



DateTime Data Types

Recommendation 2001-Oct-16

This version:

DateTimeDataTypes-1_1

Previous version:

DateTimeDataTypes-1_0

Editor:

Don Simonson, Robert Haft International
Eric Johns, ESXML
Karl Brophey, EbenX
Kim Bartkus, HR-XML
Scott Miller, IBM
Paul Kiel, HR-XML
Tim Farlow, Authoria

Authors:

Mark Marsden, Ultimate Software

Contributors:

Members of Cross Process Objects workgroup

Copyright statement

©2000, 2001 HR-XML. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America.

Abstract

The date, time and dateTime data types provided in the Schema Recommendation of May 2, 2001 allows for an optional time zone designation. In Consortium work, there will be times when the precise date, time or dateTime value must be known. The objective of this document is to provide HR-XML Schema designers the ability to require or prohibit the Time Zone designation for date, time and dateTime values in a consistent manner.

Status of this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

Table of Contents

1	Overview	3
1.1	<i>Objective</i>	3
1.1.1	Domain Issues	3
1.1.2	Business Reasons	3
1.2	<i>Design Requirements</i>	3
1.3	<i>Scope</i>	4
1.3.1	Major Components	4
1.3.2	Items Within the Design Scope.....	4
1.3.3	Items Outside of Design Scope	4
2	Supported Business Processes	5
2.1	<i>Trading Partner Roles</i>	5
2.2	<i>Business Process Decision Drivers</i>	5
3	DTD/Schema Design	7
3.1	<i>Schema Details</i>	9
3.1.1	DTD Diagrams	9
3.1.2	Schema Diagrams	9
3.1.3	Schema/DTD Elements Explained	10
4	Implementation Considerations	15
5	Appendix A - Vocabulary.....	18
6	Appendix B – Character Representations and Designators	19
7	Appendix C - Document Version History	20
8	Appendix D - Related Documents	20
9	Appendix E – Reference Examples	21
10	Appendix F – DateTime Data Types Schema.....	24

1 Overview

1.1 Objective

The objective of this document is to provide HR-XML Schema designers the tools to incorporate date, time and dateTime values into their work processes in a consistent manner. This is achieved through the use of standard data types, which permit an XML parser to ensure the validity of a desired style of data. Each work group can define its own business process needs and apply any of the applicable "dateTime" data types defined herein which match that need.

1.1.1 Domain Issues

In the HR space, there are many instances where a point in time needs to be captured. The preciseness of the date/time value is often dependent on the business context. A hire date can often be captured as date value, the exact time a person is hired is not important in many transactions. An incident date requires more precision. Many transactions require the date and time an incident took place. Also, in the case of an incident, the time zone may be required. Eleven o'clock in the evening on July 31 in Washington DC, is four o'clock in the morning on August 1 in London. This could mean a difference in coverage for the employee.

Another issue with the current Schema Recommendation is the fact that the time zone designation is optional. A business partner can receive multiple instance documents based on the same schema design. Some documents can contain a dateTime data element that has the time zone, and some of the documents can contain the same dateTime data element that does not contain the time zone. The business partner has no way of ordering the documents based on this dateTime data element.

1.1.2 Business Reasons

Almost all HR and Payroll business processes revolve around some event or string of events in time. All these events have in common a start point, an end point, a duration, or some combination of these, that must be represented in a format that can be efficiently stored, conveyed and processed by HRIS systems. By developing a standard approach to providing date information we will prevent the necessity of other HR-XML Consortium work groups from devoting time and effort to addressing this on a case-by-case basis.

1.2 Design Requirements

The date, time and dateTime data types provided in the Schema Recommendation of May 2, 2001 allows for an optional time zone designation. In Consortium work, there will be times when the precise date, time or dateTime value must be known. The objective of this document is to provide HR-XML Schema designers the ability to require or prohibit the TimeZone designation for date, time and dateTime values in a consistent manner.

The Design requirements must take into account the following:

1. A global concept of time
2. Representing Date and Time values in Schema and in DTD
3. Allow a Date and/or Time value to be stored as a data element or as a data attribute

1.3 Scope

1.3.1 Major Components

Deliver standard Date, Time and DateTime data types that will be used by all of the work groups within the consortium. The goal is consistency.

Provide a yardstick that the CPO can use as a basis for reviewing other workgroup's schema designs.

1.3.2 Items Within the Design Scope

Representing a point in time

Representing a time period (An interval of time with a start and an end)

Representing an unknown date, time or date/time value

Representing an open ended date/time value

Provide standard data types for:

1. Local Date, Local Time, Local DateTime
2. Global Date, Global Time, Global DateTime

1.3.3 Items Outside of Design Scope

Time scheduling rules, such as '7 days before Christmas' are not addressed in this document. Likewise scheduling items of the form 'Every Tuesday at 10:00AM' or 'Every Monday through Friday from 3:30PM to 4:00PM' is not addressed in this document. It is encouraged that for these needs the relevant schema data types be used when possible.

Representing a "duration of time" (2 days, 4 hours, 13 minutes) will not be addressed in this version of the document.

Representing a partial date, such as 2001-06 or a partial time, such as 12:00.

2 Supported Business Processes

2.1 Trading Partner Roles

The purpose of this recommendation is to provide a standard representation for date and time values to schema designers within the consortium, trading partner roles do not apply to this document.

2.2 Business Process Decision Drivers

Before reading this section, you should have a good understanding of the vocabulary defined in Appendix A.

Once it has been determined that a time value needs to be captured, the schema designer should consider how the following issues affect the time value being captured:

Local Time vs. Global Time – If the time value being communicated is representative of a specific point in time in a specific location, then a global time value should be captured. If the time being communicated is representative of a relative point in time, independent of time zone, then a local time value should be used. When in doubt, use a global time value.

Consider the following examples:

Global Time Value:

Business A is located in Washington DC (5 hours behind UTC).

Business B is located in London (0 hours behind UTC).

Business A sends a time value of 2001-04-01T09:30:00 for an employee living in California (8 hours behind UTC).

Problem: How can business B determine the correct time the event really happened? Is it 9:30 Washington time, 9:30 California time or 9:30 London time?

Solution: Business A sends a time value of 2001-04-01T09:30:00-05:00. Now it's clear that the time the event occurred was 9:30 Washington time. Business B can use this information to calculate the time the event occurred in California time or London time as well.

Local Time Value:

An employee lives and works in California (8 hours behind UTC).

The employee's benefit plan, including medical coverage, becomes effective at midnight 2000-08-01 local time. This applies to claims incurred on or after that point in time, where they seek care.

Problem: The employee has an accident requiring medical attention at 2000-08-01T02:00:00 London Time. (2000-07-31T17:00:00 California Time). Is the medical coverage in effect?

Solution: The coverage is in effect on 2000-08-01. The accident occurred at 2000-08-01T02:00:00+00:00. Since one date has a time zone and the other does not, the system should consider the coverage in effect (Year, Month and Day are equal). By explicitly not sending the time zone, it is being communicated that the start of coverage is 2000-08-01. Had the accident occurred at the same point in time, but in California instead of London, the coverage would not be in effect.

Precision – Is the date precise enough, or is a specific point in time (date & time) required? A hire date is an example where the date value is usually precise enough. Most employers are not concerned with the exact second an employee was hired. An incident date, on the other hand may require a precise date and time value. The time of day may be all that's required for a child data element when the parent data element contains the date.

Unbounded / Not Applicable Date Values – Is it acceptable to have a date value that conveys a 'maximum date' value, a 'minimum date' value or 'date value does not apply'? This is common in date ranges. An employee's start date is 2001-04-01. The currently employed employee's termination date has yet to be determined. For processing purposes we want to convey the termination date is **notApplicable**.

Unknown Date Values – Is it possible for the sending system to not know the date value being captured? Is this acceptable? If so, we want to be able to communicate this state to the receiving system. This is done using the **notKnown** value. An employee's start date is 2001-04-01. The sending system does not know the current status of the employee. It would send the **notKnown** value in the employee's termination date.

3 DTD/Schema Design

The best way to explain the CPO Date Data types is to describe how a designer might want their data element/attribute to behave. The issue with the Schema Date, Time and DateTime data types is that they are a little too lenient when it comes to the time zone designator. Most likely, the schema designer will want to enforce or prohibit the time zone designator of the date/time value. This would provide a consistent format for date/time values received in any given element/attribute. The date/time data type names can be broken down as follows:

short description - The value the element/attribute represents (i.e. Date, Time, DateTime)
Any description with the word 'Local' before the short description, defines a value that does not contain a time zone designator. Descriptions without the word 'Local' define a value that must contain a time zone designator, and will contain a global time value or a value that can be easily converted to a global time value.

Nk – The data element/attribute can contain a valid date/time value or the word **notKnown**

Na – The data element/attribute can contain a valid date/time value or the word **notApplicable**

Type – Indicates that this name is not meant to be a data element/attribute name, but is meant to be the foundation for the data elements and attributes.

For example an element/attribute of type **LocalDateNkNaType** will contain a valid date value in the format of **YYYY-MM-DD**, or the value **notKnown**, or the value **notApplicable**. The element/attribute cannot contain any other value. (If it does, the full document will be considered not valid.)

Data Type	Basic Format	Time Zone Required	not Known Allowed	not Applicable Allowed
DateType	YYYY-MM-DD(Z ±hh:mm)	Yes		
DateNkType	YYYY-MM-DD(Z ±hh:mm)	Yes	Yes	
DateNaType	YYYY-MM-DD(Z ±hh:mm)	Yes		Yes
DateNkNaType	YYYY-MM-DD(Z ±hh:mm)	Yes	Yes	Yes
LocalDateType	YYYY-MM-DD			
LocalDateNkType	YYYY-MM-DD		Yes	
LocalDateNaType	YYYY-MM-DD			Yes
LocalDateNkNaType	YYYY-MM-DD		Yes	Yes
TimeType	hh:mm:ss(Z ±hh:mm)	Yes		

Data Type	Basic Format	Time Zone Required	not Known Allowed	not Applicable Allowed
TimeNkType	hh:mm:ss(Z ±hh:mm)	Yes	Yes	
TimeNaType	hh:mm:ss(Z ±hh:mm)	Yes		Yes
TimeNkNaType	hh:mm:ss(Z ±hh:mm)	Yes	Yes	Yes
LocalTimeType	hh:mm:ss			
LocalTimeNkType	hh:mm:ss		Yes	
LocalTimeNaType	hh:mm:ss			Yes
LocalTimeNkNaType	hh:mm:ss		Yes	Yes
DateTimeType	YYYY-MM-DDThh:mm:ss(Z ±hh:mm)	Yes		
DateTimeNkType	YYYY-MM-DDThh:mm:ss(Z ±hh:mm)	Yes	Yes	
DateTimeNaType	YYYY-MM-DDThh:mm:ss(Z ±hh:mm)	Yes		Yes
DateTimeNkNaType	YYYY-MM-DDThh:mm:ss(Z ±hh:mm)	Yes	Yes	Yes
LocalDateTimeType	YYYY-MM-DDThh:mm:ss			
LocalDateTimeNkType	YYYY-MM-DDThh:mm:ss		Yes	
LocalDateTimeNaType	YYYY-MM-DDThh:mm:ss			Yes
LocalDateTimeNkNaType	YYYY-MM-DDThh:mm:ss		Yes	Yes
AnyDateTimeType	YYYY-MM-DD YYYY-MM-DD(Z ±hh:mm) YYYY-MM-DDThh:mm:ss YYYY-MM-DDThh:mm:ss(Z ±hh:mm)	Optional		
AnyDateTimeNkType	YYYY-MM-DD YYYY-MM-DD(Z ±hh:mm) YYYY-MM-DDThh:mm:ss YYYY-MM-DDThh:mm:ss(Z ±hh:mm)	Optional	Yes	
AnyDateTimeNaType	YYYY-MM-DD YYYY-MM-DD(Z ±hh:mm) YYYY-MM-DDThh:mm:ss YYYY-MM-DDThh:mm:ss(Z ±hh:mm)	Optional		Yes
AnyDateTimeNkNaType	YYYY-MM-DD YYYY-MM-DD(Z ±hh:mm) YYYY-MM-DDThh:mm:ss YYYY-MM-DDThh:mm:ss(Z ±hh:mm)	Optional	Yes	Yes

Data Type	Basic Format	Time Zone Required	not Known Allowed	not Applicable Allowed
<ul style="list-style-type: none"> (Z ±hh:mm) means that either the letter Z or a time zone designator in the form of +hh:mm or -hh:mm is allowed. In all of the Basic Formats with a time zone designator (±hh:mm), the letter Z can be used to represent the time zone designation of ±00:00 When the letter Z is used in a DateTime expression the value is said to be a Coordinated Universal Time value. When the time zone designator (±hh:mm) is used in a DateTime expression, the value is a local time value. The time zone designator allows the local time value to be converted to a Coordinated Universal Time value. All of the following are true: <ul style="list-style-type: none"> 2001-04-01T06:00:00-05:00 = 2001-04-01T11:00:00Z 2001-04-01T06:00:00+07:00 = 2001-03-31T23:00:00Z 2001-04-01T06:00:00+00:00 = 2001-04-01T06:00:00Z The only valid formats for Coordinated Universal Time are the following: YYYY-MM-DDThh:mm:ssZ YYYY-MM-DDThh:mm:ss+00:00 YYYY-MM-DDThh:mm:ss-00:00 				

3.1 Schema Details

3.1.1 DTD Diagrams

See section 5.4 Implementation Considerations – DTD vs Schema.

3.1.2 Schema Diagrams

The purpose of this document is to define data types for use with data elements/attributes that contain a date or a date/time value. A Schema diagram does not apply for this document. Please review the Reference Examples in appendix B for sample diagrams/schemas using the data types defined in this document. Appendix C contains the Schema source for these data types.

3.1.3 Schema/DTD Elements Explained

Element / Attribute	Content Model	Description
CPO Literal Types		
NotApplicableLiteral	Restriction Base=string Value = 'notApplicable'	Allows only the value notApplicable to be a valid value for elements/attributes of this type
NotKnownLiteral	Restriction Base=string Value = 'notKnown'	Allows only the value ' notKnown ' to be a valid value for elements/attributes of this type
CPO Date & Time Types		
LocalDateType	Restriction Base=date Pattern= \d\d\d\d-\d\d-\d\d	Allows only a valid date value to be entered in the format YYYY-MM-DD. The parser will verify that the value is a valid date.
LocalDateNkType	Union LocalDateType NotKnownLiteral	Allows the value ' notKnown ' or a valid date value in the format: LocalDateType The parser will verify that the value is a valid date.
LocalDateNaType	Union LocalDateType NotApplicableLiteral	Allows the value ' notApplicable ' or a valid date value in the format: LocalDateType The parser will verify that the value is a valid date.
LocalDateNkNaType	Union LocalDateType NotKnownLiteral NotApplicableLiteral	Allows the value ' notKnown ', ' notApplicable ', or a valid date value in the format: LocalDateType The parser will verify that the value is a valid date.
DateType	Restriction Base=date Pattern= \d\d\d\d-\d\d-\d\d (Z (- +)\d\d:\d\d)	Allows only a valid date value to be entered in the formats: YYYY-MM-DDZ YYYY-MM-DD+hh:mm YYYY-MM-DD-hh:mm The parser will verify that it is a valid date and time zone.

Element / Attribute	Content Model	Description
DateNkType	Union DateType NotKnownLiteral	Allows the value ' notKnown ' or a valid date value in the format: DateType The parser will verify that it is a valid date and time zone.
DateNaType	Union DateType NotApplicableLiteral	Allows the value ' notApplicable ' or a valid date value in the format: DateType The parser will verify that it is a valid date and time zone.
DateNkNaType	Union DateType NotKnownLiteral NotApplicableLiteral	Allows the value ' notKnown ', ' notApplicable ', or a valid date value in the format: DateType The parser will verify that it is a valid date and time zone.
LocalTimeType	Restriction Base=time Pattern= \\d\\d:\\d\\d:\\d\\d(\\.\\d\\d*)?	Allows only a valid time value to be entered in the following format: hh:mm:ss The parser will verify that the value is a valid time.
LocalTimeNkType	Union LocalTimeType NotKnownLiteral	Allows the value ' notKnown ' or a valid time value in the format: LocalTimeType The parser will verify that the value is a valid time.
LocalTimeNaType	Union LocalTimeType NotApplicableLiteral	Allows the value ' notApplicable ' or a valid time value in the format: LocalTimeType The parser will verify that the value is a valid time.
LocalTimeNkNaType	Union LocalTimeType NotKnownLiteral NotApplicableLiteral	Allows the value ' notKnown ', ' notApplicable ', or a valid time value in the format: LocalTimeType The parser will verify that the value is a valid time.

Element / Attribute	Content Model	Description
TimeType	Restriction Base=time Pattern= \d\d:\d\d:\d\d(.\d\d*)? (Z (- +)\d\d:\d\d)	Allows only a valid time value to be entered in one of the following formats: hh:mm:ssZ hh:mm:ss+hh:mm hh:mm:ss-hh:mm The parser will verify that the value is a valid time.
TimeNkType	Union TimeType NotKnownLiteral	Allows the value ' notKnown ' or a valid time value in the format: TimeType The parser will verify that the value is a valid time.
TimeNaType	Union TimeType NotApplicableLiteral	Allows the value ' notApplicable ' or a valid time value in the format: TimeType The parser will verify that the value is a valid time.
TimeNkNaType	Union TimeType NotKnownLiteral NotApplicableLiteral	Allows the value ' notKnown ', ' notApplicable ', or a valid time value in the format: TimeType The parser will verify that the value is a valid time.
LocalDateTimeType	Restriction Base=datetime Pattern= \d\d\d\d-\d\d-\d\d T\d\d:\d\d:\d\d(.\d\d*)?	Allows only a valid date/time value to be entered in the following format: YYYY-MM-DDThh:mm:ss The parser will verify that the value is a valid date and time.
LocalDateTimeNkType	Union LocalDateTimeType NotKnownLiteral	Allows the value ' notKnown ' or a valid date/time value in the format: LocalDateTimeType The parser will verify that the value is a valid date and time.

Element / Attribute	Content Model	Description
LocalDateTimeNaType	Union LocalDateTimeType NotApplicableLiteral	Allows the value ' notApplicable ' or a valid date/time value in the format: LocalDateTimeType The parser will verify that the value is a valid date and time.
LocalDateTimeNkNaType	Union LocalDateTimeType NotKnownLiteral NotApplicableLiteral	Allows the value ' notKnown ', ' notApplicable ', or a valid date/time value in the format: LocalDateTimeType The parser will verify that the value is a valid date and time.
DateTimeType	Restriction Base=datetime Pattern= \\d\\d\\d\\d-\\d\\d-\\d\\d T\\d\\d:\\d\\d:\\d\\d(\\.\\d\\d*)? (Z (- +)\\d\\d:\\d\\d)	Allows only a valid date/time value to be entered in one of the following formats: YYYY-MM-DDThh:mm:ssZ YYYY-MM-DDThh:mm:ss+hh:mm YYYY-MM-DDThh:mm:ss-hh:mm The parser will verify that the value is a valid date and time.
DateTimeNkType	Union DateTimeType NotKnownLiteral	Allows the value ' notKnown ' or a valid date/time value in the format: DateTimeType The parser will verify that the value is a valid date and time.
DateTimeNaType	Union DateTimeType NotApplicableLiteral	Allows the value ' notApplicable ' or a valid date/time value in the format: DateTimeType The parser will verify that the value is a valid date and time.
DateTimeNkNaType	Union DateTimeType NotKnownLiteral NotApplicableLiteral	Allows the value ' notKnown ', ' notApplicable ', or a valid date/time value in the format: DateTimeType The parser will verify that the value is a valid date and time.

Element / Attribute	Content Model	Description
AnyDateTimeType	Union date dateTime	<p>Allows valid date or a date/time value to be entered in one of the following formats: YYYY-MM-DD YYYY-MM-DDZ YYYY-MM-DD+hh:mm YYYY-MM-DD-hh:mm</p> <p>YYYY-MM-DDThh:mm:ss YYYY-MM-DDThh:mm:ssZ YYYY-MM-DDThh:mm:ss+hh:mm YYYY-MM-DDThh:mm:ss-hh:mm</p> <p>The parser will verify that the value is a valid date (and time).</p>
AnyDateTimeNkType	Union AnyDateTimeType NotKnownLiteral	<p>Allows the value 'notKnown' or a valid date/time value in the format: AnyDateTimeType</p> <p>The parser will verify that the value is a valid date (and time).</p>
AnyDateTimeNaType	Union AnyDateTimeType NotApplicableLiteral	<p>Allows the value 'notApplicable' or a valid date/time value in the format: AnyDateTimeType</p> <p>The parser will verify that the value is a valid date (and time).</p>
AnyDateTimeNkNaType	Union AnyDateTimeType NotKnownLiteral NotApplicableLiteral	<p>Allows the value 'notKnown', 'notApplicable', or a valid date/time value in the format: AnyDateTimeType</p> <p>The parser will verify that the value is a valid date (and time).</p>

4 Implementation Considerations

1. All elements and attributes designed to contain a date, a time or a dateTime value should be defined as one of the date, time or dateTime data types described above. By asking the following questions, you can quickly determine which data type to use:

Question	Determination
Are we capturing a specific point in time in a specific location, such as Hire Date? - OR - Are we capturing a relative point in time independent of time zones, such as Benefit Start Date?	We want to capture a time that is global in nature. Do not use data types with the word 'Local' in the name Look for the word 'Local' in the data type name. These types do not allow the time zone designator to be part of the value.
Do we need to capture a date value or do we need to be more specific and capture a date/time value?	If the date value is important and the time value is not needed, look for a type with DateType in the Content Model. If the date and time values are important, look for a type with DateTimeType in the Content Model
Are we allowing a 'maximum' or 'minimum' date value to be allowed?	If the answer is yes, look for a type with notApplicableLiteral in the Content Model
Are we allowing the sending system to admit it doesn't know the value of the date?	If the answer is yes, look for a type with notKnownLiteral in the Content Mode.
Examples	
Example 1: For a Hire Date, we don't care about a time value. Also, the sending system may not know the value.	element name = "HireDate" type = "DateNkType"
Example 2: For a date range, The sending system may not know the start date. The end date will either be a valid date, or it could be open ended.	element name = "StartDate" type = "DateNkType" element Name = "EndDate" type = "DateNaType"
Example 3: The Transaction Date must be very precise. It cannot be unknown or unbounded.	element name="TransactionDate" type = DateTimeType

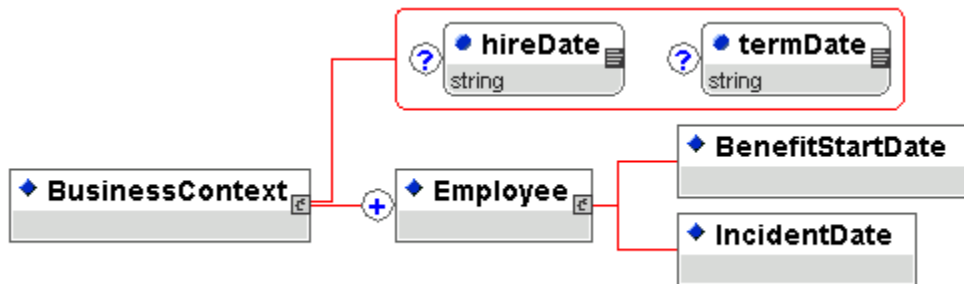
2. **Required & Optional properties, and Default Values** – There are a number of facets of a Date/Time element/attribute that can affect how the parser determines the value of the element/attribute. Consider the following:

Schema Definition	Instance Document Value	Parsed Value
StartDate defined as a data element: Optional (1 st) then Required (2 nd)		
<xsd:element name = "StartDate" type = "DateNkNaType" minOccurs = "0" />	<StartDate>notApplicable </StartDate>	notApplicable
	<StartDate>notKnown </StartDate>	notKnown
	<StartDate></StartDate>	Parser Error
	Data Element does not exist	???
<xsd:element name = "StartDate" type = "DateNkNaType" />	<StartDate>notApplicable </StartDate>	notApplicable
	<StartDate></StartDate>	Parser Error
	Data Element does not exist	Parser Error
StartDate defined as an optional attribute (with and without a default value)		
<xsd:attribute name = "startDate" use = "default" value = "notKnown" type = "DateNkNaType"/>	startDate="notApplicable"	notApplicable
	startDate="notKnown"	notKnown
	Attribute does not exist	notKnown
<xsd:attribute name = "startDate" type = "DateNkNaType"/>	startDate="notApplicable"	notApplicable
	Attribute does not exist	???
StartDate defined as a required attribute		
<xsd:attribute name = "startDate" use = "required" type = "DateNkNaType"/>	startDate="notKnown"	notKnown
	Attribute does not exist	Parser Error

The advantages of defining an optional attribute, with a default value, is (1) the instance document can be smaller, notKnown (or notApplicable) values can be omitted from the instance document; (2) The meaning of the omitted attribute is clear.

3. In any time period represented using a start and end data elements/attributes, the start and end values are inclusive in the time period.

4. This Schema design depends on the CPO DateTime Data Types Schema being available for inclusion into other schemas. While the Technical Steering Committee sets up and defines the infrastructure to make this possible, the Schema Types in Appendix F should be copied into the Schema being developed
5. DTD vs Schema – The contents of this document are focused on Schema representations of date/time values, and Schema features to implement the various CPO date & time types. Since DTD does not support data typing, the parser cannot verify that a valid date value is contained in the data element/attribute. In order to make the conversion from DTD to Schema easier to implement, it is recommended that data elements/attributes containing a date/time value follow the same rules and format as their Schema counter parts. DTD designers should also consider the use of notKnown and notApplicable and document their DTD accordingly. The following example demonstrates one method of documenting the valid values for dateTime “dataTypes” in DTD.



```

<?xml version='1.0' encoding='UTF-8' ?>
<!ELEMENT BusinessContext (Employee+)>

<!-- hireDate valid values: - YYYY-MM-DD, notKnown -->
<!ATTLIST BusinessContext hireDate CDATA "notKnown">

<!-- termDate valid values: YYYY-MM-DD, notApplicable -->
<!ATTLIST BusinessContext termDate CDATA "notApplicable">

<!ELEMENT Employee (BenefitStartDate , IncidentDate)>

<!-- BenefitStartDate valid values: YYYY-MM-DD -->
<!ELEMENT BenefitStartDate (#PCDATA)>

<!-- IncidentDate valid values: YYYY-MM-DDThh:mm:ss(Z|+hh:mm),
notKnown -->
<!ELEMENT IncidentDate (#PCDATA)>

```

Notes: Data elements containing dateTime values should be defined as **#PCDATA**.
Data attributes containing dateTime values should be defined as **CDATA**.

5 Appendix A - Vocabulary

Complete Representation – The representation that includes all the date and time elements associated with the expression. The Schema standard requires the complete representation of date, time and dateTime expressions.

Greenwich Mean Time (GMT) – The official time reference for the world until 1972. Greenwich Mean Time is the mean solar time for the meridian at Greenwich, England. This time scale is based on the earth's motion, which fluctuates in rate by a few thousandths of a second a day. The Royal Greenwich Observatory is the authority of GMT. Also known as Zulu Time, Z Time and World Time.

Coordinated Universal Time (UTC) – The official time reference for the world since 1972-01-01. UTC runs at the rate of the atomic clock. UTC is almost the same as GMT. As the variation between the two time scales approaches a second, a 1 second adjustment (a leap second) is made in UTC. The international UTC time scale is coordinated in Paris by the International Bureau of Weights and Measures (BIPM)

Midnight – The complete representation of midnight can have one of two forms. 00:00:00 represents the beginning of the day. 24:00:00 represents the end of the day. 24:00:00 on 1985 April 12 is the same as 00:00:00 on 1985 April 13 (ISO 8601:2000(E) section 5.3.2 Midnight)

date, calendar – Identification of a particular calendar day, by its calendar year, its calendar month and its ordinal number within its calendar month (ISO:8601:2000(E);Section 3.3 – date, calendar)

dateTime data type– The identification of a particular point in time by its calendar date element and its time of day element, represented as a single expression.

local time – Clock time in public use locally.

NOTE – The difference between local time and UTC time is established by the (national, regional or local) authority responsible for these matters. The difference depends upon the time zone and may also be varied in the course of a year. (ISO 8601:2000(E) section 3.13 local time)

global time – The representation of local time in one location in such a manner that it can be converted to the local time in another location, anywhere in the world.

period of time – The portion of time between two time points. Also known as a time-interval. (ISO 8601:2000(E) section 3.17 period of time)

Note: Periods of time should be represented using a pair of similarly defined data elements or attributes. The start and end values should be inclusive.

duration of time – a quantity (“length”) of time (i.e. 4 days, 2 hours, 26 minutes, 10 seconds).

6 Appendix B – Character Representations and Designators

Character Representations and Designators used in date, time and dateTime expressions (paraphrased from ISO 8601:200(E) sections 5.1.1 and 5.1.2):

YYYY – represents a four digit year time element

MM – represents a two digit month time element

DD – represents a two digit day time element

hh – represents a two digit hour time element

mm – represents a two digit minute time element

ss – represents a two digit second time element

n – represents digit(s), constituting a positive integer or zero

T – time designator, indicates the start of the representation of the time of day in combined date and time of day expressions

Z – UTC designator, immediately (without space) following a data element expressing the time of the day in Coordinated Universal Time (UTC)

Literals used in date, time and dateTime expressions:

notApplicable – When a period of time is “Unbounded”, the “notApplicable” value is used to represent this state. For a period of time, a “notApplicable” value in the “Start Date” represents the beginning of time. A “notApplicable” value in the “End Date” represents the end of time. When “notApplicable” is used to represent a point in time, its meaning must be made clear. For example, “notApplicable” in a “Deceased Date” could mean the employee is still among the living.

notKnown – When a point in time is unknown, the “notKnown” value is used to represent this state. The “notKnown” literal value and the NULL value represent the same state for a date field. By using the “notKnown” literal value, we make it possible to convey this state when a date value is represented as an attribute. Using the example above, a “notKnown” value in a “Deceased Date” would mean the sending system is not able to provide this information.

7 Appendix C - Document Version History

<u>Version</u>	<u>Date</u>	<u>Description</u>
1.1	2001-08-23	Converted to final schema recommendation namespace.
1.1	2001-Oct-16	Approved recommendation by HR-XML Consortium

8 Appendix D - Related Documents

ISO 8601 2nd Edition; ISO8601:2000(E); pub 2000-12-15 – Data elements and interchange formats – Information interchange – Representation of dates and times–
<http://www.iso.ch/cate/d26780.html/>

XML Schema Part 0: Primer– <http://www.w3.org/TR/xmlSchema-0/>

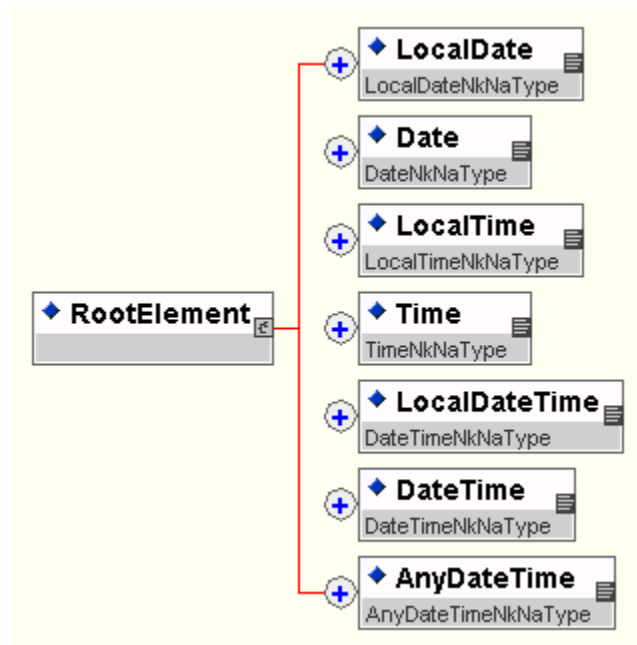
XML Schema Part 2: Datatypes – <http://www.w3.org/TR/xmlSchema-2/>

Coordinated Universal Time Defined - http://www.its.bldrdoc.gov/fs-1037/dir-009/_1277.htm

A Walk through Time; World Time Scales – <http://physics.nist.gov/GenInt/Time/world.html>

9 Appendix E - Reference Examples

Sample Schema Design:



Sample Schema

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns = "d:\cpotest\cpoDateTimeTypesTest.xsd"
  targetNamespace = "d:\cpotest\cpoDateTimeTypesTest.xsd"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "qualified">
  <xsd:include schemaLocation = "cpoDateTimeTypes-2001-06-12.xsd"/>
  <xsd:element name = "RootElement">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name = "LocalDate" type = "LocalDateNkNaType"
          maxOccurs = "unbounded"/>
        <xsd:element name = "Date" type = "DateNkNaType"
          maxOccurs = "unbounded"/>
        <xsd:element name = "LocalTime" type = "LocalTimeNkNaType"
          maxOccurs = "unbounded"/>
        <xsd:element name = "Time" type = "TimeNkNaType"
          maxOccurs = "unbounded"/>
        <xsd:element name = "LocalDateTime"
          type = "DateTimeNkNaType"
          maxOccurs = "unbounded"/>
        <xsd:element name = "DateTime" type = "DateTimeNkNaType"
          maxOccurs = "unbounded"/>
        <xsd:element name = "AnyDateTime"
          type = "AnyDateTimeNkNaType"
          maxOccurs = "unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Sample Valid XML Document

```
<RootElement xmlns = "d:\cpotest\cpoDateTimeTypesTest.xsd"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "d:\cpotest\ file:cpoDateTimeTypesTest.xsd">

  <LocalDate>notKnown</LocalDate>
  <LocalDate>notApplicable</LocalDate>
  <LocalDate>2001-12-31</LocalDate>
  <Date>notKnown</Date>
  <Date>notApplicable</Date>
  <Date>2001-12-31</Date>

  <LocalTime>notKnown</LocalTime>
  <LocalTime>notApplicable</LocalTime>
  <LocalTime>12:13:14</LocalTime>
  <Time>notKnown</Time>
  <Time>notApplicable</Time>
  <Time>12:13:14</Time>
  <Time>12:13:14Z</Time>
  <Time>12:13:14-05:00</Time>

  <LocalDateTime>notKnown</LocalDateTime>
  <LocalDateTime>notApplicable</LocalDateTime>
  <LocalDateTime>2001-12-31T12:13:14</LocalDateTime>
  <DateTime>notKnown</DateTime>
  <DateTime>2001-12-31T12:13:14Z</DateTime>
  <DateTime>2001-12-31T12:13:14-05:00</DateTime>

  <AnyDateTime>notKnown</AnyDateTime>
  <AnyDateTime>notApplicable</AnyDateTime>
  <AnyDateTime>2001-12-31</AnyDateTime>
  <AnyDateTime>2001-12-31T12:13:14</AnyDateTime>
  <AnyDateTime>2001-12-31T12:13:14Z</AnyDateTime>
  <AnyDateTime>2001-12-31T12:13:14+05:00</AnyDateTime>
</RootElement>
```

10 Appendix F - DateTime Data Types Schema

NOTE: The only changes in the schema from the 1_0 version are the schema namespace (to reflect the final XML Schema recommendation namespace) and the change from “timeInstant” to “dateTime” as a data type.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "unqualified">

  <!-- ===== Literal Types ===== -->
  <xsd:simpleType name = "NotKnownLiteral">
    <xsd:restriction base = "xsd:string">
      <xsd:enumeration value = "notKnown"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "NotApplicableLiteral">
    <xsd:restriction base = "xsd:string">
      <xsd:enumeration value = "notApplicable"/>
    </xsd:restriction>
  </xsd:simpleType>

  <!-- ===== Local Date Types ===== -->
  <xsd:simpleType name = "LocalDateType">
    <xsd:restriction base = "xsd:date">
      <xsd:pattern value = "\d\d\d\d-\d\d-\d\d"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "LocalDateNkType">
    <xsd:union memberTypes = "LocalDateType NotKnownLiteral"/>
  </xsd:simpleType>
  <xsd:simpleType name = "LocalDateNaType">
    <xsd:union memberTypes = "LocalDateType NotApplicableLiteral"/>
  </xsd:simpleType>
  <xsd:simpleType name = "LocalDateNkNaType">
    <xsd:union memberTypes = "LocalDateType NotKnownLiteral NotApplicableLiteral "/>
```



```

</xsd:simpleType>

<!-- ===== Date Types ===== -->
<xsd:simpleType name = "DateType">
  <!-- <xsd:pattern value = "\d\d\d\d-\d\d-\d\d(Z|(\+|-)\d\d:\d\d)"/> -->
  <xsd:restriction base = "xsd:date"/>
</xsd:simpleType>
<xsd:simpleType name = "DateNkType">
  <xsd:union memberTypes = "DateType NotKnownLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "DateNaType">
  <xsd:union memberTypes = "DateType NotApplicableLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "DateNkNaType">
  <xsd:union memberTypes = "DateType NotKnownLiteral NotApplicableLiteral "/>
</xsd:simpleType>

<!-- ===== Local Time Types ===== -->
<xsd:simpleType name = "LocalTimeType">
  <xsd:restriction base = "xsd:time">
    <xsd:pattern value = "\d\d:\d\d:\d\d(\.\d\d*)?"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "LocalTimeNkType">
  <xsd:union memberTypes = "LocalTimeType NotKnownLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "LocalTimeNaType">
  <xsd:union memberTypes = "LocalTimeType NotApplicableLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "LocalTimeNkNaType">
  <xsd:union memberTypes = "LocalTimeType NotKnownLiteral NotApplicableLiteral "/>
</xsd:simpleType>

```

```

<!-- ===== Time Types ===== -->
<xsd:simpleType name = "TimeType">
  <!-- <xsd:pattern value = "\d\d:\d\d:\d\d(\.\d\d*)?(Z|(\+|-)\d\d:\d\d)"/> -->
  <xsd:restriction base = "xsd:time"/>
</xsd:simpleType>
<xsd:simpleType name = "TimeNkType">
  <xsd:union memberTypes = "TimeType NotKnownLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "TimeNaType">
  <xsd:union memberTypes = "TimeType NotApplicableLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "TimeNkNaType">
  <xsd:union memberTypes = "TimeType NotKnownLiteral NotApplicableLiteral "/>
</xsd:simpleType>

<!-- ===== DateTime Types ===== -->
<xsd:simpleType name = "LocalDateTimeType">
  <xsd:restriction base = "xsd:dateTime">
    <xsd:pattern value = "\d\d\d\d-\d\d-\d\dT\d\d:\d\d:\d\d(\.\d\d*)?"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "DateTimeNkType">
  <xsd:union memberTypes = "LocalDateTimeType NotKnownLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "DateTimeNaType">
  <xsd:union memberTypes = "LocalDateTimeType NotApplicableLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "DateTimeNkNaType">
  <xsd:union memberTypes = "LocalDateTimeType NotKnownLiteral NotApplicableLiteral "/>
</xsd:simpleType>

```

```

<!-- ===== DateTime Types ===== -->
<xsd:simpleType name = "DateTimeType">
  <xsd:restriction base = "xsd:dateTime">
    <xsd:pattern value = "\d\d\d\d-\d\d-\d\dT\d\d:\d\d:\d\d(. \d\d*)?(Z|(\+|-)\d\d:\d\d)"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "DateTimeNkType">
  <xsd:union memberTypes = "DateTimeType NotKnownLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "DateTimeNaType">
  <xsd:union memberTypes = "DateTimeType NotApplicableLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "DateTimeNkNaType">
  <xsd:union memberTypes = "DateTimeType NotKnownLiteral NotApplicableLiteral "/>
</xsd:simpleType>

<!-- ===== Any DateTime Types ===== -->
<xsd:simpleType name = "AnyDateTimeType">
  <xsd:union memberTypes = " LocalDateType DateType
    LocalDateTimeType DateTimeType "/>
</xsd:simpleType>
<xsd:simpleType name = "AnyDateTimeNkType">
  <xsd:union memberTypes = "AnyDateTimeType NotKnownLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "AnyDateTimeNaType">
  <xsd:union memberTypes = "AnyDateTimeType NotApplicableLiteral"/>
</xsd:simpleType>
<xsd:simpleType name = "AnyDateTimeNkNaType">
  <xsd:union memberTypes = "AnyDateTimeType NotKnownLiteral NotApplicableLiteral "/>
</xsd:simpleType>

```

```

</xsd:schema>

```