

## 20 Names and Dates

This chapter describes an additional tag set which may be used for the encoding of proper names and other phrases descriptive of persons, places, organizations, and also of dates and times, in a manner more detailed than that possible using the elements already provided for these purposes in the core tag set described in chapter 6 *Elements Available in All TEI Documents*.

In section 6.4 *Names, Numbers, Dates, Abbreviations, and Addresses* it was noted that the elements provided in the core allow the encoder to specify that a given text segment is a proper noun, or a *referring string*, and to specify the kind of object named or referred to only by supplying a value for the type attribute. The elements provided by the present tag set allow the encoder both to supply a detailed sub-structure for such referring strings, and also to distinguish explicitly between names of persons, places or organizations.

Similarly, the elements provided here allow the encoder to supply a detailed analysis of the component parts of any expression which denotes a date or time, which is not possible using the elements described in section 6.4.4 *Dates and Times*.

It should be noted however that no provision is made by the present tag set for the representation of the abstract structures, or virtual objects to which names or dates may be said to refer. In simple terms, where the core tag set allows one to represent a *name*, this additional tag set allows one to represent a *personal name*, but neither provides for the direct representation of a *person*. Appropriate mechanisms for the encoding of such interpretative gestures may be found in chapters 15 *Simple Analytic Mechanisms* and 16 *Feature Structures*.

To enable the additional tag set described in the present chapter, a parameter entity *TEI.names.dates* must be declared in the document type subset with the value INCLUDE, as further described in section 3.3 *Invocation of the TEI DTD*. An XML document using the prose base tag set and this additional tag set will thus begin as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TEI.2 PUBLIC "-//TEI P4//DTD Main Document Type//EN" "tei2.dtd" [
  <!ENTITY % TEI.XML 'INCLUDE' >
  <!ENTITY % TEI.prose 'INCLUDE' >
  <!ENTITY % TEI.names.dates 'INCLUDE' >
]>
```

The chapter begins by discussing additional tags for the encoding of component parts of personal names (section 20.1 *Personal Names*), place names (section 20.2 *Place Names*) and organizational names (section 20.3 *Organization names*). Detailed encoding of dates and times is described in section 20.4 *Dates and Time*.

The additional tag set for names and dates, included in the file *teind2.dtd*, has the following overall structure:

```
<!-- 20.: Additional tags for names and dates-->
<!--declarations from 20.1: Personal names inserted here -->
<!--declarations from 20.2.3: Names for places inserted here -->
<!--declarations from 20.3: Organization names inserted here -->
<!--declarations from 20.4.2: Date components inserted here -->
<!-- end of 20.-->
```

When this tag set is enabled, the attribute classes *persPart*, *placePart*, and *tempexp* gain additional attributes to permit more delicate analysis, which replace the default declarations given in *teiclas2.ent*. The model classes declared in that file remain unchanged (see 3.7 *Element Classes*). The parameter entities corresponding with these modified classes are declared in the file *teind2.ent*, as follows:

```
<!-- 20.: Additional classes for names and dates-->
<!ENTITY % x.temporalExpr "" >
<!ENTITY % m.temporalExpr "%x.temporalExpr; %n.dateStruct; | %n.day; |
%n.distance; | %n.hour; | %n.minute; | %n.month; | %n.occasion; | %n.offset; |
%n.second; | %n.timeStruct; | %n.week; | %n.year;">
<!ENTITY % a.personPart '
  key CDATA #IMPLIED
```

```

        reg CDATA #IMPLIED
        type CDATA #IMPLIED
        full (yes | abb | init) "yes"
        sort CDATA #IMPLIED'>
<!ENTITY % a.placePart '
        key CDATA #IMPLIED
        reg CDATA #IMPLIED
        type CDATA #IMPLIED
        full (yes | abb | init) "yes"'>
<!ENTITY % a.temporalExpr '
        value CDATA #IMPLIED
        key CDATA #IMPLIED
        reg CDATA #IMPLIED
        type CDATA #IMPLIED
        full (yes | abb | init) "yes"'>
<!-- end of 20.-->

```

## 20.1 Personal Names

The core `<rs>` and `<name>` elements can distinguish names in a text but are insufficiently powerful to mark their internal components or structure. To conduct nominal record linkage or even to create an alphabetically sorted list of personal names, it is important to distinguish between a family name, a forename and an honorary title. Similarly, when confronted with a referencing string such as “John, by the grace of God, king of England, lord of Ireland, duke of Normandy and Aquitaine, and count of Anjou”, the analyst will often wish to distinguish among components giving some hint as to the status, occupation or residence of the person to whom the name belongs. The following elements are provided for these and related purposes:

- <persName>** contains a proper noun or proper-noun phrase referring to a person, possibly including any or all of the person’s forenames, surnames, honorifics, added names, etc. Attributes include:
  - type** describes the personal name more fully using an open-ended list of words or phrases which help to indicate the function, e.g. “married name”, “maiden name”, “pen name”, “religious name”, etc.
  - Values* Any string of characters.
- <surname>** contains a family (inherited) name, as opposed to a given, baptismal, or nick name.
- <foreName>** contains a forename, given or baptismal name.
- <roleName>** contains a name component which indicates that the referent has a particular role or position in society, such as an official title or rank.
- <addName>** contains an additional name component, such as a nickname, epithet, or alias, or any other descriptive phrase used within a personal name.
- <nameLink>** contains a connecting phrase or link used within a name but not regarded as part of it, such as “van der” or “of”.
- <genName>** contains a name component used to indicating generational information, such as “Junior”, or a number used in a monarch’s name.

As members of the names class, all of these elements share the following attributes:

- key** provides an alternative identifier for the object being named, such as a database record key.
- reg** gives a normalized or regularized form of the name used.

Additionally, all of the above elements except for `<persName>` are members of the class `personPart`, and thus share the following attributes:

- type** provides more culture- linguistic- or application- specific information used to categorize this name component.
- full** indicates whether the name component is given in full, as an abbreviation or simply as an initial.
  - Legal values are:
    - yes** the name component is spelled out in full.
    - abb** the name component is given in an abbreviated form.
    - init** the name component is indicated only by one initial.
- sort** specifies the sort order of the name component in relation to others within the personal name.

The `<persName>` element may be used in preference to the general `<name>` element irrespective of whether or not the components of the personal name are also to be marked. Its `key` and `reg` attributes are used in exactly the same way as those on the `<rs>` and `<name>` elements (see section 6.4 *Names, Numbers, Dates, Abbreviations, and Addresses*). The tag `<persName>` is synonymous with the tag `<name type="person">`, except that its `type` attribute allows for further subcategorization of the personal name for example as a “married”, “maiden”, “pen”, “pseudo” or “religious” name. Consequently the following examples are equivalent:

```
That silly man
<rs key="DPB1" reg="Brown, David Paul" type="person">
David Paul Brown</rs> has suffered the furniture of
his office to be seized the third time for rent.

That silly man
<rs key="DPB1" reg="Brown, David Paul" type="person">
  <name>David Paul Brown</name>
</rs> has suffered ...

That silly man
<name key="DPB1" reg="Brown, David Paul" type="person">
David Paul Brown</name> has suffered ...

That silly man
<persName key="DPB1" reg="Brown, David Paul">
David Paul Brown</persName> has suffered ...
```

The `<persName>` element is more powerful than the `<rs>` and `<name>` elements because distinctive name components occurring within it can be marked as such.

Many cultures distinguish between a family or inherited *surname* and additional personal names, often known as *given names*. These should be tagged using the `<surname>` and `<foreName>` elements respectively and may occur in any order:

```
<persName key="FDR1">
  <surname>Roosevelt</surname>,
  <foreName>Franklin</foreName>
  <foreName>Delano</foreName>
</persName>
<eg><![CDATA[</persName>
<persName key="FDR1">
  <foreName>Franklin</foreName>
  <foreName>Delano</foreName>
  <surname>Roosevelt</surname>
</persName>
```

The `type` attribute may be used with both `<foreName>` and `<surname>` elements to provide further culture- or project- specific detail about the name component, for example:

```
<persName key="FDR1">
  <foreName type="first">Franklin</foreName>
  <foreName type="middle">Delano</foreName>
  <surname>Roosevelt</surname>
</persName>
<persName key="MRT1">
  <foreName type="given">Margaret</foreName>
  <foreName type="abbrev">Maggie</foreName>
  <foreName type="unused">Hilda</foreName>
  <surname type="maiden">Roberts</surname>
  <surname type="married">Thatcher</surname>
</persName>
<persName key="MUAL1" type="religious">
  <foreName>Muhammad</foreName>
  <surname>Ali</surname>
</persName>
```

In the following two examples the `type` attribute of the `<surname>` element is used to indicate so-called double-barrelled or hyphenated surnames:

```

<persName key="KHS1">
  <foreName>Kara</foreName>
  <surname type="combine">Hattersley-Smith</surname>
</persName>
<persName key="NSJS1">
  <foreName>Norman</foreName>
  <surname type="combine">St John Stevas</surname>
</persName>

```

In most cases, patronymics should be treated as forenames, thus:

```

... but it remained for
<persName>
  <foreName>Snorri</foreName>
  <foreName>Sturluson</foreName>
</persName>
to combine the two traditions in cyclic form.

```

When a patronymic is used as a surname, however (e.g. by an individual who otherwise would have no surname, but lives in a culture which requires surnames), it may be tagged as such:

```

Even <persName><foreName>Finnur</foreName>
<surname>Jonsson</surname></persName>
acknowledged the artificiality of the procedure...

```

In the following example, the type attribute is used to distinguish a patronymic from other forenames:

```

<persName key="pn9">
  <foreName sort="2">Sergei</foreName>
  <foreName sort="3" type="patronym">Mikhailovic</foreName>
  <surname sort="1">Uspensky</surname>
</persName>

```

This example also demonstrates the use of the sort attribute common to all members of the personPart class; its effect is to state the sequence in which <foreName> and <surname> elements should be combined when constructing a sort key for the name.

Some names include generational or dynastic information, such as “Junior”, or “the Elder”, or a number: the <genName> element may be used to distinguish these from other parts of the name, as in the following examples:

```

<persName key="HEMA1">
  <surname>Marques</surname>
  <genName>Junior</genName>,
  <foreName>Henrique</foreName>
</persName>
<persName>
  <foreName>Charles</foreName>
  <genName>II</genName>
</persName>
<persName>
  <foreName>Rudolf</foreName>
  <genName>II</genName>
  <surname type="dynasty">Hapsburg</surname>
</persName>

```

It is also often convenient to distinguish phrases (historically similar to the generational labels mentioned above) used to link parts of a name together, such as “von”, “of”, “de” etc. It is often a matter of arbitrary choice whether or not such components are regarded as part of the surname or not; the <nameLink> element is provided as a means of making clear what the correct usage should be in a given case, as in the following examples:

```

<persName key="DUD01">
  <roleName type="honorific" full="abb">Mme</roleName>
  <nameLink>de la</nameLink>
  <surname>Roche foucault</surname>
</persName>

```

```

<persName>
  <foreName>Walter</foreName>
  <surname>de la Mare</surname>
</persName>

```

Finally, the `<addName>` and `<roleName>` elements are used to mark all name components other than those already listed. The distinction between them is that a `<roleName>` encloses an associated name component such as an aristocratic or official title which exists in some sense independently of its bearer. The distinction is not always a clear one. As elsewhere, the `type` attribute may be used with either element to supply culture- or application- specific distinctions. Some typical values for this attribute for names in the Western European tradition follow:

**nobility** An inherited or life-time title of nobility such as ‘Lord’, ‘Viscount’, ‘Baron’, etc.

**honorific** An academic or other honorific prefixed to a name e.g. ‘Doctor’, ‘Professor’, ‘Mrs.’, etc.

**office** Membership of some elected or appointed organization such as ‘President’, ‘Governor’, etc.

**military** Military rank such as ‘Colonel’.

**epithet** A traditional descriptive phrase or nick-name such as ‘The Hammer’, ‘The Great’, etc.

Note, however, that the *role* a person has in a given context (such as ‘witness’, ‘defendant’ etc. in a legal document) should not be encoded using the `<roleName>` element, since this is intended to describe the role of this part of the name, not the role of the person bearing the name.

Here are some further examples of the usage of these elements:

```

<persName key="PGK1">
  <roleName type="nobility">Princess</roleName>
  <foreName>Grace</foreName>
</persName>

<persName key="GRM01" type="pseudo">
  <addName type="honorific">Grandma</addName>
  <surname>Moses</surname>
</persName>

<persName key="MRSR01">
  <addName type="honorific">Mrs</addName>
  <surname>Robinson</surname>
</persName>

<persName key="STAU1">
  <roleName type="office">Saint</roleName>
  <foreName>Augustine</foreName>
</persName>

<persName key="SLWICL1">
  <roleName type="office">President</roleName>
  <foreName>Bill</foreName>
  <surname>Clinton</surname>
</persName>

<persName key="MOGA1">
  <roleName type="military">Colonel</roleName>
  <surname>Gaddafi</surname>
</persName>

<persName key="FRTG1">
  <foreName>Frederick</foreName>
  <addName type="epithet">the Great</addName>
</persName>

```

A name may have any combination of the above elements:

```

<persName key="EGBR1">
  <roleName type="office">Governor</roleName>
  <foreName sort="2">Edmund</foreName>
  <foreName reg="Gerald" full="init" sort="3">G</foreName>.
  <addName type="nick">Jerry</addName>
  <addName type="epithet">Moonbeam</addName>
  <surname sort="1">Brown</surname>

```

```

    <genName full="abb">Jr</genName>.
  </persName>

```

Although highly flexible, these mechanisms for marking personal name components will not cater for every personal name and processing need. Where the internal structure of personal names is highly complex or where name components are particularly ambiguous, feature structures are recommended as the most appropriate mechanism to mark and analyze them, as further discussed in chapter 16 *Feature Structures*.

The elements discussed in this section are formally defined as follows:

```

<!-- 20.1: Personal names-->
<!ELEMENT persName %om.RR; ( #PCDATA | %m.personPart;
    | %m.phrase; | %m.Incl; )* >
<!ATTLIST persName
    %a.global;
    %a.names;
    type CDATA #IMPLIED
    TEIform CDATA 'persName' >
<!ELEMENT surname %om.RR; %phrase.seq;>
<!ATTLIST surname
    %a.global;
    %a.personPart;
    TEIform CDATA 'surname' >
<!ELEMENT foreName %om.RR; %phrase.seq;>
<!ATTLIST foreName
    %a.global;
    %a.personPart;
    TEIform CDATA 'foreName' >
<!ELEMENT genName %om.RR; %phrase.seq;>
<!ATTLIST genName
    %a.global;
    %a.personPart;
    TEIform CDATA 'genName' >
<!ELEMENT nameLink %om.RR; %phrase.seq;>
<!ATTLIST nameLink
    %a.global;
    %a.personPart;
    TEIform CDATA 'nameLink' >
<!ELEMENT addName %om.RR; %phrase.seq;>
<!ATTLIST addName
    %a.global;
    %a.personPart;
    TEIform CDATA 'addName' >
<!ELEMENT roleName %om.RR; %phrase.seq;>
<!ATTLIST roleName
    %a.global;
    %a.personPart;
    TEIform CDATA 'roleName' >
<!-- end of 20.1-->

```

## 20.2 Place Names

Like other proper nouns or noun phrases used as names, place names can simply be marked up with the `<rs>` element, or with the `<name>` element. For cartographers and historical geographers, however, the component parts of a place name provide important information about the relation between the name and some spot in space and time. They also provide important evidence in historical linguistics. For such applications and others in which the internal structure of a place name is to be encoded, the `<placeName>` element and its subcomponents should be used.

**<placeName>** contains an absolute or relative place name.

**<settlement>** contains the name of the smallest component of a place name expressed as a hierarchy of geo-political or administrative units as in “Rochester”, New York; “Glasgow”, Scotland.

**<region>** in an address, contains the state, province, county or region name; in a place name given as a hierarchy of geo-political units, the **<region>** is larger or administratively superior to the **<settlement>** and smaller or administratively less important than the **<country>**.

**<country>** in an address, gives the name of the nation, country, colony, or commonwealth; in a place name given as a hierarchy of geo-political units, the **<country>** is larger or administratively superior to the **<region>** and smaller than the **<bloc>**.

**<bloc>** a geo-political unit containing one or more nation states.

**<geogName>** a name associated with some geographical feature such as “Windrush Valley” or “Mount Sinai”. Attributes include:

**type** provides more culture- linguistic- or application- specific information used to categorize this name component.

*Values* one of a set of codes defined for the application.

**<geog>** contains a common noun identifying some geographical feature contained within a geographic name, such as “valley”, “mount” etc.

**<distance>** that part of a relative temporal or spatial expression which indicates the distance between the place or time denoted by it and the place or time referred to within it. Attributes include:

**exact** indicates the degree of accuracy associated with the distance.

*Legal values are:*

Y The distance is exact.

N The distance is approximate.

U Accuracy unavailable or unknown.

**<offset>** that part of a relative temporal or spatial expression which indicates the direction of the offset between the two place names, dates, or times involved in the expression.

As members of the names class, all these elements share the following attributes:

**key** provides an alternative identifier for the object being named, such as a database record key.

**reg** gives a normalized or regularized form of the name used.

Additionally, all of the above elements are members of the class `placePart`, and thus share the following attributes:

**type** provides more culture- linguistic- or application- specific information used to categorize this name component.

**full** indicates whether the place name component is given in full, as an abbreviation or simply as an initial. Legal values are:

**yes** the name component is spelled out in full.

**abb** the name component is given in an abbreviated form.

**init** the name component is indicated only by one initial.

Like the **<persName>** element discussed in section 20.1 *Personal Names*, the **<placeName>** element may be regarded simply as an abbreviation for the tags **<name type="place">** or **<rs type="place">**. The following encodings are thus equivalent:<sup>150</sup>

```
After spending some time in our
<rs key="NY1" type="place">modern
<name key="BA1" type="place">Babylon</name></rs>,
<name key="NY1" type="place">New York</name>,
I have proceeded to the
<rs key="PH1" type="place">City of Brotherly Love</rs>.
```

```
After spending some time in our
<placeName key="NY1">modern
<placeName key="BA1">Babylon</placeName></placeName>,
<placeName key="NY1">New York</placeName>,
I have proceeded to the
<placeName key="PH1">City of Brotherly Love</placeName>.
```

<sup>150</sup> Strictly, a suitable value such as `figurative` should be added to the two place names which are presented periphrastically in the second example here, in order to preserve the distinction indicated by the choice of `<rs>` rather than `<name>` to encode them in the first version.

As indicated above, the `<placeName>` may simply contain a character string and its type attribute may be used to provide a sub-categorization of place names. Alternatively, it may contain more detailed sub components. A place name may be analysed in several different ways: as a geo-political unit, using a hierarchy of descriptive names (see section 20.2.1 *Geo-political Place Names*); in terms of geographic features such as mountains and rivers (see section 20.2.2 *Geographic Names*); relative to other place names (see section 20.2.3 *Relative Place Names*).

### 20.2.1 Geo-political Place Names

A place name is sometimes given as sequence of geo-political or administrative units, often arranged in ascending sequence according to their size or administrative importance, for example: “Rochester, New York”, or as a single such unit, for example “Belgium”. The more detailed component elements listed above (`<settle>` for a settlement, such as a village, town or city; `<region>` for any administrative unit such as a county, parish or state; `<country>` for a politically recognized national entity; or `<bloc>` for any grouping of such entities) have been chosen for their generality of application. They may be tailored more closely to project- and culture-specific needs by specifying appropriate values in their respective type attributes, as in the following example:

```
<placeName key="RNY1">
  <settlement type="city">Rochester</settlement>,
  <region type="state">New York</region>
</placeName>
<placeName key="LSEA1">
  <country type="nation">Laos</country>,
  <bloc type="sub-continent">Southeast Asia</bloc>
</placeName>
```

Note that, even in the case where only one of these component place name elements is used, the `<placeName>` element must still be present.

```
I'd rather be in
<placeName><settlement key="RNY1" type="city">Rochester</settlement></placeName>
than any other place I know.
```

### 20.2.2 Geographic Names

Places may also be named in terms of geographic features such as mountains, lakes or rivers, independently of geo-political units. The `<geogName>` is provided to mark up such names, as an alternative to the `<placeName>` element discussed above. It contains a sequence of phrase level elements, optionally extended by the following special element:

`<geog>` contains a common noun identifying some geographical feature contained within a geographic name, such as “valley”, “mount” etc.

For example:

```
<geogName key="MIRI1" type="river">Mississippi River</geogName>
```

Where the `<geog>` element is used to characterize the kind of geographic feature being named, the `<name>` element will generally also be used to mark the associated proper noun or noun phrase:

```
<geogName key="MIRI1" type="river">
  <name>Mississippi</name>
  <geog>River</geog>
</geogName>
```

A more complex example, showing a variety of practices, follows:

```
The isolated ridge separates two great corridors which run from
<name key="GLC01" type="place">Glencoe</name> into
<geogName key="GLET1" type="glen">
  <geog reg="glen">Glen</geog>
  <name>Etive</name>
</geogName>, the
<geogName key="LAGA1" type="hill">
  <geog lang="gaelic" reg="sloping hill face">Lairig</geog>
  <name>Gartain</name>
</geogName> and the
<geogName key="LAEI1" type="hill">
  <geog lang="gaelic" reg="sloping hill face">Lairig</geog>
  <name>Eilde</name>
</geogName>
```



### 20.2.3 Relative Place Names

All the place name specifications so far discussed are *absolute*, in the sense that they define only one place. A place may however be specified in terms of its relationship to another place, for example “10 miles northeast of Paris” or “near the top of Mount Sinai”. These *relative place names* will contain a place name which acts as a referent (e.g. “Paris” and “Mount Sinai”). They will also contain a word or phrase indicating the position of the place being named in relation to the referent (e.g. “the top of”, “north of”). A distance, possibly only vaguely specified, between the referent place and the place being indicated may also be present (e.g. “10 miles”, “near”).

Relative place names may be encoded using the following elements in combination with either a <placeName> or a <geogName> element.

<**offset**> that part of a relative temporal or spatial expression which indicates the direction of the offset between the two place names, dates, or times involved in the expression.

<**distance**> that part of a relative temporal or spatial expression which indicates the distance between the place or time denoted by it and the place or time referred to within it. Attributes include:

**exact** indicates the degree of accuracy associated with the distance.

*Legal values are:*

- Y The distance is exact.
- N The distance is approximate.
- U Accuracy unavailable or unknown.

Some examples of relative place names are:

```
<placeName key="NRPA1">
  <offset>near the top of</offset>
  <geogName>
    <geog>Mount</geog>
    <name>Sinai</name>
  </geogName>
</placeName>

<placeName key="NEPA1">
  <distance>10 miles</distance>
  <offset>north of</offset>
  <settlement type="city">Paris</settlement>
</placeName>
```

The internal structure of place names is like that of personal names — complex and subject to an enormous amount of variation across time and different cultures. The recommendations in this section will be adequate for a majority of users and applications. They may not, however, satisfy the most specialized inquiries and/or applications in which case it is recommended that the internal structure of place names be represented using feature structures (16 *Feature Structures*).

The elements discussed in this section are formally defined as follows:

```
<!-- 20.2.3: Names for places-->
<!ELEMENT placeName %om.RR; ( #PCDATA | %m.placePart;
  | %m.phrase; | %m.Incl; )* >
<!ATTLIST placeName
  %a.global;
  %a.names;
  TEIform CDATA 'placeName' >
<!ELEMENT settlement %om.RR; %phrase.seq;>
<!ATTLIST settlement
  %a.global;
  %a.names;
  %a.typed;
  TEIform CDATA 'settlement' >
<!ELEMENT region %om.RR; %paraContent;>
<!ATTLIST region
  %a.global;
  %a.names;
  %a.typed;
  TEIform CDATA 'region' >
```

```

<!ELEMENT country %om.RR; %paraContent;>
<!ATTLIST country
  %a.global;
  %a.names;
  %a.typed;
  TEIform CDATA 'country' >
<!ELEMENT bloc %om.RR; %phrase.seq;>
<!ATTLIST bloc
  %a.global;
  %a.names;
  %a.typed;
  TEIform CDATA 'bloc' >
<!ELEMENT offset %om.RR; ( #PCDATA | %m.Incl; )*>
<!ATTLIST offset
  %a.global;
  %a.temporalExpr;
  TEIform CDATA 'offset' >
<!ELEMENT distance %om.RR; %phrase.seq;>
<!ATTLIST distance
  %a.global;
  %a.temporalExpr;
  exact ( Y | N | U ) "U"
  TEIform CDATA 'distance' >
<!ELEMENT geogName %om.RR; (#PCDATA | geog | name | %m.Incl; )*>
<!ATTLIST geogName
  %a.global;
  %a.names;
  type CDATA #IMPLIED
  TEIform CDATA 'geogName' >
<!ELEMENT geog %om.RR; (#PCDATA)>
<!ATTLIST geog
  %a.global;
  %a.names;
  %a.typed;
  TEIform CDATA 'geog' >
<!-- end of 20.2.3-->

```

### 20.3 Organization names

Like names of persons or places, organization names can be marked as referent strings or as proper names with the <rs> and <name> elements. For certain applications it may be desirable to mark the component parts of an organization. In some historical and social scientific studies, for example, the component parts of an organization names may give crucial clues which help to characterizing the organization in terms of its geographical location, ownership, likely number of employees, management structure etc. The elements discussed in this section are recommended for this purpose and include:

**<orgName>** contains an organizational name. Attributes include:

**type** more fully describes the organization indicated in the organizational name. Possible values include “voluntary”, “political”, “governmental”, “industrial”, “commercial”, etc.

*Values*

**key** provides an alternative identifier for the organization being named, such as a database record key.

*Values* any string

**reg** (regularization) gives a normalized or regularized form of the organization name

*Values* Any string of characters.

**<orgTitle>** contains the proper name component of an organizational name. Attributes include:

**type** more fully describes the organization title. Possible values include “formal”, “colloquial”, “acronym”, etc.

*Values* any string of characters

**reg** (regularization) gives a normalized or regularized form of the organization title.

*Values* Any string of characters.

**<orgType>** indicates a part of the organization name which contains information about the organization's structure or function. Attributes include:

**type** more fully describes the organization type specified in the name component. Possible values include "function", "structure", etc.

*Values* any string of characters

**reg** (regularization) gives a normalized or regularized form of the organization type

*Values* Any string of characters.

**<orgDivn>** indicates a division, branch or department specified in an organizational name. Attributes include:

**type** more fully describes the organization division specified in the name component. Possible values include "branch", "department", "section", "division", etc.

*Values* any string of characters

**reg** (regularization) gives a normalized or regularized form of the organizational division.

*Values* Any string of characters.

The **<orgName>** element should be used when it is desirable to mark an organization name irrespective of whether or not its components are also to be marked. In effect the **<orgName>** element is a special case of a **<name>** and thus of an **<rs>** element. Consequently, the following examples are synonymous, though the last is preferred:

```
About a year back, a question of considerable
interest was agitated in the <rs key="PAS1" type="org">
Pennsyla. Abolition Society</rs>.
```

```
About a year back, a question of considerable
interest was agitated in the <rs key="PAS1" type="org">
<name>Pennsyla. Abolition Society</name></rs>.
```

```
About a year back, a question of considerable
interest was agitated in the
<name key="PAS1" type="org">Pennsyla. Abolition
Society</name>.
```

```
About a year back, a question of considerable
interest was agitated in the
<orgName type="voluntary" key="PAS1"
reg="Pennsylvania Abolition Society">
Pennsyla. Abolition Society</orgName>.
```

Like the **<rs>** and **<name>** elements, the **<orgName>** element has a key attribute with which an external identifier such as a database key can be assigned to the organization name. It also has a type attribute with which the organization named in the expression can be described, and a reg attribute with which the organization name can be presented in a regularized form.

The **<orgTitle>** element is used to mark the expression which provides the proper name component of an organization name. For example:

```
Mr Frost will be able to earn an extra fee from
<orgName type="media" key="BSB1">
  <orgTitle type="acronym">BSkyB</orgTitle>
</orgName>
rather than the
<orgName type="media" key="BBC1">
  <orgTitle type="acronym" reg="British Broadcasting Corporation">BBC</orgTitle>
</orgName>
```

Where personal names are encountered as component parts of an organization's title, as in "Ernst & Young", these may be tagged with the appropriate personal name elements as discussed in 20.1 *Personal Names*. Examples include:

```
<orgName type="accountancy partnership" key="EY1">
  <orgTitle>
    <persName>
```

```

        <surname>Ernst</surname>
    </persName> &amp;
    <persName>
        <surname>Young</surname>
    </persName>
</orgTitle>
</orgName>

```

Organization names may also contain within them place names which, in some applications, may yield vital clues as to the organization's location and or sphere of influence. These components should be tagged with the appropriate place name tags (20.2 *Place Names*). Examples include:

```

A spokesman from
<orgName type="computers" key="IBM1">
    <orgTitle reg="International Business Machines">IBM</orgTitle>
    <placeName>
        <country key="UNKI1" reg="United Kingdom">UK</country>
    </placeName>
</orgName> said ... The feeling in <placeName><country key="CAN1"
type="nation">Canada</country></placeName> is one of strong aversion to the
<orgName type="government" key="USG1">United States Government</orgName>,
and of predilection for self-government under the <orgName type="government"
reg="British monarchy">English Crown</orgName>

```

The `<orgType>` element is used to mark those components of an organization name which indicate something about the structure or function of the organization. Examples include:

```

<orgName type="utility company" key="WWPC1">
    <name type="state">Washington</name>
    <orgType type="function">Water Power</orgType>
    <orgType type="structure" reg="incorporated">Inc.</orgType>
</orgName>

THE TICKET which you will receive herewith has been formed by
the <orgName type="political" key="WHI1" reg="Whig party">
    <orgTitle>Democratic Whig</orgTitle>
    <orgType type="function">Party</orgType>
</orgName> after the most careful deliberation,
with a reference to all the great objects of NATIONAL, STATE,
COUNTY and CITY concern, and with a single eye to the
<hi>Welfare and Best Interests of the Community</hi>.

```

Organizational names may also be specified hierarchically particularly where the named organization is itself a department or a branch of a larger organizational entity. "The Department of Modern History, Glasgow University" is an example. The `<orgDivn>` element is recommended wherever it is desirable to isolate the independent levels of an organizational hierarchy that are specified in an organization name. Examples include:

```

<orgName type="academic" key="DMHGU1">
    <orgDivn type="department">Department of Modern History</orgDivn>,
    <name type="city">Glasgow</name>
    <orgType type="function">University</orgType>
</orgName>

```

Although highly flexible, the mechanisms discussed here for marking the components of organization names will not cater for every processing need or organizational name that is encountered. Where the internal structure of organization names is highly complex, where name components are particularly ambiguous, or where it is important to indicate the assumptions made in the evaluation of an organization name, then feature structure notation is recommended (16 *Feature Structures*).

The formal declaration of the elements discussed in this section include:

```

<!-- 20.3: Organization names-->
<!ELEMENT orgName %om.RR; ( #PCDATA | orgTitle | orgType |
                             orgDivn | %m.phrase; | %m.Incl; ) * >
<!ATTLIST orgName
    %a.global;

```

```

        type CDATA #IMPLIED
        key CDATA #IMPLIED
        reg CDATA #IMPLIED
        TEIform CDATA 'orgName' >
<!ELEMENT orgTitle %om.RR; %phrase.seq; >
<!ATTLIST orgTitle
    %a.global;
    type CDATA #IMPLIED
    reg CDATA #IMPLIED
    TEIform CDATA 'orgTitle' >
<!ELEMENT orgType %om.RR; %phrase.seq; >
<!ATTLIST orgType
    %a.global;
    type CDATA #IMPLIED
    reg CDATA #IMPLIED
    TEIform CDATA 'orgType' >
<!ELEMENT orgDivn %om.RR; %phrase.seq; >
<!ATTLIST orgDivn
    %a.global;
    type CDATA #IMPLIED
    reg CDATA #IMPLIED
    TEIform CDATA 'orgDivn' >
<!-- end of 20.3-->

```

## 20.4 Dates and Time

The following elements for the encoding of dates and times were introduced in section 6.4.4 *Dates and Times*:

**<date>** contains a date in any format. Attributes include:

**calendar** indicates the system or calendar to which the date belongs.

*Values* Recommended values include: *Gregorian, Julian, Roman, Mosaic, Revolutionary, Islamic.*

**value** gives the value of the date in some standard form, usually yyyy-mm-dd.

*Values* Any string representing a date in standard format; recommended form is ISO 8601:2000 5.2.1.1 Complete representation, extended format (yyyy-mm-dd)

**certainty** indicates the degree of precision to be attributed to the date.

*Values* Any appropriate value, e.g. *ca., approx, after, before.*

**<time>** contains a phrase defining a time of day in any format. Attributes include:

**zone** indicates time zone or place name wherever this is necessary to evaluate a temporal expression.

*Values* contains a word or phrase such as 'GMT' or 'Eastern Standard Time' which might be helpful in evaluating the temporal expression.

**value** gives the value of the time in some standard form, usually hh:mm.

*Values* Any string representing a time in standard format; recommended forms are the extended formats from ISO 8601:2000 (hh:mm, hh:mmZ, hh:mm±hh)

**type** indicates something about the type of temporal expression being tagged.

*Legal values are:*

**am** indicates a temporal expression made on the basis of a twelve-hour clock and referring to a time between midnight and noon.

**pm** indicates a temporal expression made on the basis of a twelve-hour clock and referring to a time between noon and midnight.

**24hour** indicates a temporal expression made on the basis of a twenty-four-hour clock.

**descriptive** indicates a temporal expression made in descriptive terms, e.g. noon.

While adequate for many applications, these elements do not allow for the representation of the internal structure of expressions indicating dates or times, which may however be of importance for the correct

interpretation of such expressions, or for certain kinds of analytic applications. In this section, we introduce the following special-purpose elements, for use when the internal structure of a temporal expression is to be encoded:

**<dateStruct>** contains an internally structured representation of a date. Attributes include:

**calendar** indicates the system or calendar to which the date belongs.

*Values* Recommended values include: *Gregorian, Julian, Roman, Mosaic, Revolutionary, Islamic.*

**exact** indicates the degree of precision to be attributed to the date.

*Values* Any appropriate value, e.g. *ca., approx, after, before.*

**<timeStruct>** contains an internally structured representation for a time of day. Attributes include:

**zone** indicates time zone or place name wherever this is necessary to evaluate a temporal expression.

*Values* contains a word or phrase such as 'GMT' or 'Eastern Standard Time' defining a time zone.

Two types of temporal expressions are envisaged for dates and times: absolute and relative. An *absolute temporal expression* is composed of a sequence of the following elements, possibly interspersed with character data:

**<day>** the day component of a structured date.

**<week>** the week component of a structured date.

**<month>** the month component of a structured date.

**<year>** the year component of a date.

**<second>** the second component of a structured time-expression.

**<minute>** the minute component of a structured time-expression.

**<hour>** the hour component of a temporal expression

**<occasion>** a temporal expression (either a date or a time) given in terms of a named occasion such as a holiday, a named time of day, or some notable event.

A *relative temporal expression* describes a date or time with reference to some other (absolute) temporal expression, and thus contains the following elements in addition to those listed above:

**<distance>** that part of a relative temporal or spatial expression which indicates the distance between the place or time denoted by it and the place or time referred to within it. Attributes include:

**exact** indicates the degree of accuracy associated with the distance.

*Legal values are:*

Y The distance is exact.

N The distance is approximate.

U Accuracy unavailable or unknown.

**<offset>** that part of a relative temporal or spatial expression which indicates the direction of the offset between the two place names, dates, or times involved in the expression.

As members of the class `temporalExpr` (temporal expression) these elements all share the following attributes:

**value** supplies the value of a date or time in a standard form.

**type** characterizes the element in some sense, using any convenient classification scheme or typology.

**reg** gives a normalized or regularized form of the name used.

#### 20.4.1 Absolute Dates and Times

An absolute temporal expression which is a date will contain only a sequence of `<day>`, `<month>`, `<week>`, `<year>` or `<occasion>` elements, as in the following examples:

```
The university's view of American affairs produced a stinging
attack by Edmund Burke in the Commons debate of
<dateStruct value="1775-10-26">
  <day value="26">26</day>
  <month value="10">October</month>
  <year value="1775">1775</year>
</dateStruct>
```

Component elements of a `<dateStruct>` may be repeated, provided that only a single temporal expression is intended:

```
<dateStruct value="1993-05-14">
  <day type="name">Friday</day>,
  <day type="number">14</day>
  <month>May</month>
  <year>1993</year>
</dateStruct>
```

The `<occasion>` element may be used for any component of a temporal expression which is given in terms of a named event, such as a public holiday for dates, or a named time such as “tea time” or “matins”:

```
In New York,
<dateStruct value="01-01">
  <occasion type="holiday">New Years Day</occasion>
</dateStruct> is the quietest of holidays,
<dateStruct value="07-04">
  <occasion type="holiday">Independence Day</occasion>
</dateStruct> the most turbulent.
```

These components may be applied to dates using any calendar system using subcomponents equivalent to those listed above:

```
<title>Le Vieux Cordelier:
Journal r&eacute;dig&eacute; par Camille Desmoulins</title>,
<dateStruct type="Revolutionary" value="1794-02-03">
  <day type="name">Quintidi</day>
  <month>Pluviose</month>
  <week>2e d&eacute;cade</week>,
  <year>l'an 2 de la R&eacute;publique Indivisible</year>
</dateStruct>
```

Absolute temporal expressions denoting times which are given in terms of seconds, minutes, hours or of well defined events (e.g. “noon”, “sunset”) may similarly be represented using the `<timeStruct>` element.

```
The train leaves for Boston at
<timeStruct type="24hour" zone="EST" value="18:45Z">
  <hour>13</hour>:<minute>45</minute>
</timeStruct>
At <timeStruct><occasion>sunset</occasion></timeStruct> we walked to the beach.
The train leaves for Boston at
<timeStruct type="descriptive" value="13:45" zone="EST">
  a quarter of <hour reg="1400">two</hour>
</timeStruct>
```

The type attribute may be used to distinguish sub-types of component elements (for example, months or days presented as words or as numbers) or to provide additional information about the function of this particular component (for example, to distinguish types of `<occasion>`). The value and reg attributes are both used to provide a standardized or regularized form of the content of an element. The distinction is that the value specified by the reg attribute is simply that chosen as a convenient way of grouping together a number of variant forms, whereas that specified for the value attribute should always be given in either an ISO 8601 form, or some application-dependent standard form described in the `<stdVals>` element of the TEI header.

For example:

```
<dateStruct value="1807-06-09">
  <month type="name" value="--06">June</month>
  <day type="number" value="---09">9th</day>
</dateStruct>: The period is approaching which will
terminate my present copartnership. On the
<dateStruct value="1808-01-01">
  <day type="number" value="---01">1st</day>
```

```
<month reg="January" type="name" value="--01">Jany.</month>
</dateStruct> next, it expires by its own limitation.
```

#### 20.4.2 Relative Dates and Times

As noted above, relative dates and times such as “in the Two Hundredth and First Year of the Republic”, “twenty minutes before noon”, and, more ambiguously, “after the lamented death of the Doctor” or “an hour after the game” have two distinct components. As well as the absolute temporal expression or event to which reference is made (e.g. “noon”, “the game”, “the death of the Doctor” “[the foundation of] the Republic”), they also contain a description of the ‘distance’ between the time or date which is indicated and the referent expression (e.g. “the Two Hundredth and First Year”, “twenty minutes”, “an hour”); and (optionally) an ‘offset’ describing the direction of the distance between the time or date indicated and the referent expression (e.g. “of” implying after, “before”, “after”).

The elements <distance> (or <measure>) and <offset> are used to encode these last two components within a <dateStruct> or <timeStruct>. The absolute temporal expression contained within the relative expression may be encoded using an <occasion> element, or by a nested <dateStruct> or <timeStruct>, or by a simple <date> or <time>. This allows for deeply nested structures such as “the third Sunday after the first Monday before Lammastide in the fifth year of the King’s second marriage ...” but so does natural language. In the following examples, the reg attribute has been used to simplify processing of variant forms of expression:

```
<dateStruct value="1786-12-11">
  <distance reg="14 days">A fortnight</distance>
  <offset>before</offset>
  <dateStruct>
    <occasion type="holiday">Christmas</occasion>
    <year>1786</year>
  </dateStruct>
</dateStruct>

I reached the station
<timeStruct value="14:15">
  <distance reg="30 minutes" exact="N">about a half hour</distance>
  <offset>after</offset>
  <occasion value="13:45">the departure of the afternoon train to Boston</occasion>
</timeStruct>
```

In the following example, the exact attribute has been used to indicate a lack of precision in the distance stated:

```
In practice, festival candles are lit
<timeStruct>
  <distance exact="N">just</distance>
  <offset>before</offset>
  <occasion reg="evening">sundown</occasion>
</timeStruct>
```

In the following example, a nested <dateStruct> element is used to show that “my birthday” and the cited date are parts of the same temporal expression, and hence to disambiguate the phrase “A week before my birthday on 9th December”:

```
<dateStruct value="12-02">
  <distance>A week</distance>
  <offset>before</offset>
  <dateStruct value="12-09">
    <occasion>my birthday</occasion>
    on <day>9th</day>
    <month>December</month>
  </dateStruct>
</dateStruct>
```

The alternative reading of this phrase would be encoded as follows:

```
<dateStruct value="09-02">
  <distance>A week</distance>
  <offset>before</offset>
```



```

    <occasion>my birthday</occasion>
    on <day>9th</day>
    <month>December</month>
  </dateStruct>

```

Where more complex or ambiguous expressions are involved, and where it is desirable to make more explicit the interpretive processes required, the feature structure notation described in chapter 16 *Feature Structures* is recommended. Consider, for example, the following temporal expression which occurs in the *Scottish Temperance Review* of August 1850, referring to the summer holiday known in Glasgow simply as “the Fair”:

```

    Not only is the city, <date ana="gf50">during the Fair</date>, a
    horrible nucleus of immorality and wickedness; it sends our
    multitudes to pollute and demoralize the country.

```

For the definition of the ana attribute, see chapter 15 *Simple Analytic Mechanisms*. It is used here to link the temporal phrase with an interpretation of it. Like most traditional fairs and market days, the Glasgow Fair was established by local custom and could vary from year to year. Consequently, in order to provide such an interpretation, it is necessary to draw upon additional information which may or may not be located in the particular text in question. In this case, it is necessary at least to know the spatial and temporal context (year and place) of the fair referred to. These and other features required for the analysis of this particular temporal expression may be combined together as one feature structure of type *date-analysis*:

```

  <fs id="gf50" type="date-analysis" rel="sb">
    <f name="event"><str>the Fair</str></f>
    <f name="place"><str>Glasgow</str></f>
    <f name="year"><nbr value="1850"/></f>
    <f name="from-value"><str>1850-08-08</str></f>
    <f name="to-value"><str>1850-09-19</str></f>
  </fs>

```

The elements described in this section are formally defined as follows:

```

<!-- 20.4.2: Date components-->
<!ELEMENT dateStruct %om.RR; (#PCDATA | %m.temporalExpr; | %m.Incl;)*>
<!ATTLIST dateStruct
  %a.global;
  %a.temporalExpr;
  calendar CDATA #IMPLIED
  exact CDATA #IMPLIED
  TEIform CDATA 'dateStruct' >
<!ELEMENT day %om.RR; (#PCDATA)>
<!ATTLIST day
  %a.global;
  %a.temporalExpr;
  TEIform CDATA 'day' >
<!ELEMENT week %om.RR; (#PCDATA)>
<!ATTLIST week
  %a.global;
  %a.temporalExpr;
  TEIform CDATA 'week' >
<!ELEMENT month %om.RR; (#PCDATA)>
<!ATTLIST month
  %a.global;
  %a.temporalExpr;
  TEIform CDATA 'month' >
<!ELEMENT year %om.RR; (#PCDATA)>
<!ATTLIST year
  %a.global;
  %a.temporalExpr;
  TEIform CDATA 'year' >
<!ELEMENT occasion %om.RR; %phrase.seq;>
<!ATTLIST occasion
  %a.global;
  %a.temporalExpr;

```

## 20 Names and Dates

```
      TEIform CDATA 'occasion' >
<!ELEMENT timeStruct %om.RR; (#PCDATA | %m.temporalExpr; | %m.Incl;)*>
<!ATTLIST timeStruct
      %a.global;
      %a.temporalExpr;
      zone CDATA #IMPLIED
      TEIform CDATA 'timeStruct' >
<!ELEMENT second %om.RR; (#PCDATA)>
<!ATTLIST second
      %a.global;
      %a.temporalExpr;
      TEIform CDATA 'second' >
<!ELEMENT minute %om.RR; (#PCDATA)>
<!ATTLIST minute
      %a.global;
      %a.temporalExpr;
      TEIform CDATA 'minute' >
<!ELEMENT hour %om.RR; (#PCDATA)>
<!ATTLIST hour
      %a.global;
      %a.temporalExpr;
      TEIform CDATA 'hour' >
<!--offset and distance were defined above-->
<!-- end of 20.4.2-->
```