



**Implementation of Custom Records Extension Component (CREC) into
MITS Data Transfer Standards**

A CustomRecords Extension Component (CREC) has been added to all MITS Data Transfer Standards (DTS) to allow firms to define and exchange custom data that is not currently defined within a particular schema through the use of name/value pairs. This approach allows for the use of custom data, but confines it to a predefined MITS standard structure. Firms utilizing MITS DTS can now meet their business needs to accommodate unique and previously undefined elements in their data exchanges. Since the CustomRecords element is optional, firms that do not wish to utilize this are free to exclude it from their data exchanges.

The annual review and update of MITS Core Data is currently underway by the MITS Technical Committee. To allow immediate implementation of the CREC by MITS users a 2.1 BETA version of all 2.0 data transfer standards has been created. They include:

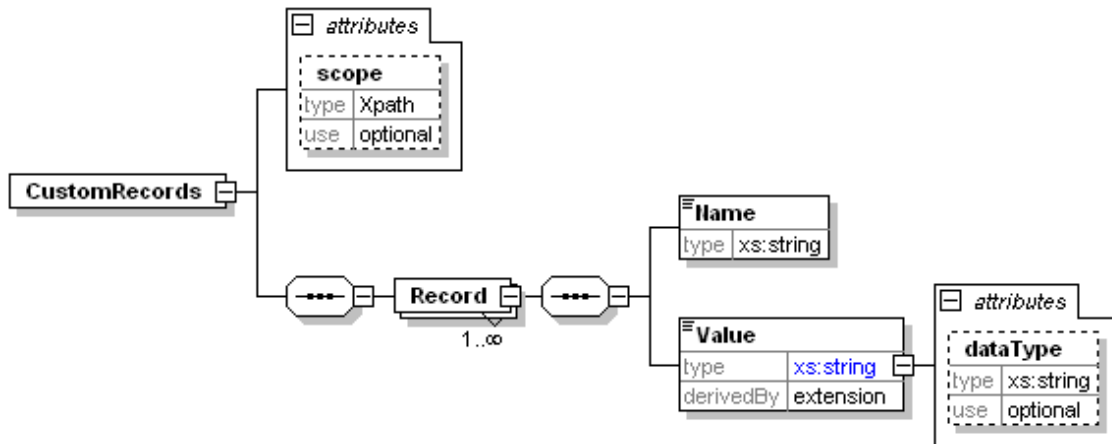
- Resident Transactions 2.1 BETA
- Resident Screening 2.1 BETA
- Property-Marketing 2.1 BETA

When the Core update is complete and MITS data transfer standards are updated to version 3.0, we will update all 2.1 BETA standards to version 3.0 thus incorporating the Custom Extension. The following provides a more detailed technical summary of the CREC.

CustomRecords Extension Component:

The CustomRecords Extension Component (CREC) will become the last child of the schema's root element. By using the Xpath attribute, you can relate new elements and their values to any of the existing elements or nodes with the schema, including the root element if desired.

Schema View of CREC:



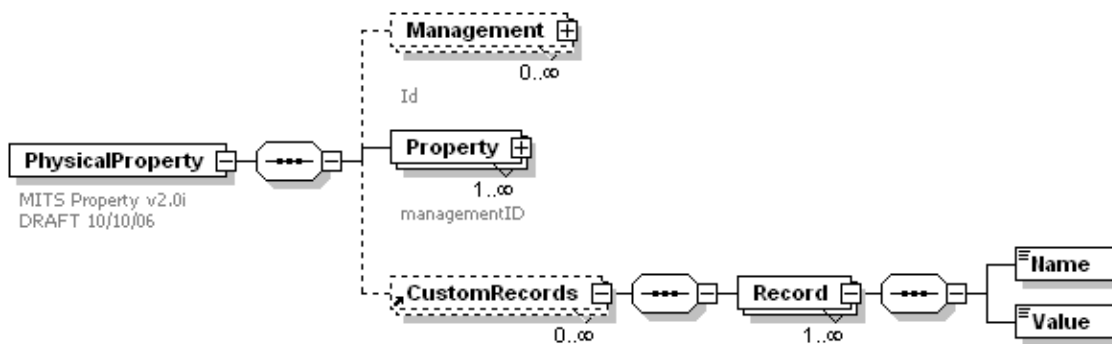
Text View of the CREC:

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="CustomRecords">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Record" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Name" type="xs:string"/>
              <xs:element name="Value">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute name="dataType" type="xs:string" use="optional"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="scope" type="Xpath" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="Xpath">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>

```

Placement Example:



Sample Code:

```
<!-- Example of adding a TertiaryID to PropertyID node with a value of 1234 -->
```

```
<CustomRecords  
scope="/PhysicalProperty/Property/PropertyID/MITS:Identification[MITS:PrimaryI  
D='123']">  
  <Record>  
    <Name>TertiaryID</Name>  
    <Value=>1234</Value>  
  </Record>  
</CustomRecords>
```

```
<!-- Example of adding a GlassCeiling to Floorplan/Amenities node with a value  
of 1 (to match other boolean values) -->
```

```
<CustomRecords  
scope="/PhysicalProperty/Property/PropertyID/MITS:Identification[MITS:PrimaryI  
D='123']/../Floorplan[@id='456']/Amenities">  
  <Record>  
    <Name>GlassCeiling</Name>  
    <Value dataType=boolean>1</Value>  
  </Record>  
</CustomRecords>
```