

---

## 2 Structure Models

To specify data models, a distinction is made between a (data) type and a (data) component.

- A type specifies the meaning and structure of some information.
  - A type has a name.
  - A type can consist of a combination of data components and is referred to as a composite data type.
  - Base data types are: strings, integers and floats.
  - If not stated otherwise, the base type string must be assumed as a base representation of the type.
- A component specifies an instance (a value) of a type.
  - A component has a name.
  - One can refer to the value of the component by using the name of the component.
  - If the type of a component is not mentioned, a corresponding type **MUST** be assumed by appending the postfix “Type” to the name of the container.

---

## 2.1 Type ContainerType

### Type ContainerType

The ContainerType is a composite data type that can contain an arbitrary number of components. It is intended to provide arbitrary artifacts regardless of underlying transport format restrictions. This type is used to allow the transport of data not known at the design time of the schema.

∇ ContainerType MAY contain the components AttRefUri, Id, IdRef, and MimeType, each can occur only 0 or 1 time. Δ [DSS-3.1-1]

∇ At least one of Id or IdRef MUST be present Δ [DSS-3.1-2]

---

### Type AttRefUriType

The AttRefUriType defines an URI according to RFC @@@ that references an attachment within a transport message. where the component is part of.

---

#### 1. Type IdType

The IdType defines a unique identification for an instance of this type.

---

#### 2. Type IdRefType

The IdRefType defines a reference to another instance of this type by providing a value that corresponds to the value of the Id component within the other instance.

---

#### 3. Type MimeType

The MimeType specifies the type of information, according to RFC @@@, that is referenced by means of the IdRef and/or AttRefUri (???? or should it only be AttRefUri ????).

#### **Non-normative Comment:**

The (Id, IdRef) pair allows to include artifacts just once even when it is used in different place in the transport message.

Typical use case are where the payload may contain binary parts that cannot be represented in the used transport format directly (e.g. null bytes), or it represents content that could interfere with the transport syntax, e.g. a XML document within a XML transport envelope.

#### 4.15.1

Format specific implementations including mappings to the generic terms are defined in 6.3 and 6.6 respectively.

---

## 2.5

||  
||

---

|| 4.

||

---

|| 5.

||

2.1

---

## 2.6 6.36.6

---

### 2.7 Type `InternationalStringType`

|| **Type `InternationalStringType`**

|| The `InternationalStringType` is a composite type and specifies a human-readable string in a specified language.

---

### **Type `HumanReadableStringType`**

|| The `HumanReadableStringType` is a string in a certain human-readable language.

---

|| **6. Type `LanguageType`**

|| The `LanguageType` specifies the language of the human-readable string according to RFC @@@.

Format specific implementations including mappings to the generic terms are defined in 6.4 and 6.7 respectively.

---

## 2.8 Type KeyInfoType

### Type KeyInfoType

The KeyInfoType type is a composite type and specifies a key pair or certificate and the owner or issuer.

∇ Instances of the KeyInfoType MAY contain one of the following components:

X509Digest,  
X509SubjectName,  
X509SKI,  
X509Certificate,  
KeyName. Δ [DSS-3.4-1]

∇ Instances of the X509DigestType MUST contain exactly one component

Algorithm. Δ [DSS-3.4-2]

---

### 7. Type X509DigestType

The X509DigestType specifies the hash-value of a X509 certificate to identify the latter, according to RFC @@@.

---

### 8. Type AlgorithmType

The AlgorithmType specifies the digest algorithm used to calculate the value of the X509Digest, according to RFC @@@.

---

### 9. Type X509SubjectNameType

The X509SubjectNameType specifies the subject name components according to RFC @@@, as converted to a string.

---

### 10. Type X509SKIType

The X509SKIType specifies a subject key identifier related to a X509 certificate according to RFC @@@.

---

### 11. Type X509CertificateType

The X509CertificateType specifies the base64 encoding of the DER-encoded certificate according to RFC @@@.

---

### 12. Type KeyNameType

The KeyNameType specifies a string identifying a given Key within the realm of the server that contains the key. There is no restriction on the format of the KeyNameType or the key material it identifies (as long as it can be represented by a base string).

Format specific implementations including mappings to the generic terms are defined in 6.5 and 6.8 respectively.