

34 Entities

component (component for dictionaries) defines the set of component-level elements for dictionaries; these are elements which can appear directly within text bodies or text divisions.

Declaration <!ENTITY % component '(%m.common; | %m.comp.dictionaries;)' >
See further 3.7 Element Classes

component (component for drama) defines the set of component-level elements for drama; these are elements which can appear directly within text bodies or text divisions.

Declaration <!ENTITY % component '(%m.common; | %m.comp.drama;)' >
See further 3.7 Element Classes

component (component for mixed or general base) defines the set of component-level elements for use with the mixed or general base; these are elements which can appear directly within text bodies or text divisions.

Declaration <!ENTITY % component '(%m.common; %mix.verse; %mix.drama; %mix.spoken; %mix.dictionaries; %mix.terminology;)' >
See further 3.7 Element Classes

component (component for prose) defines the set of component-level elements for prose; these are elements which can appear directly within text bodies or text divisions.

Declaration <!ENTITY % component '(%m.common;)' >
See further 3.7 Element Classes

component (component for transcription of spoken texts) defines the set of component-level elements for spoken texts; these are elements which can appear directly within text bodies or text divisions.

Declaration <!ENTITY % component '(%m.common; | %m.comp.spoken;)' >
See further 3.7 Element Classes

component (component for terminology) defines the set of component-level elements for terminology; these are elements which can appear directly within text bodies or text divisions.

Declaration <!ENTITY % component '(%m.common; | %m.comp.terminology;)' >
See further 3.7 Element Classes

component (component for verse) defines the set of component-level elements for verse; these are elements which can appear directly within text bodies or text divisions.

Declaration <!ENTITY % component '(%m.common; | %m.comp.verse;)' >
See further 3.7 Element Classes

component .plus (component sequence) defines a sequence of components as needed in the general base tag set, allowing components from any base to be used, but preventing their mixing.

Note When used in a content model, this entity requires at least one component-level element to occur.
Declaration <!ENTITY % component.plus '(%gen.verse; %gen.drama; %gen.spoken; %gen.dictionaries; %gen.terminology; TEI...end) | ((%m.common;)+, (%gen.verse; %gen.drama; %gen.spoken; %gen.dictionaries; %gen.terminology; TEI...end)?)' >
See further 3.7 Element Classes

component.seq (component sequence for general combined base) defines a sequence of components as needed in the general base tag set, allowing components from any base to be used, but preventing their mixing.

Note When used in a content model, this entity, like other entities with names of the form “x.seq”, allows zero or more occurrences of its content to occur.

Declaration `<!ENTITY % component.seq '(%m.common;)*, (%gen.verse; %gen.drama; %gen.spoken; %gen.dictionaries; %gen.terminology; TEI...end)?' >`

See further 3.7 *Element Classes*

component.seq (component-sequence) defines a sequence of component-level elements (such as paragraphs or lists) which can occur directly within text divisions and in similar positions.

Note This parameter entity is used in each base tag set to define the content of `<div>` and similar elements.

Declaration `<!ENTITY % component.seq '((%component;), (%m.Incl;))*' >`

See further 3.7 *Element Classes*

extPtr (extended-pointer expression) used as the declared value of an attribute, indicates that all values of that attribute must be valid expressions in the TEI extended pointer notation defined in section 14.2.1 *Extended Pointer Elements*.

Declaration `<!ENTITY % extPtr 'CDATA' >`

See further 14.2.1 *Extended Pointer Elements*

formulaContent defines the content model for the `<formula>` element.

Declaration `<!ENTITY % formulaContent '(#PCDATA)' >`

See further 22.2 *Formulae and Mathematical Expressions*

formulaNotations specifies the set of notations which may be used for the `<formula>` element.

Note This will normally be defined either as CDATA (the default), or as a string such as NOTATION (tex | eqn)

Declaration `<!ENTITY % formulaNotations 'CDATA' >`

See further 22.2 *Formulae and Mathematical Expressions*

gen.dictionaries (dictionary part of general-base component sequence) contains a string used in constructing the definition of component sequence used in the general base tag set.

Declaration `<!ENTITY % gen.dictionaries '((%m.comp.dictionaries;), (%m.common; | %m.comp.dictionaries;))* |' >`

See further 3.7 *Element Classes*

gen.dictionaries (dictionary part of general-base component sequence) contains a string used in constructing the definition of component sequence used in the general base tag set.

Note This version of this entity is used if the dictionary base is not selected.

Declaration `<!ENTITY % gen.dictionaries '' >`

See further 3.7 *Element Classes*

gen.drama (drama part of general-base component sequence) contains a string used in constructing the definition of component sequence used in the general base tag set.

Declaration `<!ENTITY % gen.drama '((%m.comp.drama;), (%m.common; | %m.comp.drama;))* |' >`

See further 3.7 Element Classes

gen.drama (drama part of general-base component sequence) contains a string used in constructing the definition of component sequence used in the general base tag set.

Note This version of this entity is used if the drama base is not selected.

Declaration <!ENTITY % gen.drama '' >

See further 3.7 Element Classes

gen.spoken (spoken-text part of general-base component sequence) contains a string used in constructing the definition of component sequence used in the general base tag set.

Declaration <!ENTITY % gen.spoken '((%m.comp.spoken;), (%m.common; | %m.comp.spoken;)*)' >

See further 3.7 Element Classes

gen.spoken (spoken-text part of general-base component sequence) contains a string used in constructing the definition of component sequence used in the general base tag set.

Note This version of this entity is used if the spoken base is not selected.

Declaration <!ENTITY % gen.spoken '' >

See further 3.7 Element Classes

gen.terminology (terminology part of general-base component sequence) contains a string used in constructing the definition of component sequence used in the general base tag set.

Declaration <!ENTITY % gen.terminology '((%m.comp.terminology;), (%m.common; | %m.comp.terminology;)*)' >

See further 3.7 Element Classes

gen.terminology (terminology part of general-base component sequence) contains a string used in constructing the definition of component sequence used in the general base tag set.

Note This version of this entity is used if the terminology base is not selected.

Declaration <!ENTITY % gen.terminology '' >

See further 3.7 Element Classes

gen.verse (verse part of general-base component sequence) contains a string used in constructing the definition of component sequence used in the general base tag set.

Declaration <!ENTITY % gen.verse '((%m.comp.verse;), (%m.common; | %m.comp.verse;)*)' >

See further 3.7 Element Classes

gen.verse (verse part of general-base component sequence) contains a string used in constructing the definition of component sequence used in the general base tag set.

Note This version of this entity is used if the verse base is not selected.

Declaration <!ENTITY % gen.verse '' >

See further 3.7 Element Classes

INHERITED as a default attribute value, indicates that if not specified, the attribute value is inherited from corresponding attribute of the parent element.

Note For the parser, this entity has the same effect as specifying a default value as IMPLIED. For the user and the application program, however, INHERITED has a different effect, specifying as it does exactly how the application program is to infer the correct default value: the value is to

be inherited from the attribute of the same name on the immediately enclosing element. If that element, too, specifies no value, then its value will have been inherited from its own immediately enclosing element, etc. (If the attribute is not declared for all elements, the value is inherited from the nearest ancestor for which the attribute is declared.) If no ancestor element has a value specified for the attribute, the value is undefined. Encoders are encouraged to provide an explicit value for inherited attributes on the outermost elements for which they are declared; it is, however, not an error for the outermost element to specify no attribute value for an attribute with a default of %INHERITED;. The most prominent example of attribute value inheritance is the TEI global attribute lang.

Declaration <!ENTITY % INHERITED '#IMPLIED' >

See further 3.7 Element Classes

ISO-date as the declared value of an attribute, indicates that the attribute value should be a legal ISO date in the format defined by ISO 8601:2000(E) *Data elements and interchange formats — Information interchange — Representation of dates and times*, e.g. yyyy-mm-dd.

Note For the parser, this entity has the same effect as specifying a declared value of “CDATA”. For the user and the application program, however, “%ISO-date;” documents an additional restriction on the legal content for this element. The most prominent examples of this declared value type are the value attribute of the <docDate> element in the core tag set, and the date attribute of the <admin> element in the base tag set for terminology.

Declaration <!ENTITY % ISO-date 'CDATA' >

See further 3.8.3 Parameter Entities for TEI Keywords

mix.dictionaries (mixed-base dictionary components) contains a string used in constructing the definition of *component* used in the mixed base tag set.

Declaration <!ENTITY % mix.dictionaries '| %m.comp.dictionaries;' >

See further 3.7 Element Classes

mix.dictionaries (mixed-base dictionaries components) default declaration of a string used in constructing the definition of *component* used in the mixed base tag set.

Declaration <!ENTITY % mix.dictionaries '' >

See further 3.7.8 Components in Mixed and General Bases

mix.drama (mixed-base drama components) contains a string used in constructing the definition of *component* used in the mixed base tag set.

Declaration <!ENTITY % mix.drama '| %m.comp.drama;' >

See further 3.7 Element Classes

mix.drama (mixed-base drama components) default declaration of a string used in constructing the definition of *component* used in the mixed base tag set.

Declaration <!ENTITY % mix.drama '' >

See further 3.7.8 Components in Mixed and General Bases

mix.spoken (mixed-base spoken-text components) contains a string used in constructing the definition of *component* used in the mixed base tag set.

Declaration <!ENTITY % mix.spoken '| %m.comp.spoken;' >

See further 3.7 Element Classes

`mix.spoken` (mixed-base spoken components) default declaration of a string used in constructing the definition of *component* used in the mixed base tag set.

Declaration `<!ENTITY % mix.spoken '' >`

See further 3.7.8 *Components in Mixed and General Bases*

`mix.terminology` (mixed-base terminology components) contains a string used in constructing the definition of *component* used in the mixed base tag set.

Declaration `<!ENTITY % mix.terminology '| %m.comp.terminology;' >`

See further 3.7 *Element Classes*

`mix.terminology` (mixed-base terminology components) default declaration of a string used in constructing the definition of *component* used in the mixed base tag set.

Declaration `<!ENTITY % mix.terminology '' >`

See further 3.7.8 *Components in Mixed and General Bases*

`mix.verse` (mixed-base verse components) contains a string used in constructing the definition of *component* used in the mixed base tag set.

Declaration `<!ENTITY % mix.verse '| %m.comp.verse;' >`

See further 3.7 *Element Classes*

`mix.verse` (mixed-base verse components) default declaration of a string used in constructing the definition of *component* used in the mixed base tag set.

Declaration `<!ENTITY % mix.verse '' >`

See further 3.7.8 *Components in Mixed and General Bases*

`om.RO` (omissibility indicator (SGML version)) Used in the declaration of an SGML element to indicate that the end-tag but not the start-tag may be omitted.

Note This parameter entity is redefined as a null string when an XML DTD is requested.

Declaration `<!ENTITY % om.RO '- 0' >`

See further 3.8.4 *Generation of an XML DTD*

`om.RR` (omissibility indicator (SGML version)) Used in the declaration of an SGML element to indicate that neither the end-tag nor the start-tag may be omitted.

Note This parameter entity is redefined as a null string when an XML DTD is requested.

Declaration `<!ENTITY % om.RR '- -' >`

See further 3.8.4 *Generation of an XML DTD*

`om.RO` (omissibility indicator (XML version)) A dummy component of an XML element declaration, retained for SGML compatibility.

Note Omissibility is not allowed for XML element declarations. This parameter entity therefore has a non-null value only when an SGML DTD is requested.

Declaration `<!ENTITY % om.RO '' >`

See further 3.8.4 *Generation of an XML DTD*

om.RR (omissibility indicator (XML version)) A dummy component of an XML element declaration, retained for SGML compatibility.

Note Omissibility is not allowed for XML element declarations. This parameter entity therefore has a non-null value only when an SGML DTD is requested.

Declaration `<!ENTITY % om.RR '' >`

See further 3.8.4 *Generation of an XML DTD*

paraContent (paragraph content) defines the legal version for paragraphs and similar elements.

Declaration `<!ENTITY % paraContent '(#PCDATA | %m.phrase; | %m.inter; | %m.Incl;)*' >`

See further 3.7 *Element Classes*

phrase defines a phrase as character data or any phrase-level element.

Note This entity is used in the declaration of *phrase.seq*.

Declaration `<!ENTITY % phrase '#PCDATA | %m.phrase; | %m.Incl; ' >`

See further 3.7 *Element Classes*

phrase.seq (phrase sequence) defines a sequence of character data and phrase-level elements.

Declaration `<!ENTITY % phrase.seq '(%phrase;)*' >`

See further 3.7.7 *Standard Content Models*

seq (sequence) defines a sequence of elements (such as paragraphs) which can occur directly within text divisions and in similar positions.

Note This parameter entity is used in each base tag set to define the content of `<div>` and other elements.

Declaration `<!ENTITY % seq '%m.common; | %m.comp.terminology;)*' >`

See further 13.4 *Overall Structure of Terminological Documents*

specialPara ('special' paragraph content) defines the content model of elements such as notes or list items, which either contain a series of component-level elements or else have the same structure as a paragraph, containing a series of phrase-level and inter-level elements.

Declaration `<!ENTITY % specialPara '(#PCDATA | %m.phrase; | %m.inter; | %m.chunk; | %m.Incl;)*' >`

See further 3.7 *Element Classes*

TEI.analysis (TEI simple analytic mechanisms DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for simple analytic mechanisms.

Note To include elements and attributes for simple analytic mechanisms in the DTD, the user should declare this entity with a value of "INCLUDE"; this will override the default.

Declaration `<!ENTITY % TEI.analysis 'IGNORE' >`

See further 3.6 *The TEI2.DTD File*

TEI.analysis.dtd (TEI analysis-base DTD) identifies the file containing element and attribute list declarations for the base tag set for simple analysis.

Note This entity is declared with this value when the user includes the base tag set for simple analysis.

Declaration `<!ENTITY % TEI.analysis.dtd PUBLIC '-//TEI P4//ELEMENTS Additional Element Set for Simple Analysis//EN' 'teiana2.dtd' >`

See further 3.2 *Core, Base, and Additional Tag Sets*

TEI.analysis.ent (TEI analysis-base entities) identifies the file containing parameter entity declarations for the element classes defined in the base tag set for simple analysis.

Note This entity is declared with this value when the user includes the base tag set for simple analysis.

Declaration `<!ENTITY % TEI.analysis.ent PUBLIC '-//TEI P4//ENTITIES Element Classes for Simple Analysis//EN' 'teiana2.ent' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.back.dtd defines the file in which tags for back matter are defined.

Note This parameter entity is used in each base tag set to include the back-matter tags, which are common to all bases.

Declaration `<!ENTITY % TEI.back.dtd PUBLIC '-//TEI P4//ELEMENTS Back Matter//EN' 'teiback2.dtd' >`

See further 7.6 Back Matter

TEI.certainty (TEI certainty DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for certainty.

Note To include elements and attributes for certainty in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.certainty 'IGNORE' >`

See further 3.5 Global Attributes; 3.6 The TEI2.DTD File

TEI.certainty.dtd (TEI certainty-base DTD) identifies the file containing element and attribute list declarations for the base tag set for certainty and uncertainty.

Note This entity is declared with this value when the user includes the base tag set for certainty and uncertainty.

Declaration `<!ENTITY % TEI.certainty.dtd PUBLIC '-//TEI P4//ELEMENTS Additional Element Set for Certainty and Responsibility//EN' 'teicert2.dtd' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.core.dtd (TEI core) identifies the file containing element and attribute list declarations for the TEI core elements.

Note This entity is included in all TEI DTDs.

Declaration `<!ENTITY % TEI.core.dtd PUBLIC '-//TEI P4//ELEMENTS Core Elements//EN' 'teicore2.dtd' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.corpus (TEI corpora DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for the description of corpora.

Note To include elements and attributes for corpora in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.corpus 'IGNORE' >`

See further 3.5 Global Attributes; 3.6 The TEI2.DTD File

TEI.corpus.dtd (TEI corpus-base DTD) identifies the file containing element and attribute list declarations for the base tag set for corpora.

Note This entity is declared with this value when the user includes the base tag set for corpora.

Declaration `<!ENTITY % TEI.corpus.dtd PUBLIC '-//TEI P4//ELEMENTS Additional Element Set for Language Corpora//EN' 'teicorp2.dtd' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.dictionaries (TEI dictionaries DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for dictionaries.

Note To include elements and attributes for dictionaries in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.dictionaries 'IGNORE' >`

See further 3.5 Global Attributes; 3.6 The TEI2.DTD File

TEI.dictionaries.dtd (TEI dictionary-base DTD) identifies the file containing element and attribute list declarations for the base tag set for dictionaries.

Note This entity is declared with this value when the user includes the base tag set for dictionaries.

Declaration `<!ENTITY % TEI.dictionaries.dtd PUBLIC '-//TEI P4//ELEMENTS Base Element Set for Print Dictionaries//EN' 'teidict2.dtd' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.dictionaries.ent (TEI dictionary-base entities) identifies the file containing parameter entity declarations for the element classes defined in the base tag set for dictionaries.

Note This entity is declared with this value when the user includes the base tag set for dictionaries.

Declaration `<!ENTITY % TEI.dictionaries.ent PUBLIC '-//TEI P4//ENTITIES Element Classes for Print Dictionaries//EN' 'teidict2.ent' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.drama (TEI drama DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for drama.

Note To include elements and attributes for drama in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.drama 'IGNORE' >`

See further 3.5 Global Attributes; 3.6 The TEI2.DTD File

TEI.drama.dtd (TEI drama-base DTD) identifies the file containing element and attribute list declarations for the base tag set for drama.

Note This entity is declared with this value when the user includes the base tag set for drama.

Declaration `<!ENTITY % TEI.drama.dtd PUBLIC '-//TEI P4//ELEMENTS Base Element Set for Drama 2001-12//EN' 'teidram2.dtd' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.drama.ent (TEI drama-base entities) identifies the file containing parameter entity declarations for the element classes defined in the base tag set for drama.

Note This entity is declared with this value when the user includes the base tag set for drama.

Declaration `<!ENTITY % TEI.drama.ent PUBLIC '-//TEI P4//ENTITIES Element Classes for Drama//EN' 'teidram2.ent' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.elementClasses file containing the parameter entity declarations which define element classes for content models, attributes shared among elements of a class, and global attributes.

Declaration `<!ENTITY % TEI.elementClasses PUBLIC '-//TEI P4//ENTITIES TEI ElementClasses//EN' 'teiclas2.ent' >`

See further 3.6 *The TEI2.DTD File*; 3.7 *Element Classes*

TEI.elementNames file containing parameter entity declarations for all generic identifiers of the encoding scheme.

Note The parameter entities in this file all take the same form as the two shown below: `<!ENTITY % n.div0 'div0' >`

`<!ENTITY % n.div1 'div1' >`

Element and attribute-list declarations in the DTDs refer to the parameter entity `n.div1`, not directly to the generic identifier `<div1>`. As a result, the declarations will function as desired even if a new generic identifier is substituted. E.g. `<caput>` for `<div1>` and `<liber>` for `<div0>`: `<!ENTITY % n.div0 'liber' >` `<!ENTITY % n.div1 'caput' >` This allows generic identifiers to be renamed conveniently, e.g., to provide names in languages other than English, or to provide shorter names than those documented here. See further chapter 29 *Modifying and Customizing the TEI DTD*.

Declaration `<!ENTITY % TEI.elementNames PUBLIC '-//TEI P4//ENTITIES Generic Identifiers//EN' 'teigis2.ent' >`

See further 3.8.2 *Parameter Entities for Element Generic Identifiers*

TEI.extensions.dtd file (if any) containing local modifications to the TEI DTDs.

Note This entity is embedded in the TEI DTDs after the TEI element classes are embedded, and immediately before the base tag set and additional tag sets selected by the user are embedded. By default, the entity expands to the empty string; the user can override this default by declaring the entity with an appropriate value, typically this will take the form `<!ENTITY % TEI.extensions.dtd SYSTEM 'project.dtd' >`

Declaration `<!ENTITY % TEI.extensions.dtd '' >`

See further 3.6 *The TEI2.DTD File*

TEI.extensions.ent file (if any) containing local modifications to the TEI element classes.

Note This entity is embedded in the TEI DTDs before the TEI element classes are embedded. By default, the entity expands to the empty string; the user can override this default by declaring the entity with an appropriate value, typically this will take the form `<!ENTITY % TEI.extensions.ent SYSTEM 'project.ent' >`

Declaration `<!ENTITY % TEI.extensions.ent '' >`

See further 3.6 *The TEI2.DTD File*

TEI.figures (TEI figures and tables DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for figures, formulae, and tables.

Note To include elements and attributes for figures, formulae, and tables in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.figures 'IGNORE' >`

See further 3.5 *Global Attributes*; 3.6 *The TEI2.DTD File*

TEI.figures.dtd (TEI figures-base DTD) identifies the file containing element and attribute list declarations for the base tag set for figures, tables, and formulae.

Note This entity is declared with this value when the user includes the base tag set for figures, formulae and tables.

Declaration `<!ENTITY % TEI.figures.dtd PUBLIC '-//TEI P4//ELEMENTS Additional Element Set for Tables, Formulae, and Graphics//EN' 'teifig2.dtd' >`

See further 3.2 *Core, Base, and Additional Tag Sets*

TEI.figures.ent (TEI figures-base entities) identifies the file containing parameter entity declarations for the element classes defined in the additional tag set for figures.

Note This entity is declared with this value when the user includes the additional tag set for figures, tables, and formulae.

Declaration `<!ENTITY % TEI.figures.ent PUBLIC "-//TEI P4//ENTITIES Formulae Notations and Contents//EN" 'teifig2.ent' >`

See further 3.7 Element Classes

TEI.front.dtd defines the file in which tags for front matter are defined.

Note This parameter entity is used in each base tag set to include the front-matter tags, which are common to all bases.

Declaration `<!ENTITY % TEI.front.dtd PUBLIC "-//TEI P4//ELEMENTS Front Matter//EN" 'teifron2.dtd' >`

See further 7.4 Front Matter

TEI.fs (TEI feature structures DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for feature structures.

Note To include elements and attributes for feature structures in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.fs 'IGNORE' >`

See further 3.6 The TEI2.DTD File

TEI.fs.dtd (TEI fs-base DTD) identifies the file containing element and attribute list declarations for the base tag set for feature structures.

Note This entity is declared with this value when the user includes the base tag set for feature structures.

Declaration `<!ENTITY % TEI.fs.dtd PUBLIC "-//TEI P4//DTD Auxiliary Document Type: Feature System Declaration//EN" 'teifs2.dtd' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.general (TEI general base DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for the ‘general’ combined base.

Note To include elements and attributes for the ‘general’ base in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.general 'IGNORE' >`

See further 3.6 The TEI2.DTD File

TEI.general.dtd (TEI general-base DTD) identifies the file containing element and attribute list declarations for the base tag set for the ‘general’ base tag set.

Note This entity is declared with this value when the user includes the ‘general’ mixed-base tag set.

Declaration `<!ENTITY % TEI.general.dtd PUBLIC "-//TEI P4//ELEMENTS General Base Element Set//EN" 'teigen2.dtd' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.header.dtd (TEI header) identifies the file containing element and attribute list declarations for the TEI header.

Note This entity is included in all TEI DTDs.

Declaration <!ENTITY % TEI.header.dtd PUBLIC "-//TEI P4//ELEMENTS TEI Header//EN" 'teihdr2.dtd' >
See further 3.2 Core, Base, and Additional Tag Sets

TEI.keywords.ent file containing the parameter entity declarations for TEI keywords.

Note The keywords defined in this file:

- define non-SGML, non-XML data types (e.g. ISO dates);
- control the selection of base and additional tag sets; and
- control whether the DTDs use SGML or XML syntax.

Declaration <!ENTITY % TEI.keywords.ent PUBLIC "-//TEI P4//ENTITIES TEI Keywords//EN" 'teikey2.ent' >

See further 3.8.3 Parameter Entities for TEI Keywords

TEI.linking (TEI DTD fragment for linking tag set) controls the inclusion, in the DTD, of element and attribute declarations for linking, segmentation and alignment.

Note To include elements and attributes for segmentation and alignment in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration <!ENTITY % TEI.linking 'IGNORE' >

See further 3.6 The TEI2.DTD File

TEI.linking.dtd (TEI segmentation and alignment elements) identifies the file containing element and attribute list declarations for the additional tag set for segmentation and alignment.

Note This entity is declared with this value when the user includes the additional tag set for segmentation and alignment.

Declaration <!ENTITY % TEI.linking.dtd PUBLIC "-//TEI P4//ELEMENTS Additional Element Set for Linking, Segmentation, and Alignment//EN" 'teilink2.dtd' >

See further 3.2 Core, Base, and Additional Tag Sets

TEI.linking.ent (TEI linking-tag-set entities) identifies the file containing parameter entity declarations for the element classes defined in the additional tag set for segmentation and alignment.

Note This entity is declared with this value when the user includes the additional tag set for segmentation and alignment.

Declaration <!ENTITY % TEI.linking.ent PUBLIC "-//TEI P4//ENTITIES Element Classes for Linking, Segmentation, and Alignment//EN" 'teilink2.ent' >

See further 3.2 Core, Base, and Additional Tag Sets

TEI.mixed (TEI ‘mixed’ base DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for the ‘mixed’ combined base.

Note To include elements and attributes for the ‘mixed’ base in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration <!ENTITY % TEI.mixed 'IGNORE' >

See further 3.6 The TEI2.DTD File

TEI.mixed.dtd (TEI mixed-base DTD) identifies the file containing element and attribute list declarations for the base tag set for the ‘mixed’ base tag set.

Note This entity is declared with this value when the user includes the base tag set for the ‘mixed’ base.

Declaration `<!ENTITY % TEI.mixed.dtd PUBLIC '-//TEI P4//ELEMENTS Base Element Set for Mixed Text Types//EN' 'teimix2.dtd' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.names.dates (TEI names and dates DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for detailed analysis of names and dates.

Note To include elements and attributes for names and dates in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.names.dates 'IGNORE' >`

See further 3.5 Global Attributes; 3.6 The TEI2.DTD File

TEI.names.dates.dtd (TEI names.dates-base DTD) identifies the file containing element and attribute list declarations for the additional tag set for detailed analysis of names and dates.

Note This entity is declared with this value when the user includes the additional tag set for names and dates.

Declaration `<!ENTITY % TEI.names.dates.dtd PUBLIC '-//TEI P4//ELEMENTS Additional Element Set for Names and Dates//EN' 'teind2.dtd' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.names.dates.ent (parameter entities for names and dates) identifies the file containing parameter entity declarations for the element classes defined in the additional tag set for names and dates.

Note This entity is declared with this value when the user includes the additional tag set for names and dates.

Declaration `<!ENTITY % TEI.names.dates.ent PUBLIC '-//TEI P4//ENTITIES Element Classes for Names and Dates//EN' 'teind2.ent' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.nets (TEI graph theoretic DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for graph theory (graphs, digraphs, and other networks)

Note To include elements and attributes for graphs and digraphs in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.nets 'IGNORE' >`

See further 3.5 Global Attributes; 3.6 The TEI2.DTD File

TEI.nets.dtd (TEI networks-base DTD) identifies the file containing element and attribute list declarations for the additional tag set for graph theory (graphs, digraphs, and other networks).

Note This entity is declared with this value when the user includes the additional tag set for graph theory.

Declaration `<!ENTITY % TEI.nets.dtd PUBLIC '-//TEI P4//ELEMENTS Additional Element Set for Graphs, Networks, and Trees//EN' 'teinet2.dtd' >`

See further 3.2 Core, Base, and Additional Tag Sets

TEI.prose (TEI prose DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for prose.

Note To include elements and attributes for prose in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.prose 'IGNORE' >`

See further 3.2 Core, Base, and Additional Tag Sets; 3.6 The TEI2.DTD File

TEI.prose.dtd (TEI prose-base DTD) identifies the file containing element and attribute list declarations for the base tag set for prose.

Note This entity is declared with this value when the user includes the base tag set for prose.

Declaration <!ENTITY % TEI.prose.dtd PUBLIC "-//TEI P4//ELEMENTS Base Element Set for Prose//EN" 'teipros2.dtd' >

See further 3.2 Core, Base, and Additional Tag Sets

TEI.singleBase (single-base flag) controls the inclusion, in the DTD, of the default text structure elements. It is declared with the value IGNORE when the mixed base or general base is selected.

Note The user should not redefine this parameter entity.

Declaration <!ENTITY % TEI.singleBase 'IGNORE' >

See further 3.8 Other Parameter Entities in TEI DTDs

TEI.singleBase (single-base flag) controls the inclusion, in the DTD, of the default text structure elements. It is declared with the value INCLUDE when only a single base tag set is selected.

Note The user should not redefine this parameter entity.

Declaration <!ENTITY % TEI.singleBase 'INCLUDE' >

See further 3.8 Other Parameter Entities in TEI DTDs

TEI.spoken (TEI spoken texts DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for spoken texts.

Note To include elements and attributes for spoken texts in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration <!ENTITY % TEI.spoken 'IGNORE' >

See further 3.5 Global Attributes; 3.6 The TEI2.DTD File

TEI.spoken.dtd (TEI spoken-base DTD) identifies the file containing element and attribute list declarations for the base tag set for spoken texts.

Note This entity is declared with this value when the user includes the base tag set for spoken texts.

Declaration <!ENTITY % TEI.spoken.dtd PUBLIC "-//TEI P4//ELEMENTS Base Element Set for Transcriptions of Speech//EN" 'teispok2.dtd' >

See further 3.2 Core, Base, and Additional Tag Sets

TEI.spoken.ent (TEI spoken-base entities) identifies the file containing parameter entity declarations for the element classes defined in the base tag set for spoken texts.

Note This entity is declared with this value when the user includes the base tag set for spoken texts.

Declaration <!ENTITY % TEI.spoken.ent PUBLIC "-//TEI P4//ENTITIES Element Classes for Transcriptions of Speech//EN" 'teispok2.ent' >

See further 3.2 Core, Base, and Additional Tag Sets

TEI.structure.dtd defines the file in which the default text structure used by many base tag sets is defined.

Declaration <!ENTITY % TEI.structure.dtd PUBLIC "-//TEI P4//ELEMENTS Default Text Structure//EN" 'teistr2.dtd' >

See further 3.2 Core, Base, and Additional Tag Sets

TEI.terminology (TEI terminological data DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for terminological data.

Note To include elements and attributes for terminological data in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.terminology 'IGNORE' >`

See further 3.6 *The TEI2.DTD File*

TEI.terminology.dtd (TEI terminology-base DTD) identifies the file containing element and attribute list declarations for the base tag set for terminological data.

Note This entity is declared with this value when the user includes the base tag set for terminological data.

Declaration `<!ENTITY % TEI.terminology.dtd PUBLIC '-//TEI P4//ELEMENTS Base Element Set for Terminological Data//EN' 'teiterm2.dtd' >`

See further 3.2 *Core, Base, and Additional Tag Sets*

TEI.terminology.ent (TEI terminology-base entities) identifies the file containing parameter entity declarations for the element classes defined in the base tag set for terminological data.

Note This entity is declared with this value when the user includes the base tag set for terminological data.

Declaration `<!ENTITY % TEI.terminology.ent PUBLIC '-//TEI P4//ENTITIES Element Classes for Terminological Data//EN' 'teiterm2.ent' >`

See further 3.2 *Core, Base, and Additional Tag Sets*

TEI.textcrit (TEI text criticism DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for text criticism.

Note To include elements and attributes for text criticism in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.textcrit 'IGNORE' >`

See further 3.6 *The TEI2.DTD File*

TEI.textcrit.dtd (TEI text criticism base DTD) identifies the file containing element and attribute list declarations for the additional tag set for text criticism.

Note This entity is declared with this value when the user includes the additional tag set for text criticism.

Declaration `<!ENTITY % TEI.textcrit.dtd PUBLIC '-//TEI P4//ELEMENTS Additional Element Set for Text-Critical Apparatus//EN' 'teitc2.dtd' >`

See further 3.2 *Core, Base, and Additional Tag Sets*

TEI.textcrit.ent (TEI text criticism entities) identifies the file containing parameter entity declarations for the element classes defined in the additional tag set for text criticism.

Note This entity is declared with this value when the user includes the additional tag set for text criticism.

Declaration `<!ENTITY % TEI.textcrit.ent PUBLIC '-//TEI P4//ENTITIES Element Classes for Critical Apparatus//EN' 'teitc2.ent' >`

See further 3.2 *Core, Base, and Additional Tag Sets*

TEI.transcr (TEI DTD fragment for transcription of primary sources) controls the inclusion, in the DTD, of element and attribute declarations for transcription of the source text.

Note To include elements and attributes for transcription in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration <!ENTITY % TEI.transcr 'IGNORE' >

See further 3.5 Global Attributes; 3.6 The TEI2.DTD File

TEI.transcr.dtd (TEI DTD for transcription of primary sources) identifies the file containing element and attribute list declarations for the base tag set for description of the source text.

Note This entity is declared with this value when the user includes the base tag set for transcription of primary sources

Declaration <!ENTITY % TEI.transcr.dtd PUBLIC '-//TEI P4//ELEMENTS Additional Element Set for Transcription of Primary Sources//EN' 'teitran2.dtd' >

See further 3.2 Core, Base, and Additional Tag Sets

TEI.transcr.ent (TEI entities for transcription of primary sources) identifies the file containing parameter entity declarations for the element classes defined in the additional tag set for transcription of primary sources.

Note This entity is declared with this value when the user includes the additional tag set for transcription of primary sources.

Declaration <!ENTITY % TEI.transcr.ent PUBLIC '-//TEI P4//ENTITIES Element Classes for Transcription of Primary Sources//EN' 'teitran2.ent' >

See further 3.7 Element Classes

TEI.verse (TEI verse DTD fragment) controls the inclusion, in the DTD, of element and attribute declarations for verse.

Note To include elements and attributes for verse in the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration <!ENTITY % TEI.verse 'IGNORE' >

See further 3.2 Core, Base, and Additional Tag Sets; 3.6 The TEI2.DTD File

TEI.verse.dtd (TEI verse-base DTD) identifies the file containing element and attribute list declarations for the base tag set for verse.

Note This entity is declared with this value when the user includes the base tag set for verse.

Declaration <!ENTITY % TEI.verse.dtd PUBLIC '-//TEI P4//ELEMENTS Base Element Set for Verse//EN' 'teivers2.dtd' >

See further 3.2 Core, Base, and Additional Tag Sets

TEI.verse.ent (TEI verse-base entities) identifies the file containing parameter entity declarations for the element classes defined in the base tag set for verse.

Note This entity is declared with this value when the user includes the base tag set for verse.

Declaration <!ENTITY % TEI.verse.ent PUBLIC '-//TEI P4//ENTITIES Element Classes for Verse//EN' 'teivers2.ent' >

See further 3.2 Core, Base, and Additional Tag Sets

TEI.wsdNames file containing parameter entity declarations for all generic identifiers used in the writing system declaration

Note The parameter entities in this file all take the same form as the two shown below: <!EN-

```
TITY % n.figure 'figure' >
```

```
<!ENTITY % n.character 'character' >
```

Element and attribute-list declarations in the DTDs refer to the parameter entity `n.character`, not directly to the generic identifier `<character>`. As a result, the declarations will function as desired even if a new generic identifier is substituted. E.g. `<char>` for `<character>` and `<wsdFig>` for `<figure>`: `<!ENTITY % n.figure "wsdFig" >`

```
<!ENTITY % n.character "char" >
```

This allows generic identifiers to be renamed conveniently, e.g., to provide names in languages other than English, or to provide shorter names than those documented here. See further chapter 29 *Modifying and Customizing the TEI DTD*.

Declaration `<!ENTITY % TEI.wsdNames PUBLIC "-//TEI P4//ENTITIES Generic Identifiers for Writing System Declaration//EN" 'wdgis2.ent' >`

See further 25.1 *Overall Structure of Writing System Declaration*

TEI.XML (TEI XML DTD fragment) controls whether the TEI DTD generated is to be in XML or SGML.

Note If this entity is undeclared, or has the value “IGNORE”, the parameter entities defining omissibility in each element declaration in the DTD will be instantiated, and the resulting DTD will therefore be processable only as an SGML DTD. To produce an XML version of the DTD, the user should declare this entity with a value of “INCLUDE”; this will override the default.

Declaration `<!ENTITY % TEI.XML 'IGNORE' >`

See further 3.8.4 *Generation of an XML DTD*

termtags system entity with definitions for basic terminology tags.

Note The default definition of this entity is used to invoke the element declarations for the ‘nested’ style of terminological markup. To invoke the alternative ‘flat’ style declarations, this entity should be defined as “PUBLIC ‘-//TEI P4//ELEMENTS Terminological Databases (Flat)//EN’ ‘teite2f.dtd’”.

Declaration `<!ENTITY % termtags PUBLIC "-//TEI P4//ELEMENTS Terminological Databases (Nested)//EN" 'teite2n.dtd' >`

See further 13.4 *Overall Structure of Terminological Documents*

version (version) defines the name to be used for the root element of a concurrent markup stream for marking pages and lines of some reference edition.

Note Some name for the edition should be supplied by defining this parameter entity within the appropriate DTD subset. If none is defined, the parameter entity (and thus the document type) default to “ref”.

Declaration `<!ENTITY % version 'ref' >`

See further 31.6 *Concurrent Markup for Pages and Lines*