01500000"/>
    </Payee>
    <Payment Amount="300.00"/>
    <Addendum Count="1">
      <Addenda ID="1" EntryID="" TypeCode="98">
        $3.16 interest for 20 days
      </Addenda>
    </Addendum>
  </FinancialTransaction>

```

### 3.4 Plastic

Data Element	Value
Name	John Doe
Phone	202-555-1234
Address	2000 14 <sup>th</sup> St SW Washington, DC 20010
Credit Card Number	0000000000000000
Security Code	000
Expiration Date (Month/Year)	03/05
Type Code	Visa
ALC	01500000
Total Amount	75.00

XML fragment sample (only showing the FinancialTransaction element and child nodes):

```

<FinancialTransaction ID="1" IsCredit="0">
  <Payer>
    <Account AccountPartyName="John Doe">
      <Plastic CardNumber="0000000000000000"
CardSecurityCode="000" MonthYearExpire="0305" CardType="VI"
IsCredit="1"/>
      <Address>
        <AddressLine>2000 14th St</AddressLine>
        <City>Washington</City>
        <State>DC</State>
      </Address>
    </Account>
  </Payer>
  <Payee>
    <Agency AgencyLocationCode="01500000"/>
  </Payee>
  <Payment Amount="75.00"/>

```

```
</FinancialTransaction>
```

### 3.5 Cash

Data Element	Value
Name	John Doe
ALC	01500000
Total Amount	75.00

XML fragment sample (only showing the FinancialTransaction element and child nodes):

```
<FinancialTransaction ID="1" IsCredit="1">
  <Payer>
    <Account AccountPartyName="John Doe"/>
  </Payer>
  <Payee>
    <Agency AgencyLocationCode="01500000"/>
  </Payee>
  <Payment Amount="75.00"/>
</FinancialTransaction>
```

## 4 Examples

### 4.1 Multiple Financial Transactions

This is an example of the case where a Business Transaction is paid with two different payment mediums: Credit and ACH.

```
<BusinessTransaction ID="1">
  <FinancialTransaction ID="1" IsCredit="1">
    <Payer>
      <Account AccountPartyName="John Doe">
        <Plastic IsCredit="1" CardNumber="0000000000000000"
CardSecurityCode="000" MonthYearExpire="0305" CardType="VI" />
        <Address>
          <AddressLine>2000 14th St</AddressLine>
          <City>Washington</City>
          <State>DC</State>
        </Address>
      </Account>
    </Payer>
    <Payee>
      <Agency AgencyLocationCode="01500000"/>
    </Payee>
    <Payment Amount="75.00"/>
  </FinancialTransaction>
```

```

    <FinancialTransaction ID="2" IsCredit="1" SettlementDate="2007-
03-20" PaymentMedium="Check" TraceNumber="AL23C6S4HH1"
ChannelType="POS-OTC">
      <Payer>
        <Account AccountPartyName="OHMS TEST">
          <BankInfo ABA_Number="255071981" AccountNumber="111111"
CheckSerialNumber="12345"/>
        </Account>
      </Payer>
      <Payee>
        <Agency AgencyLocationCode="69001105"/>
      </Payee>
      <Payment Amount="275.00"/>
    </FinancialTransaction>
  </BusinessTransaction>

```

## 4.2 Multiple TAS-BETC or Ckey Name-Value

A situation can occur where one financial payment is broken down into multiple TAS-BETCs. In this case, there should be one BusinessTransaction with one FinancialTransaction and two TAS-BETCs under BusinessTransaction. The following example shows John Doe making a cash transaction of \$75.00, where the payment is categorized into multiple TAS numbers within the Department of Interior. (Taxes; Dividends and Other Interest).

```

<BusinessTransaction ID="1">
  <FinancialTransaction ID="1" IsCredit="0">
    <Payer>
      <Account AccountPartyName="John Doe"/>
    </Payer>
    <Payee>
      <Agency AgencyLocationCode="01500000"/>
    </Payee>
    <Payment Amount="75.00"/>
  </FinancialTransaction>
  <AccountClassification>
    <TreasuryAccountSymbol Sub-levelPrefixCode="00"
AllocationTransferAgencyIdentifier="000"
Sub-accountCode="000"
AvailabilityTypeCode="X"
AgencyIdentifier="000"
EndingPeriodOfAvailability="0000"
BusinessEventTypeCode="A"
MainAccountCode="0000"/>
    <Classification Amount="25.00"/>
  </AccountClassification>
  <AccountClassification>
    <TreasuryAccountSymbol Sub-levelPrefixCode="00"
BeginningPeriodOfAvailability="0000"
AllocationTransferAgencyIdentifier="000"
Sub-accountCode="000"
AvailabilityTypeCode="X"
AgencyIdentifier="000"

```

```
        EndingPeriodOfAvailability="0000"
        BusinessEventTypeCode="A"
        MainAccountCode="0000"/>
    <Classification Amount="50.00"/>
</AccountClassification>
</BusinessTransaction>
```

### 4.3 *Transmission Response Example*

```
<?xml version="1.0" encoding="UTF-8"?>
<TransmissionRs InputSystem="EFTPS" EchoID="aa" ID="bb"
    Timestamp="2001-12-17T09:30:47.0Z"
    SequenceNumber="0000"
    xsi:schemaLocation=
        "urn:us:gov:treas:fms:BusinessTransaction:v2.1TransmissionRs.xsd"
    xmlns="urn:us:gov:treas:fms:BusinessTransaction:v2.1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Success SuccessType="Received"/>
    <Warnings>
        <Warning ErrorText="This is sample warning text"
            ErrorCode="19"
            DocumentLink="http://www.fms.treas.gov/eda"
            ErrorNode="Transmission/Batch/@TotalAmount"
            SeverityLevel="1"/>
    </Warnings>
    <Errors>
        <Error ErrorText="This is sample error text"
            ErrorCode="34"
            DocumentLink="http://www.fms.treas.gov/eda"
            ErrorNode="Transmission/Batch/@BatchNumber"
            SeverityLevel="1"/>
    </Errors>
</TransmissionRs>
```