Digital Data Exchange (DDEX) Standard

Data Dictionary Standard

Document Name:	DDEX-DICT-10-2006
Date of Declaration:	17th October 2006

© DDEX

Evaluation Licence

Subject to your compliance with the terms and conditions of this Agreement, DDEX grants you a limited, nonexclusive, non-transferable, non-sublicenseable, royalty-free licence solely to reproduce, distribute within your organisation, and use the DDEX standard specifications ("DDEX Standards") solely for the purpose of your internal evaluation. You may not make any commercial use of the DDEX Standards under this agreement. No other licences are granted under this agreement.

No representations or warranties (either express or implied) are made or offered by DDEX with regard to the DDEX Standards. In particular, but without limitation, no representations or warranties are made in relation to:

a) the suitability or fitness of the standards for any particular purpose;
b) the merchantability of the standards;
c) the accuracy, completeness, relevance or validity of the standards; or
d) the non-infringement of any third party intellectual property rights related to the DDEX Standards.

Accordingly, DDEX and/or its members shall not be liable for any direct, indirect, special, consequential or punitive loss or damages howsoever arising out of or in connection with the use of the standards. IN THE EVENT THAT ANY COURT OF COMPETENT JURISDICTION RENDERS JUDGEMENT AGAINST DDEX AND/OR ITS MEMBERS NOTWITHSTANDING THE ABOVE LIMITATION, THE AGGREGATE LIABILITY TO YOU IN CONNECTION WITH THIS AGREEMENT SHALL IN NO EVENT EXCEED THE AMOUNT OF ONE HUNDRED U.S. DOLLARS (US$ 100.00).

Users of the DDEX Standards are cautioned that it is subject to revision. Users are recommended to use the latest versions, which are available at www.ddex.net. The use of outdated versions of the standards is not recommended but may be required by agreement between implementers in particular cases.

Contents

1 Introduction

The DDEX Data Dictionary Standard was developed by the member organisations of the Digital Data Exchange, LLC (DDEX).

DDEX's purpose in creating the DDEX Data Dictionary was to provide a centrally administered system to provide the semantic meaning of all the terms used within other DDEX Standards, particularly those specifying message formats for the exchange of information between record companies, rights societies, music publishers, digital music service providers and other interested parties on an international basis.

2 Scope

The DDEX Data Dictionary provides the canonical definition of *all* Entities to be used in *all* DDEX Message Standards and other DDEX Standards. DDEX Message Standards are usually expressed in the XML language. As the DDEX Data Dictionary will grow over time, this Standard also specifies how the DDEX Data Dictionary is created and maintained (including how new Entities are added to the DDEX Data Dictionary) by specifying the requirements for a DDEX Data Dictionary Registration Authority.

The purpose of the DDEX Data Dictionary is threefold:

- To support the consistent and efficient development, use and understanding of DDEX Messages Standards and other DDEX Standards;
- To prevent ambiguity in the meaning of terms used in the DDEX Standards; and
- To enable the effective mapping of Entities and DDEX Standards to other metadata standards, where necessary.

3 Organisation of this Standard

Clause 4 provides lists of normative and non-normative references. Clause 5 defines essential terms used within this Standard. Clauses 6, 7 and 8 provide the specification of the DDEX Data Dictionary, its related Namespaces and Baseline Schema. Clause 9 then provides the rules for maintaining the DDEX Data Dictionary.

Finally, the normative Annex A includes the Core Entities of the Dictionary including the Relators.

4 References

4.1 Normative References

The following normative documents contain provisions, which through reference in this text constitute provisions of this Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

- ISO/IEC 21000-6:2004: Information technology – Multimedia framework (MPEG-21) – Rights Data Dictionary (RDD)
- ISO 639 (all parts): Codes for the representation of names of languages
- ISO 3166 (all parts): Codes for the representation of names of countries and their subdivisions
- ISO 4217:2001: Codes for the representation of currencies and funds
- ISO 6801:2004: Data Elements and interchange formats – Information exchange – Representation of dates and times
- IETF RFC 2396:1998: Uniform Resource Identifiers (URI): Generic Syntax
- XML Schema Part 1: Structures and Part 2: Data types, W3C Recommendation, 2nd May 2001

- Canonical XML Version 1.0, W3C Recommendation, 15th March 2001

4.2 Informative References

The following documents contain information that may be helpful for the understanding of this Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

- Indecs. The <indecs> Metadata Framework: Principles, model and data dictionary. 2nd Version. 2000. Available from http://www.indecs.org/.

5 Terms and Definitions

For the purposes of this Standard, the following terms should be read as having the meanings specified here.

As this is a Standard for a Data Dictionary, many Entities are defined in the body of the Standard and the Data Dictionary itself. The Entities defined here provide the necessary framework within which this Standard has been developed; they are also included in the Data Dictionary itself.

5.1 DDEX Baseline Schema

The DDEX Baseline Schema comprises one or more XML Schema files containing the Entities contained in the DDEX Data Dictionary and their XML syntactic relationships for use in DDEX Messaging Standards or other DDEX Standards.

5.2 DDEX Data Dictionary

The DDEX Data Dictionary provides the canonical definition of all Entities to be used in all DDEX Messaging Standards and other DDEX Standards.

5.3 DDEX Messaging Standards

A DDEX Messaging Standard defines a set of XML Messages using the Entities defined in the DDEX Data Dictionary and as expressed in the DDEX Baseline Schema (see also Section 8).

5.4 DDEX

The Digital Data Exchange, LLC (DDEX) standards setting organisation.

5.5 DDEX Standards

The DDEX set of integrated Standards, including identifiers, messages and tools, developed by DDEX.

6 DDEX Data Dictionary Structure

The DDEX Data Dictionary is the managed set of canonical definitions of *all* Entities to be used in *all* DDEX Messaging Standards and other DDEX Standards.

6.1 Dictionary Entities

An Entity is a term in the DDEX Data Dictionary. Entity is the parent of all other terms. This introduces the most basic concept of the DDEX Data Dictionary that “an entity must be denoted by a name before it can be recognized in an information system”. An Entity may be physical or conceptual, “real” or

imaginary – it does not need to "exist" in any other sense than having a name. Entity corresponds to fundamental terms such as "Element" or "Thing" in other models and ontologies.

The Core Entities of the Dictionary are defined in Annex A of this Standard. Further Entities may be added at any time following the procedures laid down in this Standard.

Semantically, all Entities belong to one of the following basic types:

- Simple Entities; and
- Composites.

Entities are related to one another, hierarchically and in other ways, so that the DDEX Data Dictionary is a structured "ontology". Entities depend on other Entities for the inheritance and specialisation of meaning. The relationships between Entities are defined by the use of Relators which are themselves defined as Entities within the DDEX Data Dictionary.

An Entity is included in the DDEX Data Dictionary either because it is used in a DDEX Messaging Standard or another DDEX Standard, or because it is a Structural Entity.

With the exception of allowed values (that is, Entities which feature as enumerations of XML simple types in the DDEX Baseline Schema), there is no limit on the length of the Headword of each Entity.

6.1.1 CORE ENTITIES

A Core Entity is an Entity defined explicitly in Annex A of this Standard. Core Entities comprise a set of Structural Entities including Relators which establish a basic ontological structure onto which all other Entities can be added, and constitute the foundation of the DDEX Data Dictionary: all Entities which are added to the Dictionary inherit some or all of their meaning from one or more Core Entities.

6.1.2 OTHER ENTITIES

In addition to the Core Entities defined in Annex A, the Data Dictionary contains all other Entities which are used in DDEX Messaging Standards or defined for use in other DDEX Standards; and the Structural Entities required linking them into the ontological structure of the Data Dictionary. Entities may be added to the Data Dictionary by the method laid down in Clause 9 of this Standard.

The only distinction between a Core Entity and another Entity is the method by which it is added to the Dictionary.

6.2 Simple Entities and Composites

Structurally, all Entities belong to one of two types: Simple Entities or Composites. This distinction is important because of the Namespaces in which they may be used.

6.2.1 SIMPLE ENTITIES

A Simple Entity is an Entity in the DDEX Data Dictionary which has a simple value or is itself a simple value when used in a DDEX Message Standard or other DDEX Standard. For example: PartyId, MediaType, FileSize, Broadcasting, IsPartOf or MusicalWorkContainingSamples. Simple Entities are the basic building blocks of the DDEX Messaging Standards.

All Simple Entities in the DDEX Data Dictionary are found in the DDEX Namespace.

6.2.2 COMPOSITES

A Composite is a group of two or more Entities combined for any purpose. For example: a DDEX Message Standard, a set of Reference Descriptive Metadata, a group of Entities within a Message Standard such as a Title, or a set of Allowed Values for a Type such as MessageType.

A Composite is a collection or "bag" of Entities which have a relationship with one another. Composites may contain other Composites to any level of granularity. The Composite structure enables the DDEX Data Dictionary to contain all the details of a DDEX Message Standard or other DDEX Standard. Composites in the DDEX Data Dictionary may be found in any Namespace (except the DDEX Namespace).

6.3 Entity Relationships

An Entity Relationship is a relationship in which one Entity is associated with another Entity within the DDEX Data Dictionary using a Relator. The structure of the Data Dictionary is a set of such Entity Relationships. For example, the parent-child Relationship between Identifier and PartyId is established by the following triple:

PartyId IsSubClassOf Identifier

6.3.1 RELATORS

A Relator is an Entity which links two other Entities within an Entity Relationship (for example IsSubClassOf, IsPartOf, IsA, HasMaxCardinality). Relators provide all the ontological relationships within the Dictionary. Relators used in the DDEX Data Dictionary are defined as Core Entities in Annex A.

6.3.2 ENTITY RELATIONSHIPS WITHIN COMPOSITES

A Composite comprises a group of two or more Entity Relationships of any kind. This Entity type is used to define the semantics of complex Entities such as Message Standards. For example, the Composite Entity DDEXC:PartyName contains the following triples:

DDEXC:PartyName	HasEntity	PartyName
PartyName	HasMaxCardinality	1
PartyName	HasDataType	String

This illustrates how a Messaging Standard can be defined in the Data Dictionary as a set of Composites.

6.4 Structural Entity

A Structural Entity is an Entity which is added to the Dictionary because it is necessary to complete the logical structure or ontology of the Dictionary, but which is not used in any Messaging Standard or other DDEX Standard. For example, at the top level the term Entity has been introduced as a parent for all terms, and below that a group of other Entities has been added to provide the necessary infrastructure to support more specialised groupings.

Structural Entities also include those Entities whose existence is implied by other Entities but are not, or are not yet, used in any DDEX Standard. For example, the Entity VersionId is included in several DDEX Messaging Standards, but the thing identified – a "version" – does not appear in any DDEX Message Standard or other DDEX Standard; to ensure the logical coherence of the Dictionary, the Entity Version is therefore included in the DDEX Data Dictionary as a Structural Entity.

There is no other distinction between Structural Entities and other Entities. A Structural Entity may be subsequently used in a DDEX Messaging Standard or other DDEX Standard (and therefore will cease to be a Structural Entity).

6.5 Entity Attributes

6.5.1 ENTITY NAMES

Each Entity has a main name or Headword, and may have any number of alternative names or Synonyms. Names are presented in upper and lower case, with initial capitals for each word, and with no spaces between words (for example: PartyRoleDescription). Words are normally spelled out in full (for example CatalogNumber not CatalogNo), two exceptions being Id (for Identifier) and Ref (for Reference).

In the DDEX Data Dictionary, the Namespace for each Entity is shown as an integral part of its Headword (for example: ern:MusicalWork, DDEXC:PartyDescriptor), with the exception of Entities in the "DDEX" namespace where the namespace designator is omitted for better readability.

6.5.2 LANGUAGE

The DDEX Data Dictionary is written in US English. Information in additional languages may be provided. Such information is not normative.

6.5.3 ENTITY DESCRIPTIONS

Each Entity has a single "global" Description, and a further "local" Description each time it is referenced in a Composite.

The aim of a Description is to explain the Entity's meaning without ambiguity. The first part of a Description is normally in the form of a definition. Definitions are as precise as possible. Where an Entity is a child or a dependent attribute of another Entity, it is defined by direct reference to that Entity: for example, a CreationId is defined as "An Identifier of a Creation"; it can thus be said that CreationId "inherits" or "derives" its meaning from both Creation and Identifier.

The Entities from which meaning is derived are shown with an initial capital letter in the Description. The Descriptions of these Entities can then be reviewed for further information about the meanings which they pass on to their dependents. For example, an Identifier is defined as "A Name which is unique within its namespace", and a Creation is "An entity that is made, directly or indirectly, by one or more human beings". Both of these meanings are inherited by CreationId, so that a fully expanded definition of CreationId might read "a Name which is unique within its namespace of an entity that is made, directly or indirectly, by one or more human beings".

After the formal definition, the Description may include additional comments or examples for the purpose of clarification.

6.5.3.1 Global Descriptions

Each Entity has a "global" Description which summarises its meaning. This Description applies to the Entity irrespective of any Composite within which it used.

6.5.3.2 Local Descriptions

When an Entity is used in a Composite such as an DDEX Messaging Standard, it has a "local" Description which narrows down or "specialises" the global Description to reveal the particular meaning which the Entity has within the Composite. The local meaning never contradicts the global meaning, only refines it by adding more constraints.

6.5.4 EXEMPLARY REPRESENTATION OF AN ENTITY (INFORMATIVE)

Figure 1 below shows how Entities in the DDEX Data Dictionary may be represented.

DDEX Ontology

1 2 3 7 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z All

ddexC:MusicalWorkId

A Composite containing details of an Identifier of a MusicalWork.

Meaning Type Derived

Component Structure				
Is Member Of Composites	***Composite Name***	***Description of Composite Role***	***Cardinality***	***DataType***
	ddexC:MusicalWork	A Composite containing details of an Identifier of the MusicalWork.	1-n	
Components	***Name***	***Description of Element Role***	***Cardinality***	***DataType***
	XmlSequence		1	
	ISWC	The ISWC identifying the MusicalWork.	0-1	String
	OpusNumber	The MusicalWorkId identifying the MusicalWork within the catalog of its Composer (typically of classical music) as an opus number.	0-1	String
	ComposerCatalogNumber	A MusicalWorkId identifying a MusicalWork within the catalog of its Composer (typically of classical music) according to a standardized numbering (eg "K" numbers for Koechel's catalog of Mozart).	0-n	String
	ProprietaryId	A Composite containing details of a ProprietaryId of the MusicalWork.	0-n	ddexC:ProprietaryId
Relationships				
Belongs to Class	IdentifierComposite			

Copyright © 2006, DDEX

Figure 1– Exemplary Representation of an Entity in the DDEX Data Dictionary

7 DDEX Namespaces

A Namespace identifies the DDEX Standard that an Entity belongs to. The DDEX Data Dictionary uses several Namespaces:

- "ddex" for all Simple Entities used within the DDEX Framework.
- "ddexC" for all Composites that are Core Entities;
- Various Component Namespaces for Composites that are defined for use specifically in one DDEX Messaging Standard or other DDEX Standard. Establishing and naming of a Component Namespace is managed by the DDEX Data Dictionary Registration Authority.

7.1 Examples of Usage of Component Namespaces (informative)

Examples for such Component Namespaces are "ern" for Entities stemming from the Electronic Release Notification Message Standard or "dsr" for Entities stemming from the Digital Sales Reporting Message Standard.

8 DDEX Baseline Schema

The syntax of the Entities contained in the DDEX Data Dictionary will be represented in one or more XML schema files – collectively called the DDEX Baseline Schema – for use by various DDEX Messaging Standard and other DDEX Standards.

A Registration Authority (RA) shall be established for creating and maintaining the Schema files. This RA shall be the same RA as for the DDEX Data Dictionary. The requirements for this RA are defined in Clause 9.

The RA shall also make the DDEX Baseline Schema available in accordance with policies which it shall establish.

9 DDEX Data Dictionary and Baseline Schema Maintenance

The DDEX Data Dictionary will grow and develop with each new DDEX Standard that is added and as each existing DDEX Standard is revised. Consequently, it will require constant maintenance. The same applies to the DDEX Baseline Schema, which represents the Entities contained in the Data Dictionary and provides the necessary syntax for the DDEX Messaging Standards.

Every Entity used in a DDEX Messaging Standard, or defined for use in any other DDEX Standard, shall be added to the DDEX Data Dictionary in a manner consistent with this Standard.

The principles for the management of the DDEX Data Dictionary, which will be undertaken by the DDEX Data Dictionary Registration Authority (RA), are set out below. These principles set out how the RA will maintain and make accessible the DDEX Data Dictionary and DDEX Baseline Schema. The mechanism by which these principles are achieved will involve the RA defining a clear set of governance criteria to ensure that the decisions made to meet these principles occur in a manner which maintains the veracity and reliability of all of the DDEX Standards.

1. The RA shall maintain a current version of the DDEX Data Dictionary and the DDEX Baseline Schema and make these accessible to users in accordance with policies which the RA shall establish. It is assumed that this will be achieved by the storage of the DDEX Data Dictionary and the associated Baseline Schema on a DDEX website which may or may not have protected access.

2. The RA shall also enable users of the DDEX Standards to have access to older versions of the DDEX Data Dictionary and the DDEX Baseline Schema. It is also assumed that these older versions will be stored on a DDEX website which may or may not have protected access.

3. The Procedure for the Development and Maintenance of DDEX Standards is set out as Exhibit C in the DDEX Operating Agreement. DDEX Standards can only be created and maintained in accordance with this procedure. It will be the responsibility of the RA, where the creation of new DDEX Standards or the maintenance of existing DDEX Standards are made in accordance with this Procedure which require additions or amendments to the DDEX Data Dictionary to collaborate with all other RAs responsible for DDEX Standards, to ensure that the integrity and coherence of the DDEX Data Dictionary and DDEX Baseline Schema are maintained.

4. The process that shall be implemented by the RA shall allow for timely and efficient implementation of changes to the DDEX Data Dictionary and DDEX Baseline Schema based on the requirements of new or revised DDEX Standards.

10 Externally Managed Message Standards

The DDEX Data Dictionary may contain Entities (whether as Simple Entities or Composites) which relate to a message suites which are not developed and maintained by DDEX but by a Liaison Organisation (Externally Managed Message Suites). This allows consistency to be maintained between DDEX message suites and the Externally Managed Message Suites.

Where the Board of DDEX approves such an arrangement, the DDEX Data Dictionary RA shall operate and maintain those parts of the Data Dictionary which relate to the Externally Managed Message Suite

as if that message suite where managed within DDEX but in collaboration with the Liaison Organisation rather than a DDEX Working Group.

Annex A
(normative)
Core Structural Entities of the DDEX Dictionary

This Annex contains all Core Structural Entities and their Description of the DDEX Data Dictionary.

A.1 Structural Entities

The top level Entities of the DDEX Data Dictionary are Structural Entities. Figure 2 below depicts these Entities. These Core Structural Entities are included in the DDEX Data Dictionary. They form the ontological core of the DDEX Data Dictionary. They are in the DDEX Namespace "DDEX" (see Table 1).

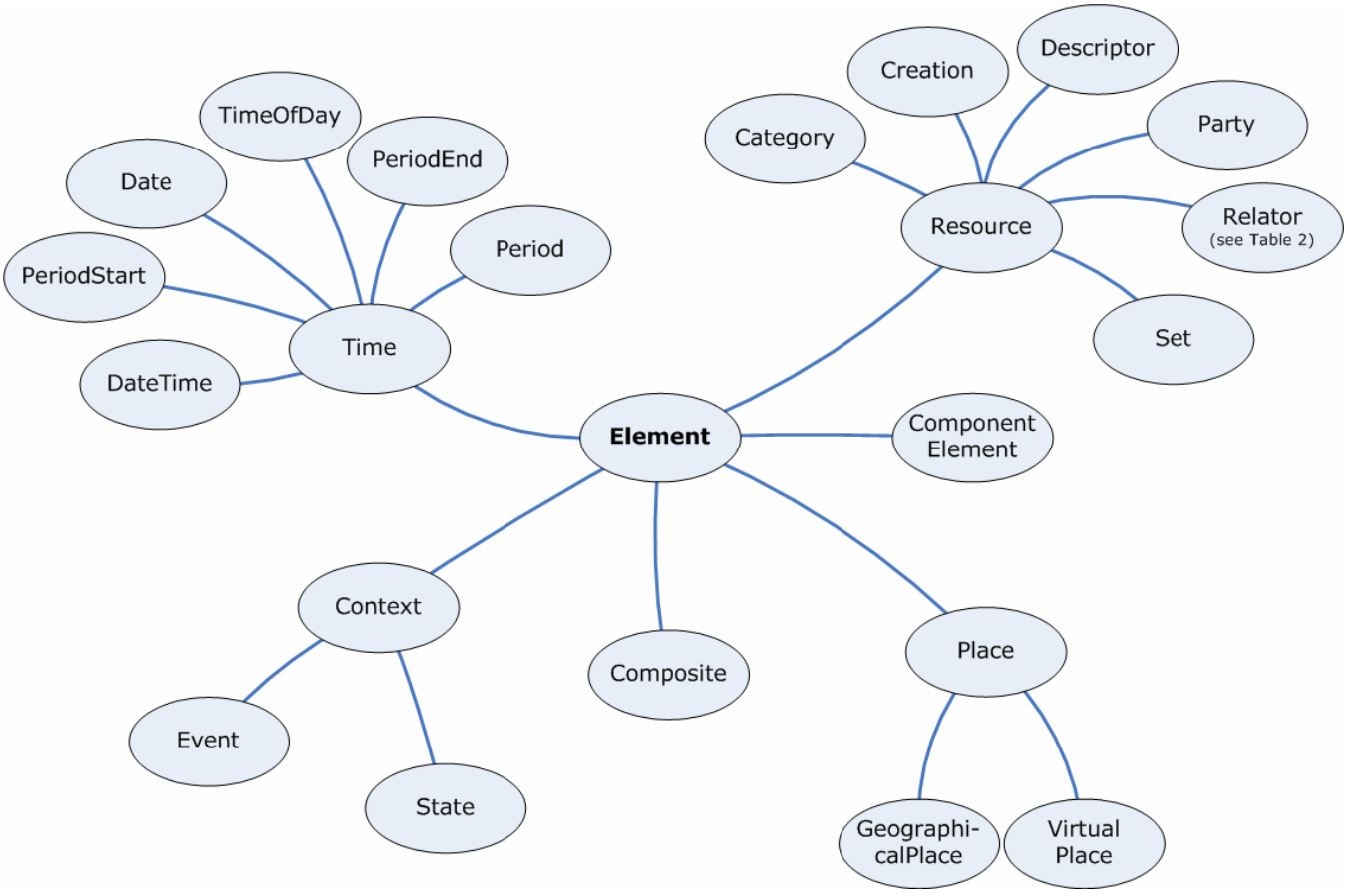

Figure 2– Top-level Core Structural Entities

Not shown in Figure 2 are the Relators that are also included in the Data Dictionary Standard. They are detailed in Table 2.

Table 1 – Core Structural Entities

Headword	Description/Synonym	Parents	Children[1]
Entity	The most basic term is called an Entity, meaning anything which can be referenced with a name. Entity is the parent of all other terms. This introduces the most basic concept of the DDEX Data Dictionary that "an entity must be denoted by a name before it can be recognized in an information system". An Entity may be physical or conceptual, "real" or imaginary – it does not need to "exist" in any other sense than having a name. Entity corresponds to fundamental terms such as "Element" or "Thing" in other models and ontologies.		CompositeEntity Composite Context Place Resource Time
ComponentEntity	A part of a Composite.	Entity	
Context	"The intersection of Time and Place in which Entities may play roles". Each Context must have at least one Time and Place, even if they may only be identified by generic values such as Sometime or Everywhere. It might be suggested that Context is not a true primitive Entity because it is dependent upon Time and Place. However, these Entities are complementary to one another: Time and Place are considered meaningless without their combination, and vice versa. A Context does not have to have any Resources playing roles within it, although Contexts without Resources are not usually of interest to anyone. *Context: Example* Releasing a Product in France on May 5th 2005 is an Event; a Product being available for sale in France from May 5th 2005 until December 25th 2005 is a State.	Entity	Event State
Event	A Context in which some act happens: for example, a person is born, a Work is Performed, a Product is released, a File is Downloaded. In an Event, some attribute is changed. In DDEX the terms "Act" and "Event" are treated as synonymous (even though they are typically represented in everyday language respectively by a verb and a noun). For example, the act "Adapt" or "Adapting" is the verb of which a particular "AdaptingEvent" is an instance, and these terms are treated as interchangeable in DDEX. So, for example, permission to "Adapt" or for "Adapting" is synonymous with a permission to be the Agent in an "AdaptingEvent". Differences in tense and plurality are similarly ignored at the global level: for example, "Create", "Creates", "Created" and "Creating" are all treated as the same term whose particular tense or plurality can be determined unambiguously from its specific context	Context	
State	A Context in which some Resource or Party exists in an unchanging state: for example, a Person lives, a Product is Available, an Error is present. Changes are brought about by Events, which give rise to new States. It is not that no attributes of a Resource or Party change, only that that their defined attributes within that Context do not change. For example, if a person is resident in Canada from 2001 to 2003, this means that their resident status does not change in that period. The fact that they may get married or sign a recording contract in that period does not affect the unchanging nature of their State of residence.	Context	

[1] The Data Dictionary will contain more children for some of the Core Structural Entities. This table only contains the parent-child relationship between Core Structural Entities.

Headword	Description/Synonym	Parents	Children[1]
Place	Place is one of the three primitive EntityTypes and depends on the concept of locality: “Three dimensional place (OED: "A particular portion of space") or virtual place”. At its most abstract, a Place is represented by a set of spatial or virtual co-ordinates. At its most concrete, it may be represented by an Entity which occupies those co-ordinates, such as a building or a computer.	Entity	GeographicalPlace VirtualPlace
GeographicalPlace	A Place in the universe which can be localized by reference to spatial co-ordinates.	Place	
VirtualPlace	A Place which cannot be localized by reference to spatial co-ordinates.	Place	
Composite	A Composite is the name given to a set of Triples grouped together to represent in a structured group of elements in a schema. After the Triple, the Composite is the third basic building block of the DDEX Data Dictionary. Typically a Composite represents a complex formal data structure such as an XML schema, an OO ObjectClass or a relational database table, but a Composite can be used to describe any class of formal relationships. Composites may contain other Composites to any level of granularity. Composites may be used to represent both classes and instances of schemas. *Composite: Example* Examples: an XML Message, a set of ReferenceDescriptiveMetadata, a group of elements within a Message such as "TitlesComposite". A Composite does not represent anything in its own right but is only a collection or "bag" of entities which normally have some kind of relationship. Composites may contain other Composites to any level of granularity.	Entity	
Time	Time is one of the three primitive EntityTypes and depends on the concept of "linear" time, defined in the Oxford English Dictionary as "The indefinite continued progress of existence, events, etc., in past, present and future"). A time may be identified as a *PointInTime* or as a *Period*. *Time: Format* In DDEX times (including dates) are expressed in an ISO 8601:2004 compliant format	Entity	Date DateTime Period PeriodEnd PeriodStart TimeOfDay
Date	A Date represented as a Calendar Year, Month or Day (in ISO 8601 format: YYYY, YYYY-MM or YYYY-MM-DD).	Time	
DateTime	A PointInTimeID which is composed of a Date and a TimeOfTheDay.	Time	
Period	A Time represented as a range between two other Times (with measurable Duration, as opposed to a PointInTime).	Time	
PeriodEnd	A Time which marks the end of a Period.	Time	
PeriodStart	A Time which marks the beginning of a Period.	Time	
TimeOfDay	A Time in a day (in ISO 8601 format: hh:mm:ss)	Time	

Headword	Description/Synonym	Parents	Children[1]
Resource	An Entity in a Context playing a role other than that of Time or Place. Resource is the "catch-all" parent for other entities. Resources may be physical, digital or abstract concepts, and include the Products, Releases, Recordings, Works and other human-made Creations which are the objects of creative and commercial activity; but Resource also includes the people or organizations (Parties) who can act as agents, and all forms of tools and materials which may be used or referenced in the processes described within DDEX.	Entity	Category Creation Descriptor Party Relator Set
Category	A Characteristic of an Entity or a Class to which the Entity belongs. A Category is a "controlled value" which enables an Entity to be grouped or described according to common or measurable attributes. A Category may be expressed in a variety of forms: as A Flag ("True/False"), a Class (eg "DigitalResource"), an adjectival Quality (eg "Green"), a nounal Quality (eg "Audibility") or as a numeric Quantity (eg "4.33MB").	Resource	
Creation	An Output made, directly or indirectly, by one or more human beings	Resource	
Descriptor	A Name or an Annotation of an Entity.	Resource	
Party	A Resource that is a human or other animate being (real or imaginary), a legal person or an organization capable of playing a role as an Agent (or active Resource). Party includes non-human and imaginary "beings".	Resource	
Relator	An entity that describes the relationship between one Entity and another. Relators are not normally used directly in DDEX Messages: Relators are Dictionary linking Entities (or "ontological Entities") which are used to create all of the links which hold the Dictionary structure together. For example, a relationship such as "DealerPrice IsSubClassOf Price" results in a parent-child link being established between "Price" and "DealerPrice". Each Relator has a reverse or "reciprocal" Relator which is used to describe the same Relationship from the reverse point of view. For example, "HasSubClass" is the reciprocal Relator of "IsSubClassOf", and so a statement such as "Price HasSubClass DealerPrice" has the same meaning as "DealerPrice IsSubClassOf Price". This enables the Dictionary to be navigated in all directions. Some Relators (for example, "IsSameAs") are used for describing a Relationship in both directions and so are their own reciprocals.	Resource	See Table 2 for a list of the children of Relator
Set	A Resource that is an enumerated group of Entities or a Class.	Resource	

A.2 Relators

The Relators or “linking Entities” shown in Table 2 below are not used in any Messages, but are all included in the DDEX Data Dictionary as Core Entities. All Relators are in the DDEX Namespace “DDEX”. The following Relators exist within the DDEX Data Dictionary.

Table 2 – Relators

Headword	Description	Parents	Children
HasEntity	The Relator between a Composite and an Entity which is Part of it. Example: "PartyNameComposite HasEntity FullName". Relationships made with this Relator are shown as "Components" in the Dictionary output. The reciprocal Relator of this term is "IsEntityOf".	HasPart	HasXmlAttribute HasXmlEntity
HasInstance	The Relator between a Class and an Entity which belongs to it. Example: "CarrierType HasInstance DVD". Relationships made with this Relator are shown as "Instances" in the Dictionary output. The reciprocal Relator of this term is "IsA".		
HasPart	The Relator between an Entity and another which is a Part of it. Example: "GRid HasPart GridCheckCharacter". Relationships made with this Relator are shown as "Parts" in the Dictionary output. The reciprocal Relator of this term is "IsPartOf".		HasEntity
HasSubClass	The Relator between a Class and another Class which is a specialization of it. Example: "Medium HasSubClass Television". Relationships made with this Relator are shown as "Children" in the Dictionary output. The reciprocal Relator of this term is "IsSubClassOf".		
HasSubRelator	The Relator between a Relator and another Relator which is a specialization of it. Example: "IsPartOf HasSubRelator IsEntityOf". Relationships made with this Relator are shown as "Children" in the Dictionary output. The reciprocal Relator of this term is "IsSubRelatorOf".		
HasXmlAttribute	The Relator between an Entity and another Entity which is expressed as an XmlAttribute of it in an XmlSchema. Example: "ddexc:ProprietaryId HasXmlAttribute Namespace". The reciprocal Relator of this term is "IsXmlAttributeOf".	HasEntity	
HasXmlEntity	The Relator between an Entity and another Entity which is expressed as an XmlEntity of it in an XmlSchema. Example: "DDEXC:ProprietaryId HasXmlEntity Namespace". The reciprocal Relator of this term is "IsXmlEntityOf".	HasEntity	
HasXmlExtension	The Relator between an XmlComposite and another which is an XML Extension of it in an XML Schema. Example: "DDEXC:PartyDescriptor HasXmlExtension DDEXC:Writer". Relationships made with this Relator are shown in the list "Has Same Components As" in the Dictionary output. The reciprocal Relator of this term is "IsXmlExtensionOf".	HasEntity	
IsA	The Relator between an Entity and a Class to which it belongs. Example: "DVD IsA CarrierType". Relationships made with this Relator are shown in the list "Belongs To Class" in the Dictionary output. The reciprocal Relator of this term is "HasInstance".		

Headword	Description	Parents	Children
IsChildOf	The Relator between a specialized Entity and a parent Entity from which it inherits attributes. Example: "Audiovisual IsChildOf Visual". Relationships made with this Relator are shown in the list of "Parents" in the Dictionary output. The reciprocal Relator of this term is "IsParentOf".		
IsEntityOf	The Relator between an Entity and a Composite of which it is a Part. Example: "FullName IsEntityOf PartyNameComposite". Relationships made with this Relator are shown as "Is Member of Composites" in the Dictionary output. The reciprocal Relator of this term is "HasComponentEntity".	IsPartOf	IsXmlAttributeOf IsXmlEntityOf
IsParentOf	The Relator between an Entity and its specialized Entity from which inherits attributes from it. Example: "Audio IsParentOf Audiovisual". Relationships made with this Relator are shown in the list of "Children" in the Dictionary output. The reciprocal Relator of this term is "IsChildOf".		
IsPartOf	The Relator between an Entity and another of which it is a Part. Example: "GRidCheckCharacter IsPartOf GRid". Relationships made with this Relator are shown as "Is Part Of" in the Dictionary output. The reciprocal Relator of this term is "HasPart".		IsEntityOf
IsReciprocalOf	The Relator between a Relator and its reciprocal Relator. Example "IsA IsReciprocalOf HasInstance". Relationships made with this Relator are not shown in the Dictionary output.		
IsSubClassOf	The Relator between a specialized Class and its parent Class. Example: "Television IsSubClassOf Medium". Relationships made with this Relator are shown in the list of "Parents" in the Dictionary output. The reciprocal Relator of this term is "HasSubClass".		
IsSubRelatorOf	The Relator between a specialized Relator to its parent Relator. Example: "IsEntityOf IsSubRelatorOf IsPartOf". Relationships made with this Relator are shown in the list of "Parents" in the Dictionary output. The reciprocal Relator of this term is "HasSubRelator".		
IsXmlAttributeOf	A SubRelator of IsEntityOf describing a Relationship which is expressed as an XmlAttribute in an XmlSchema. Example: "Namespace IsXmlAttributeOf DDEXC:ProprietaryId". The reciprocal Relator of this term is "HasXmlAttribute".	IsEntityOf	
IsXmlEntityOf	A SubRelator of IsEntityOf describing a Relationship which is expressed as an XmlEntity in an XmlSchema. Example: "Namespace IsXmlEntityOf DDEXC:ProprietaryId". The reciprocal Relator of this term is "HasXmlEntity".	IsEntityOf	
IsXmlExtensionOf	The Relator between an XmlComposite and another of which it is an XML Extension in an XML Schema. Example: "DDEXC:Writer IsXmlExtensionOf DDEXC:PartyDescriptor". Relationships made with this Relator are shown in the list "Has Same Components As" in the Dictionary output. The reciprocal of this Relator is "HasXmlExtension".		