XML Implementation Guide:
General Information
Copyright 2000 Mortgage Industry Standards Maintenance Organization (MISMO).
All rights reserved

Permission to use, copy, modify, and distribute the MISMO DTD and its accompanying documentation for any purpose and without fee is hereby granted in perpetuity, provided that the above copyright notice and this paragraph appear in all copies. The copyright holders make no representation about the suitability of the DTD or documentation for any purpose.
TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION........................................................................................................ 1-1
PURPOSE OF THIS DOCUMENT .................................................................................................. 1-1
WHAT IS XML? .......................................................................................................................... 1-1
BACKGROUND .......................................................................................................................... 1-1
MISMO ...................................................................................................................................... 1-1
The MISMO Development Process .......................................................................................... 1-2
Work Groups ............................................................................................................................ 1-2
Version and Release .................................................................................................................. 1-2
Comments ................................................................................................................................ 1-3

CHAPTER 2: MISMO XML ARCHITECTURE OVERVIEW.......................................................... 2-1
THE ELECTRONIC LOAN PACKAGE .......................................................................................... 2-1
MORTGAGE PROCESS AREAS .................................................................................................. 2-2
ELEMENTS AND ATTRIBUTES: THE DTD BUILDING BLOCKS .................................................. 2-4
Element Types .......................................................................................................................... 2-4
Attribute Types ......................................................................................................................... 2-6
CONTAINMENT AND POINTING RELATIONSHIPS ................................................................. 2-10
THE PRIMARY COMPONENT ELEMENTS .................................................................................. 2-13
COMMONLY USED ELEMENTS ............................................................................................... 2-15
Party Elements ......................................................................................................................... 2-15
Request Elements ...................................................................................................................... 2-17
Response Elements ................................................................................................................... 2-23

CHAPTER 3: XML IMPLEMENTATION ISSUES....................................................................... 3-1
XML RESERVED CHARACTERS ............................................................................................... 3-1
Processing Received XML Data ................................................................................................. 3-1
Generating XML Data .............................................................................................................. 3-2
SPECIFYING THE DTD IN THE XML DATA FILE .................................................................... 3-3
ORDERING ELEMENTS AND ATTRIBUTES .......................................................................... 3-4
“Mortgage Data” Elements ..................................................................................................... 3-4
Other Elements ........................................................................................................................ 3-4
Attributes .................................................................................................................................. 3-4
WHITE SPACE IN XML DOCUMENTS .................................................................................... 3-5
HANDLING ELEMENTS WITH NO DATA ................................................................................. 3-6
Chapter 1: Introduction

Purpose of this Document

This document is designed to assist individuals who are implementing the MISMO XML standards by providing helpful information and sample files. It includes guidance on usage of the standard. Although it is not intended as an XML tutorial, certain aspects of XML that are important for the proper implementation of the standard are highlighted. The remainder of the introduction will give a brief background of the MISMO effort, followed by an overview of the data architecture, discussions of specific areas of importance and some sample data.

Separate document addendums are provided for each of the mortgage process areas like credit reporting, mortgage insurance, mortgage application, underwriting and so on that provide more detailed implementation guidance.

What is XML?

Extensible Markup Language (XML) sets up standardized rules for marking up documents that can be shared on the Internet, allowing for data exchange across platforms and applications. We believe that XML will be widely adopted by the mortgage industry as a means to format data for exchange and believe that the MISMO standards will enable that interoperability. MISMO uses the XML syntax standards Version 1.0 developed and approved by the World Wide Web Consortium (W3C).

Background

MISMO

The Mortgage Bankers Association (MBA) created the Mortgage Industry Standards Maintenance Organization, (MISMO), in October 1999. MISMO gathered individuals from a widely varied group of mortgage industry leaders with extensive business knowledge about the industry. This body of individuals is creating a data dictionary that defines the meaning of each business data element used within the mortgage industry. This body of individuals is creating a data dictionary that defines the meaning of each business data element used within the mortgage industry. The creation of a data dictionary is the key to the success of the MISMO effort. This data dictionary is the seed for generating the XML structures or any other type of structure that may be used in the future.
The result is a single common data set for the mortgage industry. The borrower, employment, property and other commonly used information have a common data definition, no matter which mortgage process is using the data.

**The MISMO Development Process**

The first and probably most important products developed by the MISMO work group are the data dictionary and data models. The data dictionary defines all data elements used by a mortgage process. The data model visually describes how all the data elements are related to each other. The data modeling effort helps to identify the data structures that are used in common by many mortgage processes and helps form linkages between the structures. The model simultaneously helps identify the structures that have common data. It also becomes the basis for organizing the XML Document Type Definition (DTD) or data schema that will be used in the future.

Process area work groups (i.e. Credit Reporting, Underwriting, etc.) identify relevant data points and containers. A representative from each work group is responsible for entering the data into a web-enabled tool that both warehouses the data dictionary and generates the XML DTDs. The tool generates the DTDs for each process area upon request. Changes made to the data points and containers are monitored by the Core Data Work Group to ensure the integrity of all the DTDs.

An XML Architecture Work Group makes key decisions about the XML standards and their implementation within the mortgage industry. The representatives of the various mortgage process area work groups meet frequently to iron out issues about the data, definitions, and organization of commonly used business data.

**Work Groups**

A list of the current MISMO Work Groups, their purpose and contacts can be found at the [http://www.mismo.org](http://www.mismo.org) web site.

**Version and Release**

This implementation guide is based on versions 1.x of the MISMO standard. The MISMO Data Dictionary, Data Models, and DTDs (commented and uncommented) can be downloaded from the [www.mismo.org](http://www.mismo.org) web site.
Comments

Comments, questions, and suggestions for improvement of this document may be submitted in writing to the Secretariat who will forward them to the appropriate MISMO Work Group.

MISMO Secretariat  (Email: mismo@disa.org)
Data Interchange Standards Association
333 John Carlyle Street
Alexandria, Virginia 22314-2852
Chapter 2: MISMO XML Architecture Overview

The Electronic Loan Package

The life of a mortgage begins when the loan application is first filled out by a borrower, and then proceeds on through qualifying, closing, servicing, bulk transfers in the secondary market and so on. The life of a typical loan package can be as long as thirty years. Today, the majority of the loan package data still persists in a paper format as it is passed from one mortgage process to another. Recently there have been pilot demonstrations of electronic closings and the advent of the Internet is prodding the mortgage industry away from its paperbound roots.

Because of the verbose, descriptive nature of its elements and attributes, the XML format becomes a natural choice for storing persistent mortgage data that can be as easily transmitted, stored, viewed and understood today, as it will be ten, twenty or thirty years from now.

To demonstrate one advantage that the verbose nature of XML has for storing loan data for long periods of time, here are three current formats for the same data element:

1. MCD*245900
2. 07A 245900
3. <PurchasePriceAmount>245900</PurchasePriceAmount>

The nature of the data is clearly understandable in the third format (XML), whereas the first two commonly used formats have meaning only to the current users of those formats.

In the previous decade the mortgage industry participants built a series of ANSI X12 data transactions for exchanging data between business partners. In the X12 data world, one mortgage business translates information from their database into an X12 transaction, transmits that transaction data to their business partner, who then translates the X12 into their own internal database format. The active “life” of an X12 transaction usually ends at this point after no more than a second or so.

However, with the advent of XML we have an information format that can persist beyond a few seconds and become a stable and easily understood platform for transacting and storing electronic mortgage data for many years into the future.

The remainder of this chapter will give a general overview of the mortgage process areas and then examine the XML building blocks – elements and attributes. Then it will show how those building blocks can be combined to construct a comprehensive data format that allows the mortgage processes to easily combine, analyze, extract and archive mortgage data using XML.
Mortgage Process Areas

MISMO has organized the business process areas across the mortgage industry under four broad categories: Origination, Servicing, Secondary, and Real Estate Services. Processes under Origination include Mortgage Application, Underwriting and Closing. Servicing includes Loan Setup and Transfer, and Investor and Default Reporting. Services include Credit Reporting, Mortgage Insurance, Flood, etc. Business process areas are still being identified and are subject to modification as the work progresses. At the center of the process areas is “Core Data”, which contains data common to multiple process areas, like Borrower and Property elements.
MISMO released Version 1.0 DTDs in the summer of the year 2000, representing the following five process areas:

- **Credit Reporting**
- **Mortgage Insurance**
- **Secondary** (Initially covers only loan pool summary and transfer-related data used in the pricing process)
- **Service Request** (Appraisal, Credit, Escrow, Flood and Title)
- **Underwriting**

As you can see, the five areas initially represented are only a small subset of the entire mortgage process.

As more mortgage process data is supported with XML DTDs and schemas, it will become possible to begin building an XML loan package from the component process area data. Since the entire package of mortgage loan data from the different process areas is stored in a single standard format, it allows for easy analysis and data mining.

The “Core Data” areas consist of any data elements that are used by more than one process area. Borrower, Employment, and Property are obvious examples of data elements used commonly by multiple process areas. One of the nice features of the MISMO DTD structure is that information common to more than one process area is always in the same naming format and structure.

Whether an XML file was generated from the Credit Reporting DTD, Underwriting DTD, Mortgage Insurance DTD or Mortgage Application DTD, you extract Employment Data in the exact same manner, traveling down the same XML component node path. This type of common structure allows software developers to create data objects that can be used across multiple process areas.
Elements and Attributes: The DTD Building Blocks

XML provides two types of structures to define data -- elements and attributes. There are no set XML rules for how these two structures are to be used, but the MISMO XML Architecture Work Group has defined guidelines for their use, so that a uniform set of XML data structures can be defined for the mortgage industry.

Elements are used as single data points or containers for sets of data points. Elements contain the actual text or data that you might see on a human-readable loan file or report. Including the full text description in the element data eliminates the need to convert a code into text when preparing a human-readable report or web page. The MISMO standard does not support the use of codes for representing data.

Attributes are a special type of data item currently used only with element containers in the MISMO DTDs. Attributes provide more information about the element such as identifiers, references to other element identifiers, and enumerated data option lists.

Element Types

There are five types of XML elements commonly used in the MISMO DTDs:

- **Container Elements** – These elements contain attributes, elements and other container elements. Container element names will always be in UPPERCASE letters. In the following XML data file segment, PURCHASECREDIT is a container for holding data related to other credits related to the loan transaction.

  ```xml
  <PURCHASECREDIT PurchaseCreditSourceType="Employer"
    PurchaseCreditType="RelocationFunds">
  <PurchaseCreditAmount>14000</PurchaseCreditAmount>
  <PurchaseCreditDescription>
    Employer Relocation Voucher
  </PurchaseCreditDescription>
  </PURCHASECREDIT>
  ```

- **String Elements** – A String element can hold any “string” of characters - word, phrase, sentence or even a paragraph depending on its purpose. Other than container elements, all element names will always be in the “UpperCamelCase” format. Here are two examples of String elements.

  ```xml
  <LastName>Jones</LastName>

  <FloodProductDescription>
    Flood Field Inspection
  </FloodProductDescription>
  ```
- **Date/Time Elements** – MISMO has adopted the ISO 8601 international standard for representing dates and times. The Date/Time element can hold a date only, or a combined date and time. A full date is formatted in a CCYY-MM-DD format as in the date shown below.

  `<BorrowerApplicationSignedDate>2000-03-10</BorrowerApplicationSignedDate>`

  If there is no “day” value for a date it will be stored in a CCYY-MM format as in the following sample credit liability account opened date.

  `<AccountOpenedDate>2000-03</AccountOpenedDate>`

  When a time is included in the element, the date and time are separated by the letter “T” as shown in the Request Date Time element shown below. The time portion is set in a HH:MM:SS format. If the “seconds” value is not available the time is set in the HH:MM format. Time values use the 24-hour format (i.e. 11:00 = 11am, 13:00 = 1pm, 14:00 = 2pm, etc.).

  `<RequestDateTime>2000-04-30T10:13:59</RequestDateTime>`

- **Numeric Elements** – Numeric elements are used for “non-money” data, like social security numbers, rates, counts or totals, etc. Even though the current MISMO DTDs cannot enforce data types, Numeric elements should only contain the numbers “0” through “9”, plus or minus signs and the decimal point.

  `<Late30DaysCount>5</Late30DaysCount>`

- **Money Elements** – This MISMO-defined element holds money values. Money elements have the same character limitations as numeric elements. Fractional dollar amounts are expressed to two decimal places. Whole dollar amounts do not have to include the “.00” decimal value and should not contain dollar signs or commas. The money element values are always assumed to be in U.S. dollars.

  **Valid Values**

  `<BaseLoanAmount>225000.00</BaseLoanAmount>`

  `<DownPaymentAmount>45000</DownPaymentAmount>`

  `<MonthlyPaymentAmount>1934.85</MonthlyPaymentAmount>`
Invalid Values

(Invalid - contains a comma)
<BaseLoanAmount>225,000.00</BaseLoanAmount>

(Invalid – contains a dollar sign)
<BaseLoanAmount>$225000</BaseLoanAmount>

Attribute Types

There are five types of XML attributes used in MISMO DTDs:

- **Enumerated Attributes** – This type of attribute has an enumerated list of allowable values defined in the MISMO DTD or Schema. When an enumerated attribute is used in an XML data file, only a value defined in the enumerated list may be used. The following sample from the Underwriting DTD shows an enumerated attribute list for the ARM payment adjustment attributes of “Payment Adjustment Calculation Type” and “Payment Adjustment Optional Type”.

```xml
<!ATTLIST ARMPAYMENTADJUSTMENT
  PaymentAdjustmentCalculationType
    (AddFixedDollarAmountToCurrentPayment | AddPercentToCurrentPaymentAmount | AddPercentToEffectivePaymentRate)    #IMPLIED
  PaymentAdjustmentOptionalType
    (Optional | Scheduled)                           #IMPLIED>
```

The following sample shows how the data defined in the DTD sample above would appear in an XML data file. The selected Enumerated Attribute values are enclosed in quotes.

```xml
<ARMPAYMENTADJUSTMENT
  PaymentAdjustmentCalculationType= "AddPercentToCurrentPaymentAmount"
  PaymentAdjustmentOptionalType= "Optional">

...other ARM payment adjustment data ...

</ARMPAYMENTADJUSTMENT>
```
• **Boolean Attributes** – This is a special type of enumerated attribute that always has the values “Y” or “N” representing a “Yes” or “No” value. The DECLARATIONS section of the MISMO DTD has a whole group of Boolean attributes as shown in the partial DTD sample below.

```xml
<!ATTLIST DECLARATIONS
  AlimonyChildSupportObligationsIndicator     (Y|N) #IMPLIED
  BankruptcyPastSevenYearsIndicator           (Y|N) #IMPLIED
  BorrowedDownPaymentIndicator                (Y|N) #IMPLIED
  ComakerEndorserOfNoteIndicator              (Y|N) #IMPLIED
  HomeownerPastThreeYearsIndicator            (Y|N) #IMPLIED
  LoanForeclosureOrJudgmentIndicator          (Y|N) #IMPLIED
  OutstandingJudgmentsIndicator               (Y|N) #IMPLIED
  PropertyForeclosurePastSevenYearsIndicator  (Y|N) #IMPLIED >
```

The following sample shows how the data defined in the DTD sample above would appear in an XML data file.

```xml
<DECLARATIONS
  AlimonyChildSupportObligationsIndicator="N"
  BankruptcyPastSevenYearsIndicator="N"
  BorrowedDownPaymentIndicator="Y"
  ComakerEndorserOfNoteIndicator="Y"
  HomeownerPastThreeYearsIndicator="Y"
  LoanForeclosureOrJudgmentIndicator ="N"
  OutstandingJudgmentsIndicator="N"
  PropertyForeclosurePastSevenYearsIndicator="N"/>
```
- **ID Attributes** – ID attributes can be used to identify a specific instance of a repeating element. They provide a mechanism for quickly locating specific data elements. Each XML ID attribute value used in an XML data file must be unique throughout the entire file. It must also be a valid XML name – that is, it begins with a letter and is composed of alphanumeric characters and the underscore without white space (spaces, tabs, carriage returns, line feeds).

One method of making sure the ID attributes are unique while maintaining a numeric count is to combine an alpha identifier with a zero padded number. The example below shows a BORROWER container with a BORROWERID attribute of “BorRec0001”. The BORROWERID could just as easily had a value of “Borrower-1”, “JonathanConsumer”, or “548603388” (his Social Security Number).

```xml
<BORROWER BORROWERID="BorRec0001">
  <UnparsedName>JONATHAN CONSUMER</UnparsedName>
  <FirstName>JONATHAN</FirstName>
  <LastName>CONSUMER</LastName>
  <SSN>548603388</SSN>
</BORROWER>
```

- **IDREF Attributes** – IDREF attributes are used to “refer to” a specific ID attribute in another container element. The ID attribute for the BORROWER record in the previous example, has a value of “BorRec0001”. The example below shows the data supplied in a credit file for a specific borrower in a CREDIT FILE VARIATION container. To identify which borrower the credit file is associated with, a BORROWERIDREF attribute in the CREDITFILEVARIATION container is given the value that “points to” the specific BORROWER container. In this example, the Credit File belongs to the borrower with an ID equal to “BorRec0001”.

```xml
<CREDITFILEVARIATION BORROWERIDREF="BorRec0001" DataRepositorySourceType="Experian">
  <UnparsedName>JOHN Q CONSUMER</UnparsedName>
  <FirstName>JOHN</FirstName>
  <MiddleName>Q</MiddleName>
  <LastName>CONSUMER</LastName>
  <SSN>442628888</SSN>
  <BorrowerBirthDate>1955-06-18</BorrowerBirthDate>
</CREDITFILEVARIATION>
```
• **IDREFS Attributes** – IDREFS attribute is identical to an IDREF attribute except that it can “refer to” more than one container with an ID attribute. Continuing the BORROWER container example used previously, the following example shows a LOAN container. Since a loan file can refer to more than one borrower, the BORROWERIDREFS attribute allows a way of “pointing to” more than one borrower.

In the example below, the LOAN container has a reference to the “BorRec0001” BORROWER container AND the “BorRec0002” BORROWER container, indicating that both borrowers are associated with the loan.

Also note that in the example below, each reference in the BORROWERIDREFS attribute is separated by a space.

```xml
<LOAN LoanPurposeType="Purchase"
       BORROWERIDREFS="BorRec0001 BorRec0002">
  <BaseLoanAmount>265000</BaseLoanAmount>
  <PurchasePriceAmount>325000</PurchasePriceAmount>
</LOAN>
```
Containment and Pointing Relationships

The diagram of a simple credit report on the next page demonstrates XML “containment” relationships. MORTGAGE DATA is the “root” container for all MISMO transactions.

- MORTGAGE DATA contains two BORROWER container elements,
- MORTGAGE DATA contains a CREDIT REPORT container element.
- CREDIT REPORT contains three MERGED LIABILITY container elements.
- CREDIT REPORT contains one MERGED PUBLIC RECORD container element.

Simple containment is not always adequate to express the more complex relationships that exist in a loan file. For example, Credit Reporting, Underwriting, Mortgage Insurance and other mortgage processes, all use data within the BORROWER container. If we were to simply structure the XML using only containment, then CREDIT REPORT would contain one or more BORROWER elements, LOAN would contain one or more BORROWER elements, and MORTGAGE INSURANCE would also need to contain one or more BORROWER elements.

When you combine all of the individual components of a loan (mortgage insurance, underwriting, credit report, etc.) into a loan “package”, then distributing borrower data repeatedly throughout the file using this simple containment structure quickly becomes repetitious. However, it is okay to have common data structures that do not have common data, repeated in multiple data element containers (such as address elements, or verification elements).

XML provides a simple mechanism that allows a container element to “point to” another container element. This mechanism is comprised of the ID attributes and IDREF attributes discussed earlier in this guide. In simplest terms:

- ID attributes are placed in the container element you want to point to.
- IDREF attributes are placed in the container where you are pointing from.
- An IDREF “refers to” or “points to” an ID.

Since multiple process areas use the borrower data within the BORROWER container, the ID and IDREF attributes allow the user access to a single source for borrower data.

The diagram on the next page shows how containment and pointing relationships are used with CREDIT REPORT data. Following the diagram is a sample XML file corresponding to the diagram.
MORTGAGE DATA

BORROWER
BORROWER ID = "Bor001"
<FIRSTNAME> JOHN
<LASTNAME> DOE
<SSN> 546 06 0091

BORROWER
BORROWER ID = "Bor002"
<FIRSTNAME> JANE
<LASTNAME> DOE
<SSN> 546 06 0092

CREDIT REPORT

MERGED LIABILITY
BORROWER ID REFS = Bor001
Creditor Name = American Express
Unpaid Balance = 1250

MERGED LIABILITY
BORROWER ID REFS = Bor001 Bor002
Creditor Name = Countrywide
Unpaid Balance = 35200

MERGED LIABILITY
BORROWER ID REFS = Bor002
Creditor Name = Sears
Unpaid Balance = 220

MERGED PUBLIC RECORD
BORROWER ID REFS = Bor002
Public Record Type = Tax Lien
Legal Obligation = 750
Credit Report data identifies liabilities, public records and other data for borrowers applying for a mortgage loan. The MISMO DTD uses BORROWERIDREFS attributes to "point to" the borrower or borrower(s) who "own" each liability and public record.

When ID attributes and IDREF attributes are added a more complete data model can be defined:

- The first borrower has a BORROWERID of "Bor001". The second borrower has a BORROWERID of "Bor002".
- The BORROWERIDREFS attribute of the "American Express" liability refers to or points to "Bor001" – John Doe.
- The BORROWERIDREFS attribute of the "Country Wide" liability refers to or points to "Bor001" and "Bor002" – both borrowers, which means that it is a joint account that both John and Jane Doe are responsible for.
- The BORROWERIDREFS attribute of the "Sears" liability refers to or points to "Bor002" – Jane Doe.
- The BORROWERIDREFS attribute of the "Tax Lien" public record refers to or points to "Bor002" – Jane Doe.

Sample XML Data

NOTE: The sample XML elements below correspond to the diagram on the previous page. The data below contains extra “white space” such as tab and space characters to improve readability, which may or may not be present in actual data.)

```xml
<?xml version="1.0"?>
<!DOCTYPE MORTGAGEDATA SYSTEM "CreditReporting1_0.DTD">

<MORTGAGEDATA>

<BORROWER BORROWERID="Bor001">
    <FirstName>JOHN</FirstName>
    <LastName>DOE</LastName>
    <SSN>546060091</SSN>
    ... other borrower data ...
</BORROWER>

<BORROWER BORROWERID="Bor002">
    <FirstName>JANE</FirstName>
    <LastName>DOE</LastName>
    <SSN>546060092</SSN>
    ... other borrower data ...
</BORROWER>

<CREDITREPORT>

<MERGEDLIABILITY BORROWERIDREFS="Bor001">
    <CreditorName>American Express</CreditorName>
    ... other liability data ...
    <UnpaidBalanceAmount>1250</UnpaidBalanceAmount>
</MERGEDLIABILITY>

<MERGEDLIABILITY BORROWERIDREFS="Bor001 Bor002">
    <CreditorName>Country Wide</CreditorName>
    ... other liability data ...
    <UnpaidBalanceAmount>35200</UnpaidBalanceAmount>
</MERGEDLIABILITY>

<MERGEDLIABILITY BORROWERIDREFS="Bor002">
    <CreditorName>Sears</CreditorName>
    ... other liability data ...
    <UnpaidBalanceAmount>220</UnpaidBalanceAmount>
</MERGEDLIABILITY>

<MERGEDPUBLICRECORD PublicRecordType="TaxLien" BORROWERIDREFS="Bor002">
    ... other public record data ...
    <LegalObligationAmount>1250</LegalObligationAmount>
</MERGEDPUBLICRECORD>

</CREDITREPORT>

</MORTGAGEDATA>
```
The Primary Component Elements

Now you’ve been introduced to the concept of an electronic loan package, the mortgage process areas, and the XML building blocks, we’ll begin to look in a little more detail at the structure of the MISMO XML DTD.

The **MORTGAGE DATA** element is always the primary or “root” container of any MISMO DTD. The MORTGAGE DATA element contains every other element and appears in each process area DTD.

The first column of the table on the next page shows the primary component container elements that can appear directly under MORTGAGE DATA. The other columns show the process area DTDs that use the primary components. This table makes it easy to see which major component areas are shared by more than one process area in the mortgage industry.

All of the existing process areas use the BORROWER container, for example. The data in this container first becomes part of the loan package when the borrower fills out a loan application. When the loan officer begins pre-qualifying the borrower, a credit report is requested from a credit bureau and this BORROWER container data is packaged as part of the credit request. Credit scores, liability data and other borrower information is returned by the credit bureau and the loan file begins to accumulate more containers.

As the mortgage process continues toward closing more container data is extracted from the loan file and passed on to the mortgage service companies to request mortgage insurance, title services, appraisals, etc. Data returned from those companies is collected in the loan file.

As loans are sold and transferred in the servicing and secondary markets it will be possible for partners to exchange the entire MISMO XML loan file. This process of building and re-using data continues over the life of the loan.

Only a small slice of the mortgage process is covered by the current DTDs. More process areas are currently under development and others are waiting for work groups to form.
<table>
<thead>
<tr>
<th>DTD ELEMENT USAGE</th>
<th>(FUTURE)</th>
<th>CREDIT REPORT</th>
<th>(FUTURE)</th>
<th>(FUTURE)</th>
<th>(FUTURE)</th>
<th>MORTG. INSURANCE</th>
<th>(FUTURE)</th>
<th>SECONDARY</th>
<th>SERVICE REQ.</th>
<th>UNDERWRITING</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM PAYMENT ADJUSTMENT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM RATE ADJUSTMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASSET</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORROWER</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORROWER RECONCILED LIABILITY</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CREDIT REPORT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CREDIT REQUEST</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CREDIT SCORE</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPENSE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLOOD REQUEST</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INCOME</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAN</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MORTGAGE INSURANCE</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MORTGAGE INSURANCE RESPONSE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PARTY</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRODUCT</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROJECT</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROPERTY (REO &amp; SUBJECT PROPERTY)</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REQUEST GROUP</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESPONSE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TITLE REQUEST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VALUATION REQUEST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VALUATIONS</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Commonly Used Elements

Party Elements

The PARTY elements act like the “Rolodex” of the loan file. This container element holds the name, address, contact, email, fax, and phone for the various “parties” to the loan transaction. By storing all of this data in a central, common location, users of the loan data have easy access to contact information for all parties related in any way to the loan.

The following is a list of some of the party types used in the MISMO Version 1.0 DTD:

- **Asset Holder** – bank, stock brokerage, life insurance company, etc.
- **Credit Bureau** – agency supplying the credit report
- **Credit Trade Reference** – the entity reporting or requesting credit data
- **Employer** – company employing the borrower
- **Fulfillment Party** – agency actually performing the mortgage service
- **Interviewer** – person interviewing the loan applicants
- **Interviewers Employer** – employer of interviewer of loan applicants
- **Lender** – company providing the funds
- **Lender’s Contact Person** – name of contact at the lending company
- **Liability Holder** – name entity that is owed money by the borrower
- **Owning Bureau** – name of credit agency that “owns” the credit data
- **Receiving Party** – the entity receiving a request or response transaction
- **Requesting Party** – entity requesting a mortgage service
- **Response Party** – (see next section for full definition)
- **Submitting Party** – entity submitting request on behalf of another party

One integral part of the PARTY structure is the CONTACT DETAIL container. This flexible structure allows for storing of the contact name, phone numbers (home, work and mobile) fax numbers, and email addresses, plus which of those is the contact method preferred by the contact.

In the MISMO XML Architecture, the Borrower is a special type of party that has more data requirements that the other parties to a loan. The Borrower Name is parsed into individual elements. Elements are provided for date of birth, marital status, alias names and so on. For these reasons, the Borrower has his/her own container – BORROWER.
The sample employer data below shows that the contact name for the SEC Institute is Andrew Smith. It lists an email address and a work phone number. Email is his preferred contact method.

**Sample Data for an Employer Party Element:**

```xml
<PARTY PARTYID="PryID0028" PartyType="Employer">
  <PartyName>SEC Institute</PartyName>
  <Address1>2924 Ponce de Leon</Address1>
  <Address2>Suite 550</Address2>
  <City>Coral Gables</City>
  <State>FL</State>
  <PostalCode>33046</PostalCode>
  <CONTACTDETAIL>
    <ContactName>Andrew Smith</ContactName>
    <CONTACTPOINT ContactPointType="Email" PreferenceIndicator="Y">
      <ContactPointValue>ASmith@SECI.com</ContactPointValue>
    </CONTACTPOINT>
    <CONTACTPOINT ContactPointType="Phone" ContactPointRoleType="Work">
      <ContactPointValue>3054469300</ContactPointValue>
    </CONTACTPOINT>
  </CONTACTDETAIL>
</PARTY>
```
Request Elements

The MISMO Architecture provides a variety of request elements that allow for a flexible request structure that can be adapted to a variety of business-to-business scenarios.

Requesting Party

The Requesting Party is the ultimate customer of the service needed for the mortgage transaction. It is usually the entity that is ultimately billed for the service.

Receiving Party

Of all the parties in a request or response transaction, the Receiving Party probably has the simplest definition – it simply identifies the party that is on the receiving end of the transaction. In a request transaction, the Receiving Party is the entity receiving the request.

In the following sample, MFC Mortgage (Requesting Party) is generating a request for flood services from Valley Flood Services (Receiving Party).

Sample REQUEST with Requesting and Receiving Parties:

```xml
<MORTGAGEDATA>
  <REQUESTGROUP>
    <REQUESTINGPARTY PARTYIDREF="PrtyID0001"/>
    <RequestDateTime>2000-11-15T10:42:16</RequestDateTime>
    <REQUEST RequestType="Flood">
      <InternalAccountIdentifier>MFC45</InternalAccountIdentifier>
      <RECEIVINGPARTY PARTYIDREF="PrtyID0002"/>
      <KEY>
        <KeyName>MFC Loan ID</KeyName>
        <KeyValue>2320323YRWC78IMS</KeyValue>
      </KEY>
    </REQUEST>
  </REQUESTGROUP>

  <PARTY PARTYID="PrtyID0001" PartyType="RequestingParty">
    <PartyName>MFC MORTGAGE INC.</PartyName>
    <Address1>333 S. ANITA #150</Address1>
    <City>ORANGE</City>
    <State>CA</State>
    <PostalCode>92868</PostalCode>
  </PARTY>

  <PARTY PARTYID="PrtyID0002" PartyType="ReceivingParty">
    <PartyName>Valley Flood Services</PartyName>
    <Address1>12355 Ventura Blvd</Address1>
    <City>Woodland Hills</City>
    <State>CA</State>
    <PostalCode>91364</PostalCode>
  </PARTY>

  ... other request elements ...

</MORTGAGEDATA>
```
Fulfillment Party

The Fulfillment Party is the name of the ultimate party processing the service request if it differs from the Receiving Party. When a service bureau has business agreements with multiple service providers, the Requesting Party may want to specify a specific service provider in the request transaction. This can be accomplished by specifying a Fulfillment Party. The sample data for Fulfillment Party builds on the example from Submitting Party.

In the sample file below, MFC Mortgage (Requesting Party) is generating a request for title services from an Internet portal, American Home Network (Receiving Party), which is a service broker. MFC Mortgage wants Western Title Services (Fulfillment Party) to do the actual title work. Western Title will bill MFC Mortgage for the service.

Sample REQUEST with Requesting, Receiving and Fulfillment Parties:

```xml
<MORTGAGEDATA>
  <REQUESTGROUP>
    <REQUESTINGPARTY PARTYIDREF="PrtyID0001"/>
    <REQUEST RequestDateTime="2000-11-15T10:46:21"/>
    <REQUEST RequestType="Title">
      <InternalAccountIdentifier>MFC45</InternalAccountIdentifier>
      <RECEIVINGPARTY PARTYIDREF="PrtyID0002"/>
      <FULFILLMENTPARTY PARTYIDREF="PrtyID0003"/>
      <KEY>
        <KeyNo>MFC Loan ID</KeyNo>
        <KeyValue>2320323YRC78IMS</KeyValue>
      </KEY>
    </REQUEST>
  </REQUESTGROUP>
  <PARTY PARTYID="PrtyID0001" PartyType="RequestingParty">
    <PartyName>MFC MORTGAGE INC.</PartyName>
    <Address1>333 S. ANITA #150</Address1>
    <City>ORANGE</City>
    <State>CA</State>
    <PostalCode>92868</PostalCode>
  </PARTY>
  <PARTY PARTYID="PrtyID0002" PartyType="ReceivingParty">
    <PartyName>American Home Network</PartyName>
    <Address1>324 Redstone Ave</Address1>
    <City>Denver</City>
    <State>CO</State>
    <PostalCode>81351</PostalCode>
  </PARTY>
  <PARTY PARTYID="PrtyID0003" PartyType="FulfillmentParty">
    <PartyName>Western Title Services</PartyName>
    <Address1>2103 Ventura</Address1>
    <City>Oxnard</City>
    <State>CA</State>
    <PostalCode>91323</PostalCode>
  </PARTY>
  ... other request elements ...
</MORTGAGEDATA>
```
Submitting Party

The Submitting Party is an individual or organization that makes a request for a service on behalf of the requesting party. For example, when a mortgage industry business portal makes a request for a service on behalf of the end user of the service, they would identify themselves in the request transaction as the Submitting Party, and identify the end user as the Requesting Party.

In the sample file below, the Internet portal - **American Home Network** (Submitting Party) is generating a title request on behalf of **MFC Mortgage** (Requesting Party). American Home Network sends the title request to **Western Title Services** (Receiving Party), who processes the request. Western Title Services will bill MFC Mortgage for the service.

**Sample REQUEST with Requesting, Submitting and Receiving Parties:**

```
<MORTGAGEDATA>
  <REQUESTGROUP>
    <REQUESTINGPARTY PARTYIDREF="PrtyID0001"/>
    <SUBMITTINGPARTY PARTYIDREF="PrtyID0002"/>
    <RequestDateTime>2000-11-15T10:42:16</RequestDateTime>
    <REQUEST RequestType="Title">
      <InternalAccountIdentifier>MFC45</InternalAccountIdentifier>
      <RECEIVINGPARTY PARTYIDREF="PrtyID0003"/>
      <KEY>
        <KeyName>MFC Loan ID</KeyName>
        <KeyValue>2320323YRWC78IMS</KeyValue>
      </KEY>
    </REQUEST>
  </REQUESTGROUP>
  <PARTY PARTYID="PrtyID0001" PartyType="RequestingParty">
    <PartyName>MFC MORTGAGE INC.</PartyName>
    <Address1>333 S. ANITA #150</Address1>
    <City>ORANGE</City>
    <State>CA</State>
    <PostalCode>92868</PostalCode>
  </PARTY>
  <PARTY PARTYID="PrtyID0002" PartyType="SubmittingParty">
    <PartyName>American Home Network</PartyName>
    <Address1>324 Redstone Ave</Address1>
    <City>Denver</City>
    <State>CO</State>
    <PostalCode>81351</PostalCode>
  </PARTY>
  <PARTY PARTYID="PrtyID0003" PartyType="ReceivingParty">
    <PartyName>Western Title Services</PartyName>
    <Address1>2103 Ventura</Address1>
    <City>Oxnard</City>
    <State>CA</State>
    <PostalCode>91323</PostalCode>
  </PARTY>
  ... other request elements ...
</MORTGAGEDATA>
```
Response Party (Respond To Party)

The request transaction also contains data elements that allow the Requesting Party to designate one or more entities to receive the response transaction – the Response Party. For example, when ordering a credit report a mortgage broker may want to select the borrower as one Response Party, the lender as another Response Party.

*NOTE: In the MISMO 1.1 release, this Response Party element is being renamed to Respond To Party to distinguish it from the Response Party used in Response transactions.*

For each Response Party, there are additional elements that allow the Requesting Party to specify a “Response Format” (XML, PCL, PDF etc.), and a “Response Method” (File, FTP, HTTP, HTTPS, Message Queue, SMTP, VAN, etc.).

Unlike the other parties listed in a request transaction, the Response Party reference is contained in the process-specific request container. For example, for a credit report request transaction, the Response Party data is inside the CREDITREQUEST container. This allows different “respond to” parties to be designated for different request types (credit, appraisal, etc.).

*NOTE: When Response Parties are not specified, a response file should be returned to the Requesting Party by default.*

In the following sample file, MFC Mortgage (Requesting Party) is requesting a credit report from Western Credit Services (Receiving Party). MFC Mortgage (1st Response Party) wants an XML copy of the file to be returned to its FTP site, and PCL copy of the credit report faxed to the borrower (2nd Response Party). Notice that in this example the Requesting Party and the first Response Party are the same company, MFC Mortgage, and have the same Party Record ID – “PrtYID0001”.

**Sample REQUEST File with Requesting, Receiving, Response Parties:**

```xml
<MORTGAGEDATA>
  <REQUESTGROUP>
    <REQUESTINGPARTY PARTYIDREF="PrtYID0001"/>
    <RequestDateTime>2000-11-15T10:42:16</RequestDateTime>
    <REQUEST RequestType="Credit">
      <InternalAccountIdentifier>MFC45</InternalAccountIdentifier>
      <RECEIVINGPARTY PARTYIDREF="PrtYID0002"/>
      <KEY>
        <KeyName>MFC Loan ID</KeyName>
        <KeyValue>2320323YrWC78IMS</KeyValue>
      </KEY>
    </REQUEST>
  </REQUESTGROUP>
  <CREDITREQUEST CreditReportRequestedActionType="Submit"
    CreditReportType="Merge"
    CreditRequestType="Joint"
    EquifaxIncludedIndicator="Y">
```
<RESPONSEPARTY ResponseFormatType="XML"
ResponseMethodType="FTP"
PARTYIDREF="PrtyID0001">
    <ResponseDestinationDescription>
    </ResponseDestinationDescription>
</RESPONSEPARTY>

<RESPONSEPARTY ResponseFormatType="PCL"
ResponseMethodType="Fax"
PARTYIDREF="PrtyID0003">
    <ResponseDestinationDescription>
        7082449024,,304
    </ResponseDestinationDescription>
</RESPONSEPARTY>

<CREDITREQUEST>
    <PARTY PARTYID="PrtyID0001" PartyType="RequestingParty">
        <PartyName>MFC MORTGAGE INC.</PartyName>
        <Address1>333 S. ANITA #150</Address1>
        <City>ORANGE</City><State>CA</State>
        <PostalCode>92868</PostalCode>
    </PARTY>
    <PARTY PARTYID="PrtyID0002" PartyType="ReceivingParty">
        <PartyName>Western Credit Services</PartyName>
        <Address1>2103 Ventura</Address1>
        <City>Oxnard</City><State>CA</State>
        <PostalCode>91323</PostalCode>
    </PARTY>
    <PARTY PARTYID="PrtyID0003" PartyType="ResponseParty">
        <PartyName>Tracey Gracey</PartyName>
    </PARTY>

    ... other request elements ...
</MORTGAGEDATA>
Key Elements

One of the features of the MISMO Request and Response transactions is the “KEY” element. This container allows the passing of one or more reference or identification “keys” in the request transaction to the service provider. The service provider will then return the same “keys” in the response transaction.

The use of the KEY element in this manner allows the requester to send reference data in the request, which will make it easy for the requester to match the response transaction to the original request. Examples of data that could be included in the KEY element could include Loan IDs, Request IDs, Mailbox Numbers, or any other data that would allow the requester to match a response to a request.

The following sample from a title request contains three KEY elements, which will be returned in the response file.

Sample REQUEST Container with Reference “Keys”:

```xml
<REQUEST RequestType="Title">
  <InternalAccountIdentifier>GM3233</InternalAccountIdentifier>
  <KEY>
    <KeyName>Loan Ref</KeyName>
    <KeyValue>01-29949230-CBR</KeyValue>
  </KEY>
  <KEY>
    <KeyName>CMID</KeyName>
    <KeyValue>9-661-1934cmid0427</KeyValue>
  </KEY>
  <KEY>
    <KeyName>Mailbox</KeyName>
    <KeyValue>BF9442</KeyValue>
  </KEY>
</REQUEST>
```
Response Elements

This section describes the various “party” elements used in the RESPONSE container of a response transaction. Even though some of the party element names are the same, their usage is different and requires explanation.

Response Party (Response From Party)

When used in the RESPONSE container, the Response Party identifies the entity that prepared the response transaction.

NOTE: In the MISMO 1.1 release, this Response Party element is being renamed to Response From Party, to distinguish it from the Response Party defined in the request.

Receiving Party

In a response transaction, the Receiving Party is the entity receiving the response.

Key Elements

The same “KEY” element(s) sent in the request transactions, are “echoed back” to the request originator in the response transaction.

The sample RESPONSE element on the next page contains the same three KEY elements that were initially sent by the customer in the request file.
Sample RESPONSE showing Response and Receiving Parties and Returned Response “Keys”:

```xml
<MORTGAGEDATA>
  <RESPONSE>
    <RESPONSEPARTY PARTYIDREF="PrtyID0001"/>
    <RECEIVINGPARTY PARTYIDREF="PrtyID0002"/>
    <KEY>
      <KeyName>Loan Ref</KeyName>
      <KeyValue>01-29949230-CBR</KeyValue>
    </KEY>
    <KEY>
      <KeyName>CMID</KeyName>
      <KeyValue>9-661-1934cmid0427</KeyValue>
    </KEY>
    <KEY>
      <KeyName>Mailbox</KeyName>
      <KeyValue>BF9442</KeyValue>
    </KEY>
    <ResponseDateTime>2000-11-08T09:42:16</ResponseDateTime>
  </RESPONSE>
  <PARTY PARTYID="PrtyID0001" PartyType="ResponseParty">
    <PartyName>Valley Flood Services</PartyName>
    <Address1>12355 Ventura Blvd</Address1>
    <City>Woodland Hills</City><State>CA</State>
    <PostalCode>91364</PostalCode>
  </PARTY>
  <PARTY PARTYID="PrtyID0002" PartyType="ReceivingParty">
    <PartyName>MFC MORTGAGE INC.</PartyName>
    <Address1>333 S. ANITA #150</Address1>
    <City>ORANGE</City><State>CA</State>
    <PostalCode>92868</PostalCode>
  </PARTY>
  ... other response elements ...
</MORTGAGEDATA>
```
Chapter 3: XML Implementation Issues

XML Reserved Characters

The topic of XML “reserved” characters is normally covered in XML tutorials, but is worth emphasizing here because ignoring these reserved characters is a frequent cause of errors in mortgage data. There are five characters that are reserved and cannot be used directly in XML element or attribute data, they must be replaced with what are called – XML Entity References. Some XML software systems will automatically do the conversion of the XML reserved characters; otherwise logic must be added to perform the conversion of the characters. The following table shows the reserved character in the first column, followed by the XML Entity Reference that it is replaced with in the second column.

<table>
<thead>
<tr>
<th>Reserved Character</th>
<th>Substitute With</th>
<th>Character Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>&amp;</td>
<td>Ampersand</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Less Than</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
<td>Greater Than</td>
</tr>
<tr>
<td>'</td>
<td>'</td>
<td>Apostrophe</td>
</tr>
<tr>
<td>&quot;</td>
<td>&quot;</td>
<td>Quote</td>
</tr>
</tbody>
</table>

Processing Received XML Data

The following examples show the XML data received in a mortgage transaction being converted into “normal” data, as it would be stored in an internal database.

Example 1:

XML Data:

```
<CreditorName>Lord &amp; Taylor</CreditorName>
```

Converts To:

```
Lord & Taylor
```

Example 2:

XML Data:

```
<PartyName>Macy&amp;apos;s</PartyName>
```

Converts To:

```
Macy’s
```
Generating XML Data

The following examples show the conversion of “normal” data into XML data using the table from the previous page.

Example 1:
Original Data:

AT&T

XML Data:

<InquiringPartyName>AT&amp;T</InquiringPartyName>

Example 2:
Original Data:

William “Billy” Goat

XML Data:

<AttorneyName>William &quot;Billy&quot; Goat</AttorneyName>
Specifying the DTD in the XML Data File

Near the beginning of XML data files, there is a DOCTYPE section that specifies the location of the DTD that is used to validate the data in the XML file. Although the DTD itself can be included within the DOCTYPE section, it is usually maintained as a separate file. The DOCTYPE declaration specifies the name and/or location of the DTD file. The following line is a sample DOCTYPE declaration line from an XML data file.

```xml
<!DOCTYPE MORTGAGEDATA SYSTEM "Underwriting1_0.DTD"> 
```

The first argument of the DOCTYPE declaration specifies the name of the “root” or main element of the data file – in this case, MORTGAGEDATA. The remainder of the line specifies the type of DTD and its name and/or location. In this case, the DTD name is shown as “Underwriting1_0.DTD”. Since a directory path or URL (Uniform Resource Locator) is not specified for the DTD, the software processing the XML data file will look for the DTD in the same directory as the XML data file.

The next two DOCTYPE sample specify a DTD in a specific directory path on a network and the following one specifies that the DTD is located at the MISMO web site.

```xml
<!DOCTYPE MORTGAGEDATA SYSTEM "C:/DTD/Underwriting1_0.DTD"> 
<!DOCTYPE MORTGAGEDATA SYSTEM "http://www.mismo.org/mismo.dtd/Underwriting1_0.DTD"> 
```

The important point of this section is that when sending XML data files to your trading partners, they may have differing requirements for where the DTD is to be found by their processing software. Some may not require a specific location for the DTD and will use the default DOCTYPE that specifies the DTD name without a path or URL. Other trading partners may ask you to specify a specific directory path or URL for locating the DTD.
**Ordering Elements and Attributes**

“Mortgage Data” Elements

The container elements directly under the root element, MORTGAGE DATA, may appear in any order. So both of the following samples of XML Underwriting data would be valid.

```
<MORTGAGEDATA>
  <ASSET>
    ...asset data goes here...
  </ASSET>
  <BORROWER>
    ...borrower data goes here...
  </BORROWER>
  <INCOME>
    ...income data goes here...
  </INCOME>
</MORTGAGEDATA>

<MORTGAGEDATA>
  <BORROWER>
    ...borrower data goes here...
  </BORROWER>
  <ASSET>
    ...asset data goes here...
  </ASSET>
  <INCOME>
    ...income data goes here...
  </INCOME>
</MORTGAGEDATA>
```

Other Elements

Data elements contained within any of the major components must appear in the order specified by the DTD or schema. In the following example, the BORROWER elements (First, Middle, Last Name, etc.) appear in the same order specified for the BORROWER element in the DTD.

```
<BORROWER BORROWERID="Bor001">
  <FirstName>JOHN</FirstName>
  <MiddleName>W</MiddleName>
  <LastName>DOE</LastName>
  <SSN>546060091</SSN>
</BORROWER>
```

Attributes

When an element contains multiple attributes, the attributes may appear in any order.
**White Space in XML Documents**

All of the XML sample data shown in this document includes formatting characters to make the data more readable. Each element is on its own line, and data within element containers is indented. These space, tab, carriage return and line feed characters are known as “white space” and create a nicely formatted output as shown below.

**Sample XML Data with “White Space”:**

```xml
<BORROWERRESIDENCE>
  <Address1>504 DAPHINE</Address1>
  <City>NEWPORT BEACH</City>
  <State>CA</State>
  <PostalCode>92663</PostalCode>
</BORROWERRESIDENCE>
```

The XML files that are exchanged between trading partners will normally be sent without any “white space” characters as shown below.

**Sample XML Data without “White Space”:**

```xml
<BORROWERRESIDENCE><Address1>504 DAPHINE</Address1><City>NEWPORT BEACH</City><State>CA</State><PostalCode>92663</PostalCode></BORROWERRESIDENCE>
```

“White space” characters are generally eliminated from production XML files for two reasons:

1. Eliminating the “white space” characters makes the XML files smaller (approximately 7% to 9%).
2. Most Internet browser software and XML editors that your trading partners use automatically format the XML data with “white space” to make it more readable.
Handling Elements with No Data

When processing XML data from your business partners, you should be aware that there are three valid ways of showing (or not showing) that an element has no data value.

The preferred method is just to not include the element name in the XML file. The following data sample shows a container record for a borrower with no middle name – and no “MiddleName” element.

Sample Borrower - No Middle Name Element:

```xml
<BORROWER>
  <FirstName>NATALIE</FirstName>
  <LastName>FERGUSON</LastName>
</BORROWER>
```

Some XML generation software will create an “empty” element, where there is a single element tag that includes the “slash” character after the element name, as shown below. This is a valid XML expression for an element with no data.

Sample Borrower – “Empty” MiddleName Element:

```xml
<BORROWER>
  <FirstName>NATALIE</FirstName>
  <MiddleName/></MiddleName>
  <LastName>FERGUSON</LastName>
</BORROWER>
```

From some systems you may also see both beginning and ending tags – without data. This is also a valid, although not ideal, XML expression.

Sample Borrower – MiddleName Tags But No Data:

```xml
<BORROWER>
  <FirstName>NATALIE</FirstName>
  <MiddleName></MiddleName>
  <LastName>FERGUSON</LastName>
</BORROWER>
```