

*In this functional part template, headings that should be retained are not surrounded by < ..>, not in italic text and are not contained in a comment box(such as this box).
 Elements that are within <...> should be replaced by user defined or developed text
 Elements in italic and marked, coloured comment boxes are for guidance and should be removed from the final version.
 Delete this box from the final version.*

Functional Part

Name	<replace with name of exchange requirement without er_ prefix>
-------------	--

Identifier	<give the exchange requirement a unique identifier>	IFC Release	<identify the IFC release to which this functional part is bound>
-------------------	---	--------------------	---

Change Log		
<yyyy-mm-dd>	<action>	<author email address>

Overview

<replace with overview text which may include formulae and symbols >

Provide a textual overview of the functional part. It should provide a human readable view of the intent of the functional part. The text for this section of the functional part documentation may be based on the description section of the equivalent part text within an exchange requirement.

Take care with formulae and symbols used in the overview. They will be ultimately displayed in HTML and may need complex escape sequences to display.

Results

<replace with results text >

Describe the results of execution of the functional part.

A functional part may comprise several concepts. The sequence in which concepts occur within a functional part should be progressively defined.

Each concept will have a name which should be defined by the user. In the case of a reusable concept, it should have the same name as in previous occurrences.

CONCEPT : <Name>

<replace with concept description >

Provide a description of the concept

<provide any graphics that are needed to illustrate the concept>

Each concept will contain a table with columns headed as below. Within each column, information should be provided that specifies the particular parts of the model used.

[Type text]

Information Needed	Entity/Pset/Functional Part	MAN	REC	OPT	NOT
<p><describe the information needed></p> <p><provide further qualification about the information needed, where it can be found or propositions about its use.></p>	<p>EITHER</p> <p><entity name.attribute name→datatype></p> <p><i>Note that the data type shown for an attribute may be an entity, a defined data type, a simple data type or another functional part e.g.</i> <i>IfcTask.Status → IfcLabel</i></p> <p>OR</p> <p><pset name.property name→property type::datatype></p> <p><i>Note that the property type shown for a property may reference a data type which is conventionally shown following a pair of colons e.g.</i> <i>Pset_WorkPlanPeriod.PlanTaskPeriod → IfcPropertySingleValue::IfcInteger</i></p> <p>OR</p> <p><refer to a functional part that provides the information needed according to column 1></p> <p><i>e.g. fp_define_by_properties or fp_nests (IfcTask) where a parameter may be given to the called functional part</i></p> <p><i>Where the data type is an attribute enumeration or a property enumeration, the predefined or selected type may also be shown after an = sign e.g. as in = SERVICELIFE</i></p>	✓	✓	✓	✓
<p>< continue for each item of information needed></p>		✓	✓	✓	X
		Tick if used/else delete	Tick if used/ else delete	Tick if used/ else delete	Use X for not used attributes

IFC Entities Required

- <provide a list of all the IFC entities used in the functional part; place in alphabetical order>

The list should contain all of the entities referenced from the IFC model and not just be the entity in a flattened model.

[Type text]

IFC Datatypes Required

- <provide a list of all the IFC datatypes used in the functional part; place in alphabetical order>

The list should contain all of the datatypes referenced from the IFC model including simple types, defined types, enumerations and selects.

IFC Functions Required

- <provide a list of all the IFC functions used in the functional part; place in alphabetical order>

*Take care to confirm the functions that need to be used from the IFC model and ensure that they are copied into the functional part. Then **DOUBLE CHECK TO ENSURE THAT FUNCTIONS ARE INCLUDED.***

IFC Property Sets Required

- <provide a list of all the IFC property sets used in the functional part; place in alphabetical order>

IDM Functional Parts Required

- <provide a list of all the IDM functional parts used in this functional part; place in alphabetical order>

EXPRESS-G

<provide a copy of the EXPRESS-G graphical notation of the functional part>

EXPRESS Schema

<Provide a copy of the EXPRESS language data definition of the functional part>

The EXPRESS language schema is usually displayed in a Courier New 8pt font style to identify it as different to the remaining content of the functional part.

Examples

Examples provide implementation guidance for the functional part. Progressively, implementation guidance within the functional parts will take over the current role of the IFC Implementation Guide.

Several examples may be given. Each example should be given a sequential number within the functional part and should have a name

Example <#>: <Name>

<replace with example text>

<replace with example graphics>

<replace with example physical file fragment>

using Courier New 8pt text and place the file fragment (may be surrounded by a border for emphasis but this is not necessary) e.g.

```
/* instance of the IfcCostValue */
#10000=IFCCOSTVALUE($,$,#11001,#11002,#12001,#12002,'List Price',$);
```

[Type text]

