

STEPMod DTD for EXPRESS

This document contains the DTD for the ISO EXPRESS language used by the STEP Modules Repository.

Table of contents

1 STEPMod DTD.....	2
--------------------	---

1. STEPMod DTD

The DTD used in the STEP Modules Repository as follows. Only the structure of EXPRESS schemas is supported. Any algorithms are simply represented as text (i.e. #PCDATA)

```

<!--
  $Id: express_model.dtd,v 1.14 2003/07/18 21:50:20 thendrix Exp $
-->
<!-- EXPRESS MODEL DECLARATION MODULE -->

<!-- This set of declarations defines the EXPRESS-2 Model as of the
date shown in the Formal Public Identifier (FPI) for this entity.

This set of declarations may be referred to using a public external
entity declaration and reference as shown in the following two lines:

<!ENTITY % express.model PUBLIC "-//TC184 SC4//DTD EXPRESS Model 20010901">
%express.model;

If the parameter entity description.content used within this set
of declarations is to be given non-default values, the appropriate
declaration should be given before calling in this package
(i.e., before the "%express.model;" reference).
-->

<!-- This file was developed by Eurostep and supplied to NIST under
contract.
Authors:
  Rob Bodington
  David Price
  Nigel Shaw
  Phil Spiby
Reviewed by:
  Josh Lubell
-->

<!-- underlying type -->
<!ENTITY % underlying
"aggregate*, (typename | builtintype )"
>

<!ENTITY % description.content "#PCDATA">

<!-- Attributes for source control software like RCS or CVS -->
<!-- The rcs attributes are keywords for RCS source control systems -->
<!-- rcs.date $Date: 2003/07/18 21:50:20 $ -->
<!-- rcs.revision $Revision: 1.14 $ -->
-->
<!-- reference is the source of the schema, e.g. ISO 10303-41 -->
<!ELEMENT express (description?, application?, schema+)>

```

```

<!ATTLIST express
  language_version (1 | 2 | 3) "1"
  rcs.date CDATA #REQUIRED
  rcs.revision CDATA #REQUIRED
  reference CDATA #IMPLIED
  description.file CDATA #IMPLIED
>

<!ELEMENT description (%description.content;)*>

<!ELEMENT application EMPTY>
<!ATTLIST application
  name CDATA #IMPLIED
  owner CDATA #IMPLIED
  url CDATA #IMPLIED
  version CDATA #IMPLIED
  source CDATA #REQUIRED
>

<!ELEMENT schema (description?, interface*, constant*, (entity | function |
procedure | rule | subtype.constraint | type)*, graphic.element?)>

<!ATTLIST schema
  name NMTOKEN #REQUIRED
  version NMTOKEN #IMPLIED
>

<!ELEMENT interface (interfaced.item*, described.item*)>
<!ATTLIST interface
  kind (use | reference) "use"
  schema NMTOKEN #REQUIRED
>

<!ELEMENT interfaced.item EMPTY>
<!ATTLIST interfaced.item
  name NMTOKEN #REQUIRED
  alias NMTOKEN #IMPLIED
>

<!--
  A description of any item in the interface. In other words the
  documentation of an imported construct.
  item is the name of the imported construct
  attribute is only used if the imported construct is an entity in whihc
  case, item is the name of the entty and item is the name of the attribute
-->
<!ELEMENT described.item (description?)>
<!ATTLIST described.item
  item NMTOKEN #REQUIRED
  attribute NMTOKEN #IMPLIED
  kind (ENTITY | TYPE | ATTRIBUTE | FUNCTION | PROCEDURE | CONSTANT )
"ENTITY">

```

```

<!ELEMENT constant (description?, %underlying;)>
<!ATTLIST constant
    name NMTOKEN #REQUIRED
    expression CDATA #REQUIRED
>

<!ELEMENT type (description?, aggregate*, (typename | builtintype |
enumeration | select), where*, graphic.element?)>
<!ATTLIST type
    name NMTOKEN #REQUIRED
>

<!ELEMENT typename EMPTY>
<!ATTLIST typename
    name NMTOKEN #REQUIRED
>

<!ELEMENT builtintype (graphic.element?)>
<!ATTLIST builtintype
    type (BINARY | BOOLEAN | GENERIC | GENERICENTITY | INTEGER |
LOGICAL | NUMBER | REAL | STRING) "STRING"
    width CDATA #IMPLIED
    fixed (YES | NO) "NO"
    precision CDATA #IMPLIED
    typelabel NMTOKEN #IMPLIED
>
<!-- Width to allow for Binary and string -->
<!-- Fixed to allow for Binary and string -->
<!-- Precision to allow for REAL -->
<!ELEMENT aggregate EMPTY>
<!ATTLIST aggregate
    type (AGGREGATE | ARRAY | BAG | LIST | SET) "SET"
    optional (YES | NO) "NO"
    unique (YES | NO) "NO"
    lower CDATA #IMPLIED
    upper CDATA #IMPLIED
    typelabel NMTOKEN #IMPLIED
>

<!ELEMENT select (graphic.element?)>
<!ATTLIST select
    extensible (YES | NO) "NO"
    genericentity (YES | NO) "NO"
    basedon NMTOKEN #IMPLIED
    selectitems NMTOKENS #IMPLIED
>

<!ELEMENT graphic.element EMPTY>
<!ATTLIST graphic.element
    image CDATA #IMPLIED
    page CDATA #REQUIRED
    xcoord CDATA #IMPLIED
    ycoord CDATA #IMPLIED
>
<!ELEMENT enumeration (graphic.element?)>

```

```

<!ATTLIST enumeration
    extensible (YES | NO) "NO"
    basedon NMTOKEN #IMPLIED
    items NMTOKENS #IMPLIED
>
<!ELEMENT entity (description?, explicit*, derived*, inverse*, unique*,
where*, graphic.element?)>
<!ATTLIST entity
    name NMTOKEN #REQUIRED
    abstract.entity (YES | NO) "NO"
    abstract.supertype (YES | NO) "NO"
    supertypes NMTOKENS #IMPLIED
    super.expression CDATA #IMPLIED
>
<!ELEMENT explicit (description?, %underlying;, redeclaration?)>
<!ATTLIST explicit
    name NMTOKEN #REQUIRED
    optional (YES | NO) "NO"
>
<!ELEMENT derived (description?, %underlying;, redeclaration?)>
<!ATTLIST derived
    name NMTOKEN #REQUIRED
    expression CDATA #REQUIRED
>
<!ELEMENT inverse (description?, inverse.aggregate?, redeclaration?)>
<!ATTLIST inverse
    name NMTOKEN #REQUIRED
    entity NMTOKEN #REQUIRED
    attribute NMTOKEN #REQUIRED
>
<!ELEMENT inverse.aggregate EMPTY>
<!ATTLIST inverse.aggregate
    type (BAG | SET) "SET"
    lower CDATA #IMPLIED
    upper CDATA #IMPLIED
>
<!ELEMENT redeclaration EMPTY>
<!ATTLIST redeclaration
    entity-ref NMTOKEN #REQUIRED
    old_name NMTOKEN #IMPLIED
>
<!ELEMENT where (description?)>
<!ATTLIST where
    label NMTOKEN #IMPLIED
    expression CDATA #IMPLIED
>
<!--<!ELEMENT unique (description?, unique.attribute)-->
<!ELEMENT unique (description?, unique.attribute+)>
<!ATTLIST unique
    label NMTOKEN #IMPLIED
>
<!ELEMENT unique.attribute EMPTY>
<!ATTLIST unique.attribute
    entity-ref NMTOKEN #IMPLIED

```

```

        attribute NMTOKEN #REQUIRED
    >
<!ELEMENT subtype.constraint (description?, graphic.element?)>
<!ATTLIST subtype.constraint
    name NMTOKEN #REQUIRED
    entity NMTOKEN #REQUIRED
    abstract.supertype (YES | NO) "NO"
    totalover NMTOKENS #IMPLIED
    super.expression CDATA #IMPLIED
>

<!ELEMENT function (description?, parameter*, %underlying;, algorithm)>
<!ATTLIST function
    name NMTOKEN #REQUIRED
>

<!ELEMENT parameter (description?, %underlying;)>
<!ATTLIST parameter
    name NMTOKEN #REQUIRED
>

<!ELEMENT procedure (description?, parameter*, algorithm?)>
<!ATTLIST procedure
    name NMTOKEN #REQUIRED
>

<!ELEMENT rule (description?, algorithm?, where+)>
<!ATTLIST rule
    name NMTOKEN #REQUIRED
    appliesto NMTOKENS #REQUIRED
>

<!ELEMENT algorithm (#PCDATA)>

```