

## Facilities Management

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on May 27, 2008 ([view change](#))

Labels: (None)

## Facilities

---

Incorporates all IDM components that are about facilities management where facilities management deals with operation and maintenance following the completion of construction.

## Process Maps

---

[Facilities Management \(PM\)](#)

## Exchange Requirements

---

[Exchange Asset \(ER\)](#)

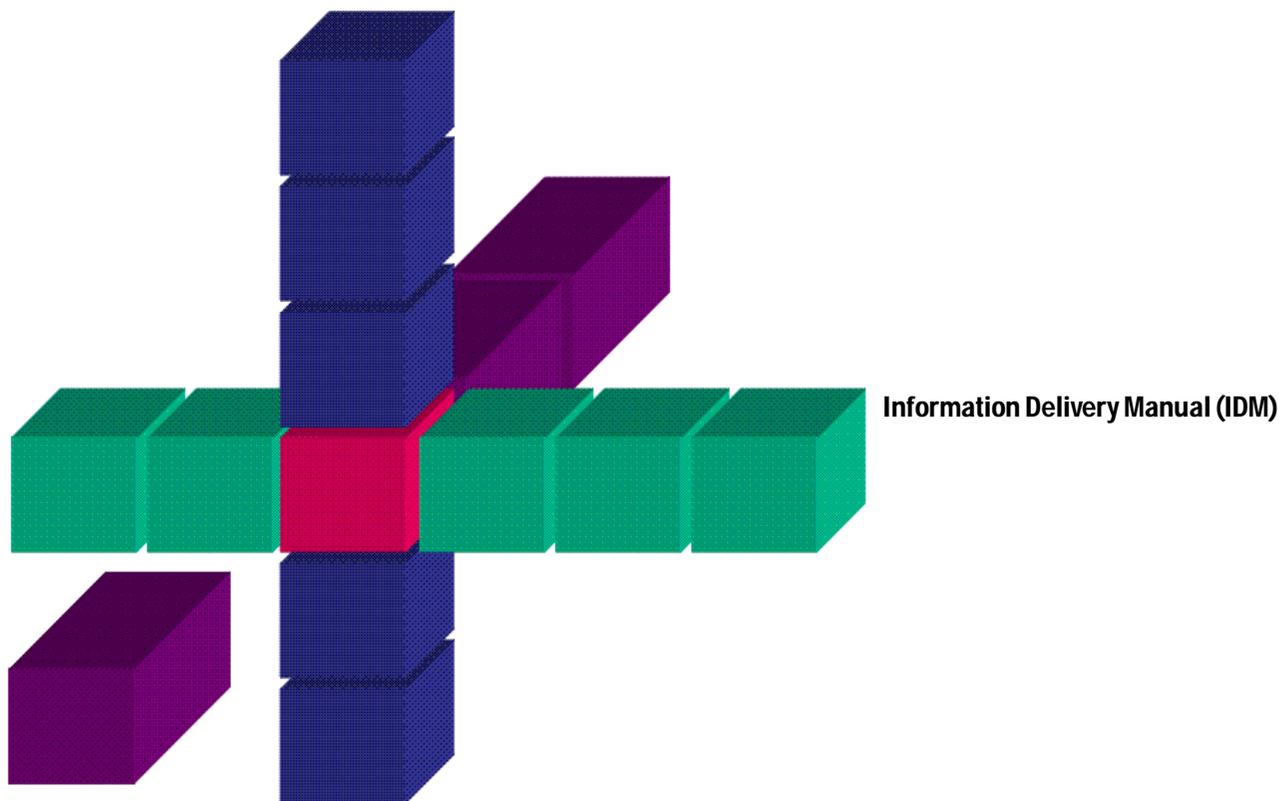
[Exchange Asset Inventory \(ER\)](#)

[Exchange Condition Assessment \(ER\)](#)

[Exchange Maintenance Plan \(ER\)](#)

[Exchange Maintenance Schedule \(ER\)](#)

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## Facilities Management (PM)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Jul 23, 2007 ([view change](#))

Labels: (None)

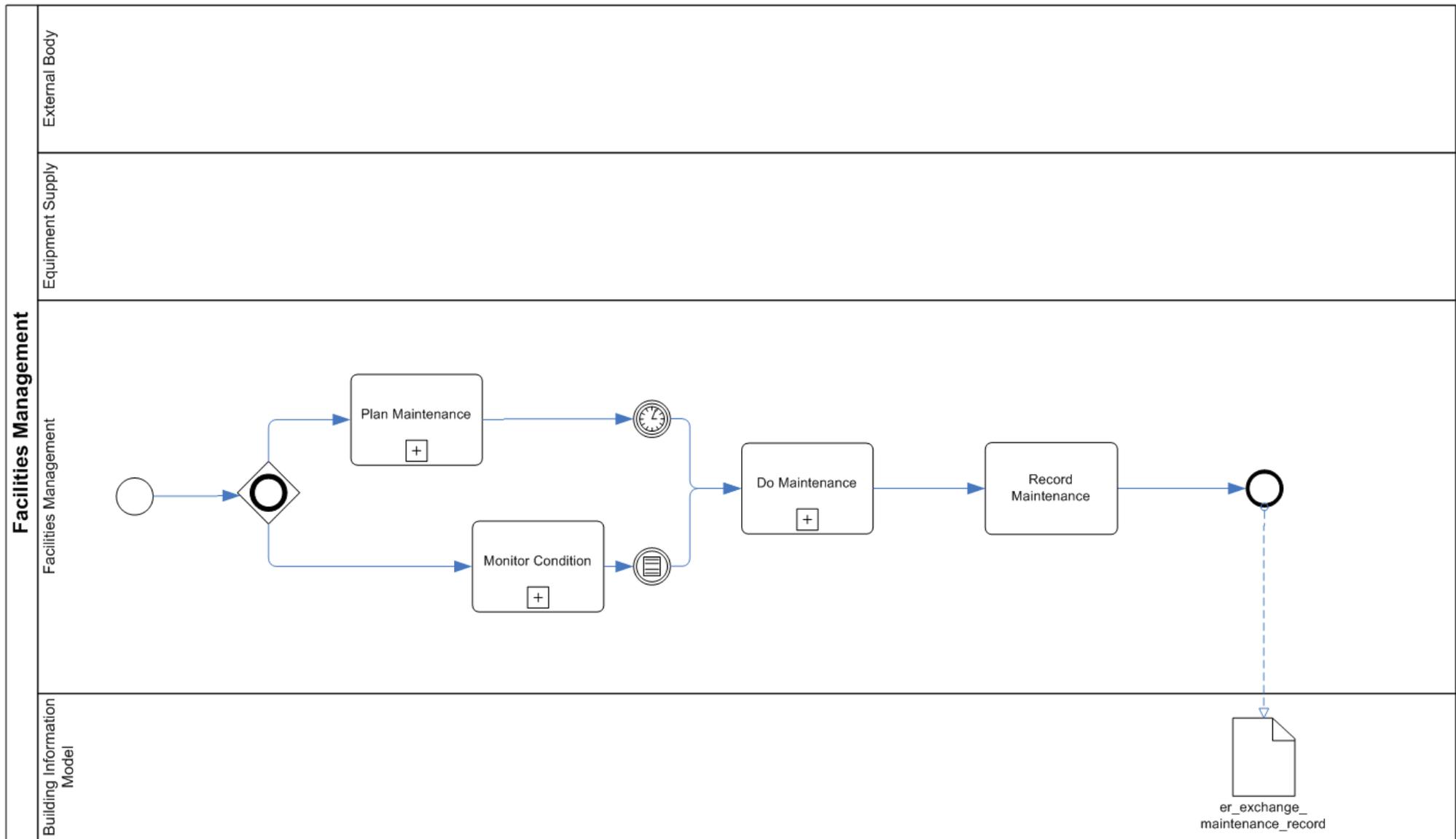
## Facilities Management A - Process Maps (PM)

Facilities Management comprises many processes. For present purposes, this document is concerned with the maintenance aspect of facilities management and outlines the following processes: -

- Plan Maintenance including identification of the assets to be maintained, maintenance actions required and the scheduling of those actions.
- Monitor Condition; in this case strictly the assessment of condition from visual inspection, the recording of condition and the determination of maintenance requirement from the condition assessment.
- Do Maintenance including work order creation and execution.
- Record Maintenance.

For present purposes, an asset may be considered as something which contributes to the value of the organization that owns it and consequently may, or may not, be subject to maintenance requirements. For the purpose of this process, it is assumed that only assets to be maintained are in scope.

Much of the information required by maintenance applications is available from the design and installation stages of a project.



<b>Facilities Management</b>	Author: Jeffrey Wix	Created: 29/09/2006	
	Version: 0.2 Status: Created	Modified: 28/02/2007	
© International Alliance for Interoperability 2007	bpmn_facilities_management		

## Process Overview

Id	Task	Name	Description	Process Stage(s)
1	Sub-Process (collapsed)	Plan Maintenance		Operation and maintenance(9)
2	Sub-Process (collapsed)	Monitor Condition		Operation and maintenance(9)
3	Sub-Process (collapsed)	Do Maintenance		Operation and maintenance(9)
4	Task	Record Maintenance		Operation and maintenance(9)

## Role Overview

The primary roles shown in the process model diagram (as "swimlanes") are defined as:

- Building Information Model
- Equipment Supplier
- External Body
- Facilities Management

## Exchange Requirements

The following Exchange Requirements describe the information need in the activities identified in this [Process Map](#).

Name	Description
er_exchange_maintenance_plan	Planning of maintenance actions by interval
er_exchange_maintenance_schedule	Scheduling maintenance actions in time
er_exchange_maintenance_work_order	<i>to be done</i>
er_exchange_maintenance_record	<i>to be done</i>
er_exchange_condition_data	<i>to be done</i>

## Process Descriptions

In this section, all of the processes shown in the diagram above are described.

### Plan Maintenance - ID:1

Type	Sub-Process (collapsed)
Name	Plan Maintenance
Documentation	<p>This process is concerned with acquiring sufficient information to enable planning and scheduling of the maintenance to be undertaken. Maintenance planning can encompass:</p> <ul style="list-style-type: none"> <li>● periodic maintenance requirements that can be discovered from data available during the design and construction phases;</li> <li>● maintenance requirements determined from condition monitoring and assessment where the need for maintenance is not immediately urgent and can be included within a planned requirement;</li> <li>● maintenance requirements determined from demand requirements where the need for maintenance is not immediately urgent and can be included within a planned requirement;</li> </ul> <p>Inclusion of condition and demand requirements would normally be on the basis that work can be scheduled to occur at the same time as a periodic maintenance activity on the same or an adjacent asset. Where condition or demand maintenance is considered to be urgent, it can still be dealt with directly by issuing a work order (see Do Maintenance).</p> <p>Using the identities of equipment requiring maintenance from the asset register, the scope of this process is to identify the required planned maintenance actions at an early stage and to prepare scheduled work orders from which individual work orders can be derived as work is required. The generation of actual work orders and the carrying out of maintenance actions is outside the scope of this process. The need to provide information from a maintenance system to a financial system is encompassed within this process. This is through the ability to specify a cost for the scheduled work order (which will be the estimated cost of carrying out work against a derived work order and not the actual cost resulting from the work undertaken against that work order).</p>

Planned maintenance is undertaken on the basis of actions being carried out on assets at known points in time. This may be by definition of a set period between maintenance activities or by planning required condition or demand maintenance at some known future point. Maintenance plans can be established for the various assets and, from these, scheduled work orders can be specified which set out the criteria for the work to be done including identification of labor, tools and consumables requirements, time allowances, budgets etc.

Process Scope:

- Identify Assets To Be Maintained
- Identify Maintenance Action
- Assign Action To Asset
- Prepare Scheduled Work Order
- Schedule Maintenance

### Monitor Condition - ID:2

Type	Sub-Process (collapsed)
Name	Monitor Condition
Documentation	<p>This process is concerned with determining if the current condition of an asset is such that it requires maintenance in order to bring it to a required level of operating efficiency.</p> <p>There are three possible scenarios that may result from monitoring the condition of an asset:</p> <ol style="list-style-type: none"> <li>1 Maintenance is required in which case the work may be either -             <ul style="list-style-type: none"> <li>-- sufficiently urgent as to warrant the issue of a demand work order</li> <li>-- able to be carried out at some planned future point at which time work will be undertaken on the asset anyway or an adjacent asset.</li> </ul> </li> <li>2 The condition of the asset is such that it requires replacement in which case the work requirement may be subject to a project requirement.</li> <li>3 Condition is satisfactory in which case no maintenance is currently required.</li> </ol> <p>Monitoring the condition of an asset may be undertaken in two ways:</p> <ol style="list-style-type: none"> <li>1 Instrumentation is in place that enables operating parameters to be continuously monitored; the parameters being selected such that they indicate the current condition of the asset (e.g. vibration). Where condition is continuously monitored, the point at which maintenance will be required will be determined by the values of the parameters being monitored exceeding a given value.</li> <li>2 Condition of the asset is periodically monitored by inspection that may be either visual, carried out with the assistance of instruments or a combination. Where condition is periodically monitored, the point at which an inspection takes place may be determined either by -             <ul style="list-style-type: none"> <li>-- A planned program of inspection, the inspection itself being considered to be part of a planned maintenance regime and subject to the issue of a work order.</li> <li>-- A fault report being generated by a user of the asset, the fault needing to be inspected to determine its cause and whether maintenance is required.</li> </ul> </li> </ol> <p>Process Scope is:</p> <ul style="list-style-type: none"> <li>- Assess Condition (by inspection)</li> <li>- Report Condition</li> <li>- Determine Maintenance Requirement</li> </ul>

### Do Maintenance - ID:3

Type	Sub-Process (collapsed)
Name	Do Maintenance
Documentation	<p>The requirement is the actual creation and execution of a work order in response to a maintenance requirement. The requirement for maintenance may originate from various sources namely:</p> <ul style="list-style-type: none"> <li>- Planned (within the planned preventive maintenance system). In this case, the resulting work order is an instance of a Master Work Order that is defined as a result of a previous process.</li> <li>- Condition. This is similar in nature to the planned maintenance action in that the work order may be an instance of a Master Work Order. However, the execution of the work order is the result of the asset having reached a certain condition through usage or it having been operated for a set amount of runtime (as opposed to calendar time). Therefore, whilst the maintenance action may be planned, it may not be possible to associate it with a planned maintenance schedule. Issuing of the work order is as the result of inspection of condition or knowledge of parameters determining condition through instrumentation (e.g. vibration analysis indicating the condition of bearings).</li> <li>- Demand. A demand maintenance action is carried out in response to a request for action from the users of the facility or from the occurrence of an unplanned event(s) affecting the satisfactory normal operation of the facility. Frequently, demand maintenance is executed through a Helpdesk facility which takes the request, logs it, ensures that it generates a required maintenance action and issues the necessary work order for the action.</li> </ul> <p>It is necessary to be concerned with each of these types of work order since each plays a part in the establishment and operation of an engineering maintenance regime. Demand maintenance may be responsible for over 50% of the calls made on the maintenance capability.</p> <p>The result of carrying out work orders is that the life of the asset is extended since the objective of maintenance is to return it to an optimal operating condition. Additional benefits occur from the information which is acquired whilst executing the work order. These include information for:</p>

- maintenance history so that long term analysis and decision making is supported (see separate process);
- better control of spares;
- assessment of staff performance through time sheet provision and logging of time against work orders (which may be particularly relevant when using external maintenance contractors).

Process Scope:

- Receive Fault Report
- Carry Out Inspection
- Decide on Maintenance/ Small Works or Refurbishment
- Generate Work Order
- Assign Work Order
- Execute Work Order
- Quality Control

Maintenance work is carried out against an instruction that is known as a work order. The work order may be required as a result of one or more of the following:

1. Planned Preventive Maintenance designated by a Master Work Order;
2. Report of condition of an asset following an inspection.
3. Report of condition of an asset through provision of readings from instrumentation.
4. Fault Report

The key factors in the process are the generation, assignment and execution of the work order. If the work order results from a fault report, there are prior activities involved in receiving and recording the report and possible inspection of the fault to determine the required work. Inspection may also be appropriate in certain condition based maintenance activities.

A number of legal and organizational constraints may apply to the completion of a work order. Since maintenance can occur on any type of site, the range of constraints needs to be understood. For instance, in a high security establishment, access may only be permitted at certain times and in accordance with certain rules. On an industrial site, the presence of hazardous substances may require that particular precautions are taken when doing work.

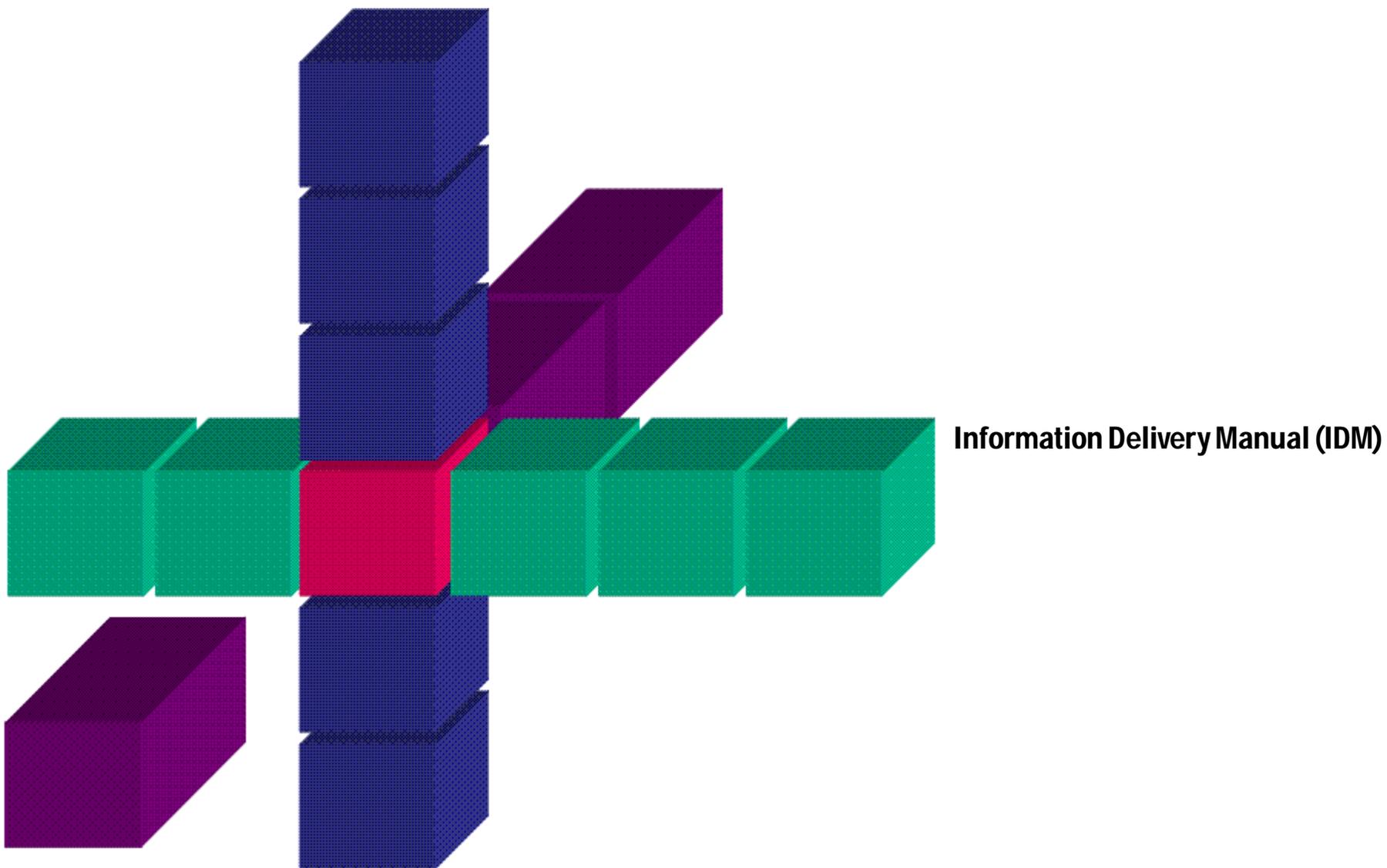
The process identified below is preceded by the necessity to identify assets and, in the case of Planned Preventive Maintenance, by the definition of Master Work Orders and the scheduling of maintenance work.

#### Record Maintenance - ID:4

Type	Task
Name	Record Maintenance
Documentation	Records the results of maintenance actions completed.

#### Children [Hide Children](#) | [View in hierarchy](#)

-  [FM Coordination Gateways](#) (Information Delivery Manual (IDM))
-  [FM Data Objects](#) (Information Delivery Manual (IDM))
-  [FM Exchange Requirement Data Objects](#) (Information Delivery Manual (IDM))
-  [Plan Maintenance](#) (Information Delivery Manual (IDM))



## FM Coordination Gateways

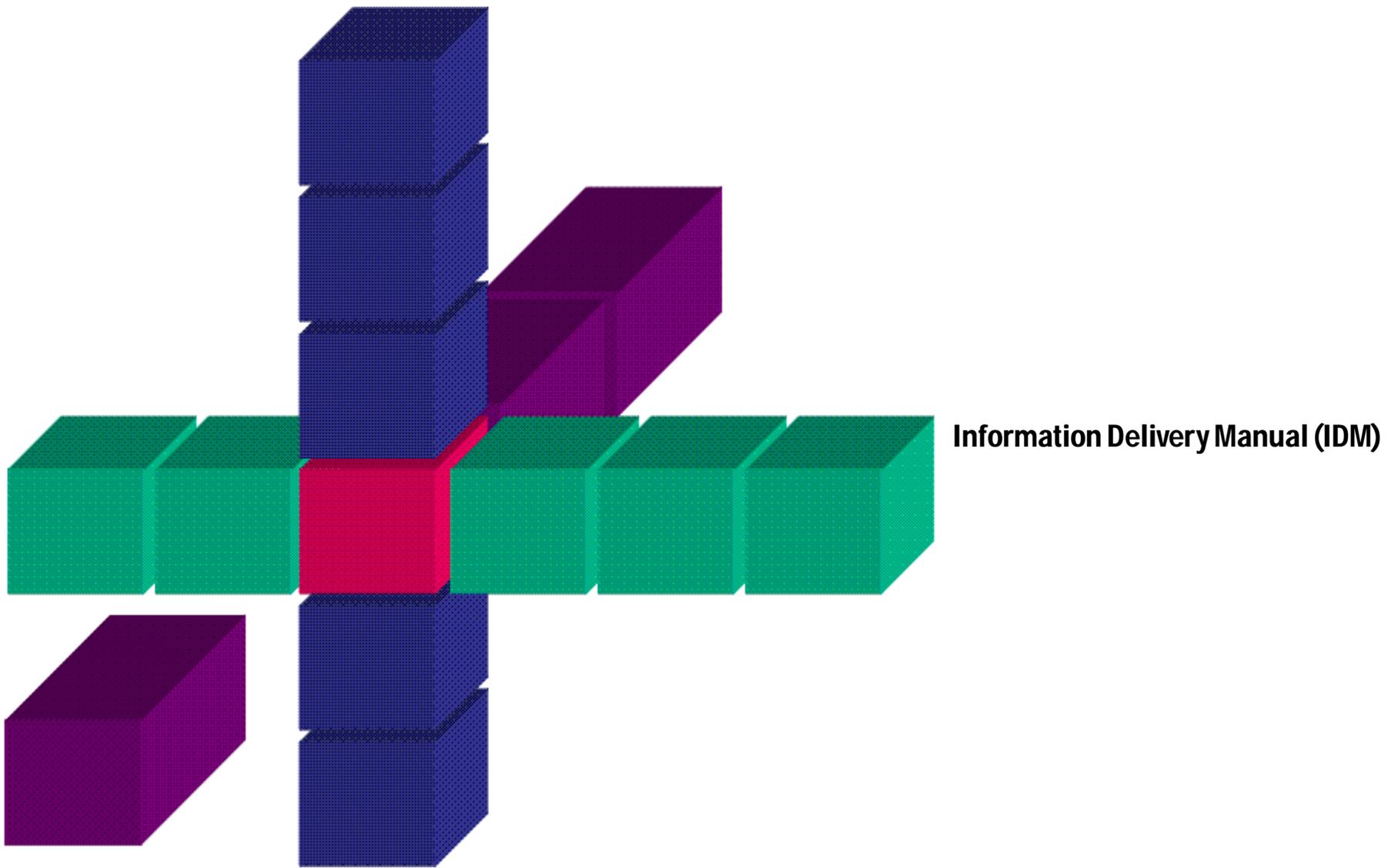
Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Jul 23, 2007 ([view change](#))

Labels: (None)

# Coordination Gateway

---

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## FM Data Objects

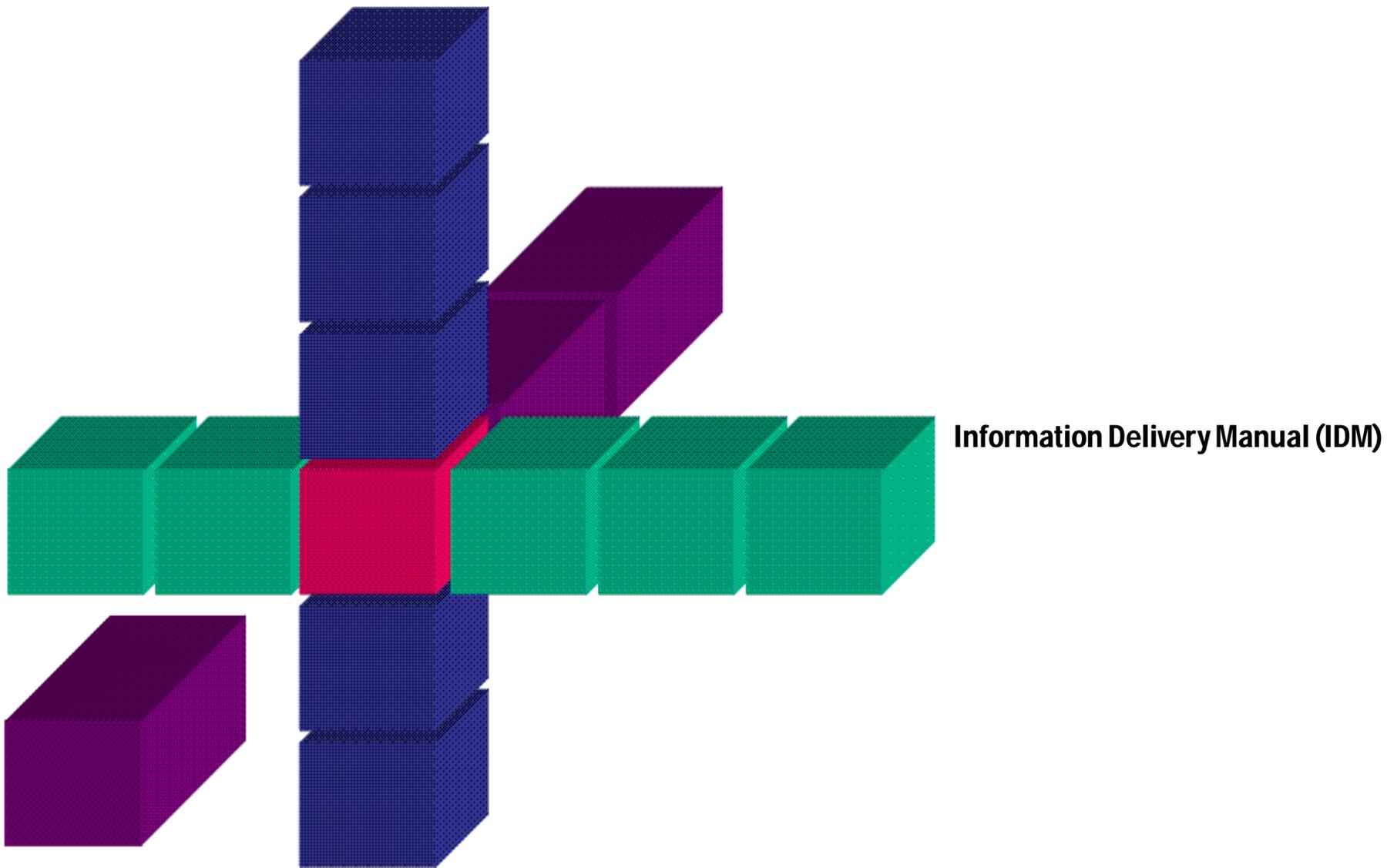
Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Jul 23, 2007

Labels: (None)

# Data Objects

---

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## FM Exchange Requirement Data Objects

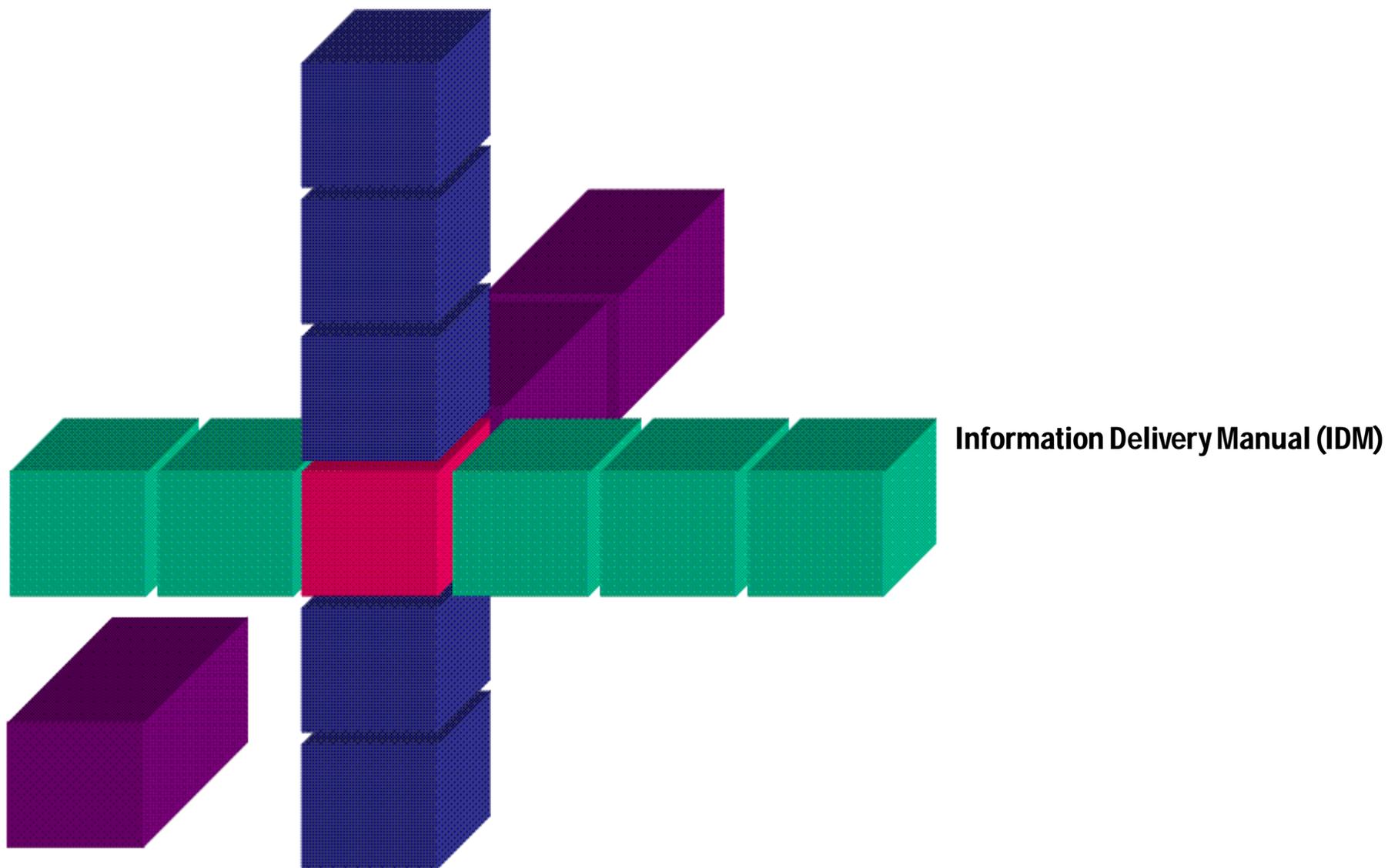
Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Jul 23, 2007

Labels: (None)

# Exchange Requirement Data Objects

---

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## Plan Maintenance

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Jul 23, 2007 ([view change](#))

Labels: (None)

## Plan Maintenance (ID: 1)

---

This process is concerned with acquiring sufficient information to enable planning and scheduling of the maintenance to be undertaken. Maintenance planning can encompass:

- periodic maintenance requirements that can be discovered from data available during the design and construction phases;
- maintenance requirements determined from condition monitoring and assessment where the need for maintenance is not immediately urgent and can be included within a planned requirement;
- maintenance requirements determined from demand requirements where the need for maintenance is not immediately urgent and can be included within a planned requirement;

Inclusion of condition and demand requirements would normally be on the basis that work can be scheduled to occur at the same time as a periodic maintenance activity on the same or an adjacent asset. Where condition or demand maintenance is considered to be urgent, it can still be dealt with directly by issuing a work order (see Do Maintenance).

Using the identities of equipment requiring maintenance from the asset register, the scope of this process is to identify the required planned maintenance actions at an early stage and to prepare scheduled work orders from which individual work orders can be derived as work is required. The generation of actual work orders and the carrying out of maintenance actions is outside the scope of this process.

The need to provide information from a maintenance system to a financial system is encompassed within this process. This is through the ability to specify a cost for the scheduled work order (which will be the estimated cost of carrying out work against a derived work order and not the actual cost resulting from the work undertaken against that work order).

Planned maintenance is undertaken on the basis of actions being carried out on assets at known points in time. This may be by definition of a set period between maintenance activities or by planning required condition or demand maintenance at some known future point. Maintenance plans can be established for the various assets and, from these, scheduled work orders can be specified which set out the criteria for the work to be done including identification of labor, tools and consumables requirements, time allowances, budgets etc.

### Identify Asset - ID: 1.1)

Type	Sub-Process (collapsed)
Name	Design Electrical System
Documentation	The objective of the design activity for a project is the progressive development and refinement of the requirements expressed in the program (brief) and then to propose design solutions to those requirements including construction/installation methods, materials, technical solutions. Design solutions proposed should be validated against the provision of building codes and regulations for the specific location of the project and should be in accordance with sustainability criteria including environmental impact and service life. The cost of the design solutions proposed should be estimated and managed through the progressive development of a cost plan. In this case, the design activity is concerned with electrical systems.

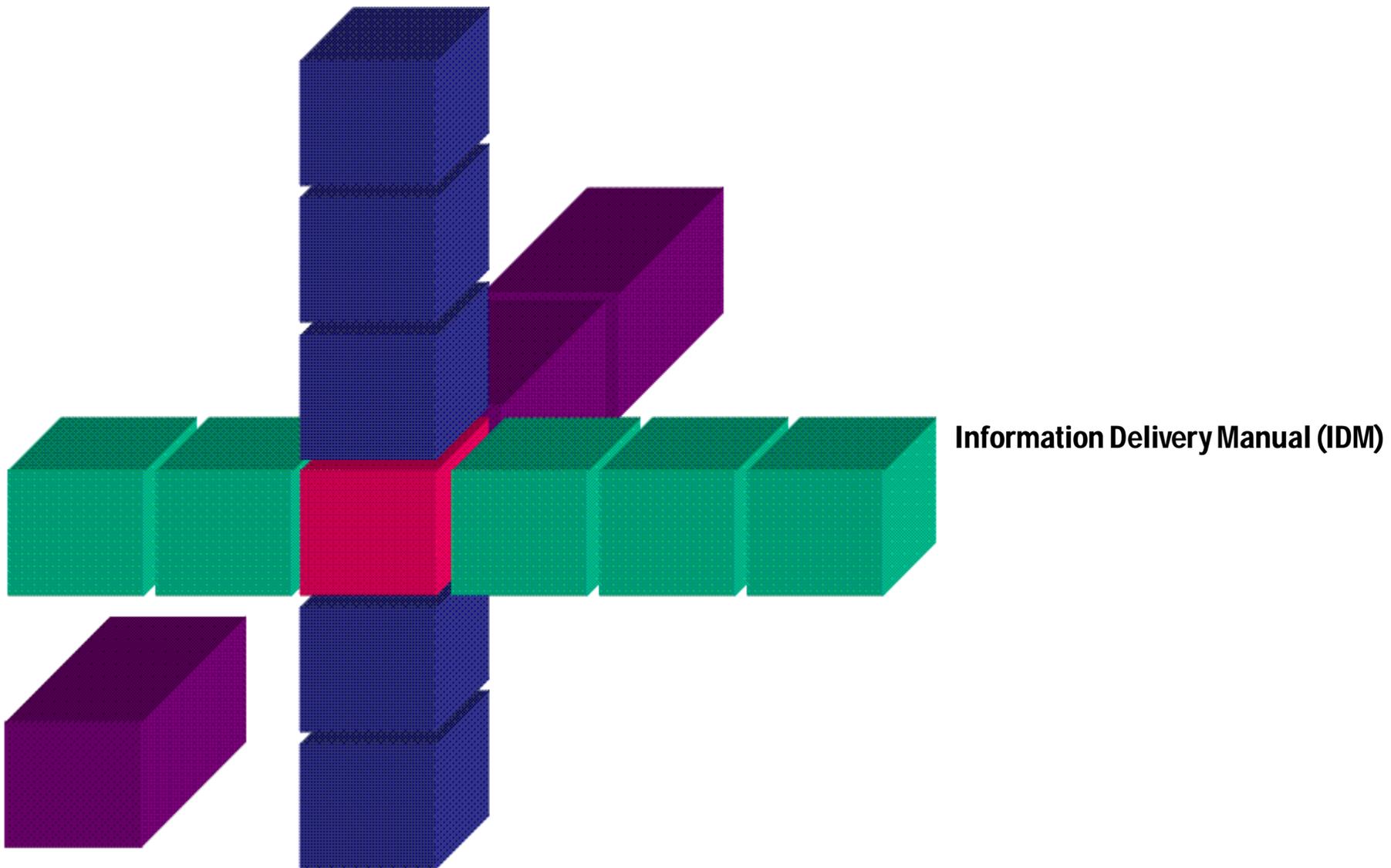
Unable to render embedded object: File (bpmn\_elec\_design\_2.PNG) not found.

## Process Overview

---

In the following subsections each process element will be described in more detail.

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## Exchange Asset (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 14, 2007 ([view change](#))

Labels: (None)

# Exchange Asset

---

## Project Stage

---

Stage	Name	Valid
0	Portfolio requirements	
1	Conception of need	
2	Outline feasibility	
3	Substantive feasibility	
4	Outline conceptual design	
5	Full conceptual design	
6	Coordinated design and procurement	
7	Production information	
8	Construction	
9	Operation and maintenance	
10	Disposal	

## Overview

---

The scope of this exchange requirement is to support the exchange of information concerning assets where an asset is a grouping of elements that are identified together for financial or operational or maintenance purposes.

The intent is to know:

- What elements make up the asset
- Asset identity
- Where the asset is
- When it was incorporated
- The value of the asset
- Who owns, uses and is responsible for the asset

The exchange of asset information should occur within the construction phase of a project (at the latest). Practically, it is expected that asset information should be available at the point of turnover (handover) from the organization performing the construction, installation or erection role to the facility owner or operator. However,

owning/operating knowledge may be required for grouping products into assets and this could cause the completion of this exchange requirement at the beginning of the operation and maintenance stage. This is typically the case in current working; this exchange requirement seeks to improve this process.

If the knowledge of the facility owner/operator is brought to bear early in the project, there is no reason why the grouping of elements into assets cannot be applied even during the design stage of a project. Much of the information that may be needed about products and assets is available even at that time and can be applied.

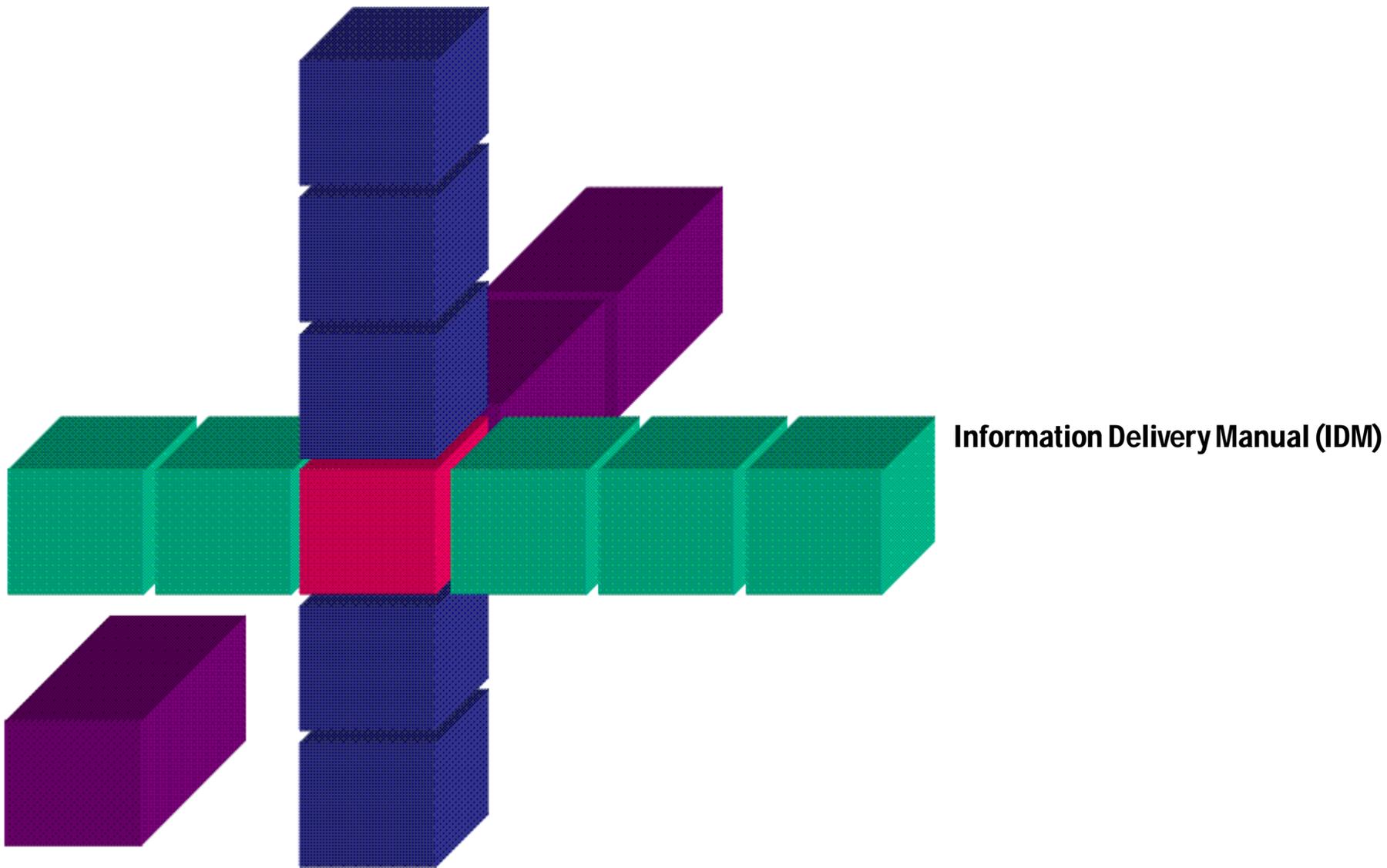
This is shown in the project stage/attribute matrix at the end of this exchange requirement which shows attributes that may be (should be) asserted at each point in the lifecycle leading up to the contractual exchange requirement.

## Information Requirements

Context	Description
Precursor	All objects that are to be incorporated into assets are properly identified (through a globally unique identifier) and have their own histories.
<a href="#">Group</a>	Assigns the set of elements and sub assets that will participate in the asset
<a href="#">Asset</a>	Provides for the collection of all information about the asset
<a href="#">Optional Information</a>	Further optional information that may be provided about the asset

### Children [Hide Children](#) | [View in hierarchy](#)

-  [1 - Group - Exchange Asset \(ER\)](#) (Information Delivery Manual (IDM))
-  [2 - Asset - Exchange Asset \(ER\)](#) (Information Delivery Manual (IDM))
-  [3 - Optional Information - Exchange Asset \(ER\)](#) (Information Delivery Manual (IDM))



## 1 - Group - Exchange Asset (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 14, 2007 ([view change](#))

Labels: (None)

## Group

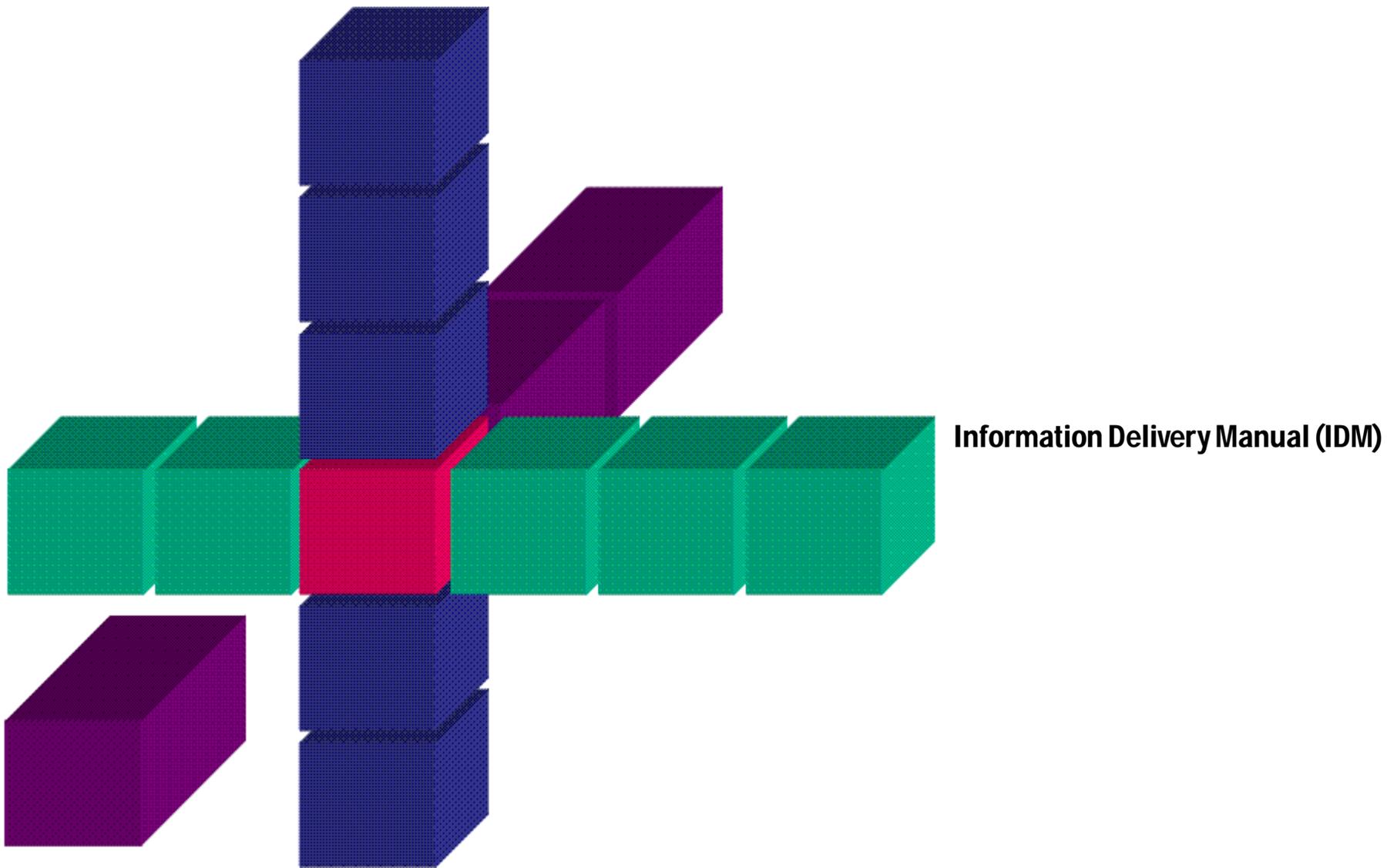
---

Assigns the set of elements and sub assets that will participate in the asset.

For technical detail, refer to **fp\_assigns\_to\_group (asset)**

	Collect together the elements that will comprise the asset group
	This group will be defined as an asset (a specific type of group) with globally unique identifier and owner history.
	A relationship object will be created that handles the assignment of the set of elements to the asset group. This relationship will have its own globally unique identifier and owner history.

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## 2 - Asset - Exchange Asset (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 14, 2007 ([view change](#))

Labels: (None)

## Asset

---

Provides for the collection of all information about the asset.

For technical detail, refer to **fp\_model\_asset**

All of the following information can be asserted through the functional part but is identified in sub-groups for convenience.

### Identification

	Set the globally unique identifier and owner history for the asset. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Set the value of the asset identifier. <i>Note that the asset identifier is not the same thing as the globally unique identifier of the asset.</i>
	Provide a 'familiar' name for the asset by which it may be further identified in the asset register. <i>Note that the name need not be unique so could be used as an informal way of defining asset supergroups.</i>
	Provide a description for the asset to further qualify the asset information within an asset register.

### Date

	Set the date on which the asset was incorporated into the project. This should include the day, month and year. <i>Incorporation is taken here to mean the date from which the asset started depreciating and will generally be considered to be the same as the date at which construction/ erection/ installation was completed.</i>
---	---

### Cost

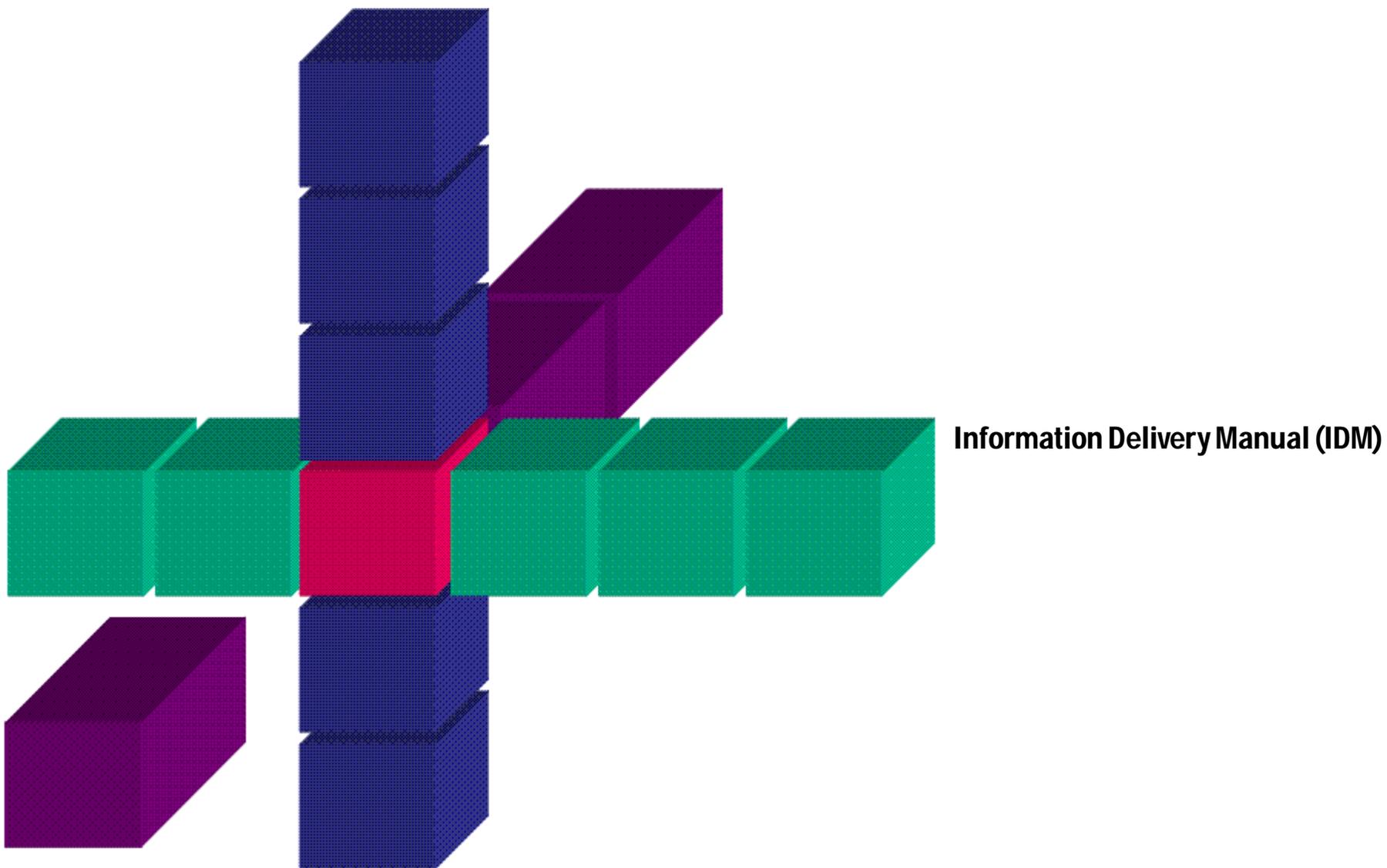
	Define the original value of the asset <i>Original value should be asserted as the value or cost of the asset at the time when the initial asset register entry was created.</i>
	Define the current value of the asset <i>Current value should be asserted as the cost of the asset if purchased at the present time.</i>
	Define the total replacement cost of the asset <i>Total replacement cost should be asserted as the cost of completely replacing the asset and including all work associated with such replacement.</i>
	Define the depreciated value of the asset <i>Depreciated value is the current value of the asset related to the date of incorporation and taking into account the discount rate over the intervening period.</i>

### Actor

	Identify the owner of the asset <i>The owner may be either a person or an organization.</i>
	Identify the user of the asset <i>This should be the primary user of the asset which may be an organization (lessee, tenant etc.), an organizational unit (department, division etc.), or an individual person.</i>
	Identify the person responsible for the asset <i>Typically, this is the point of contact for the asset and is a person.</i>

## Location

	The location of an asset is determined by the location of the elements that make up the asset. At least one of the elements in an asset group must therefore have an asserted location.
---	---



### 3 - Optional Information - Exchange Asset (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 14, 2007 ([view change](#))

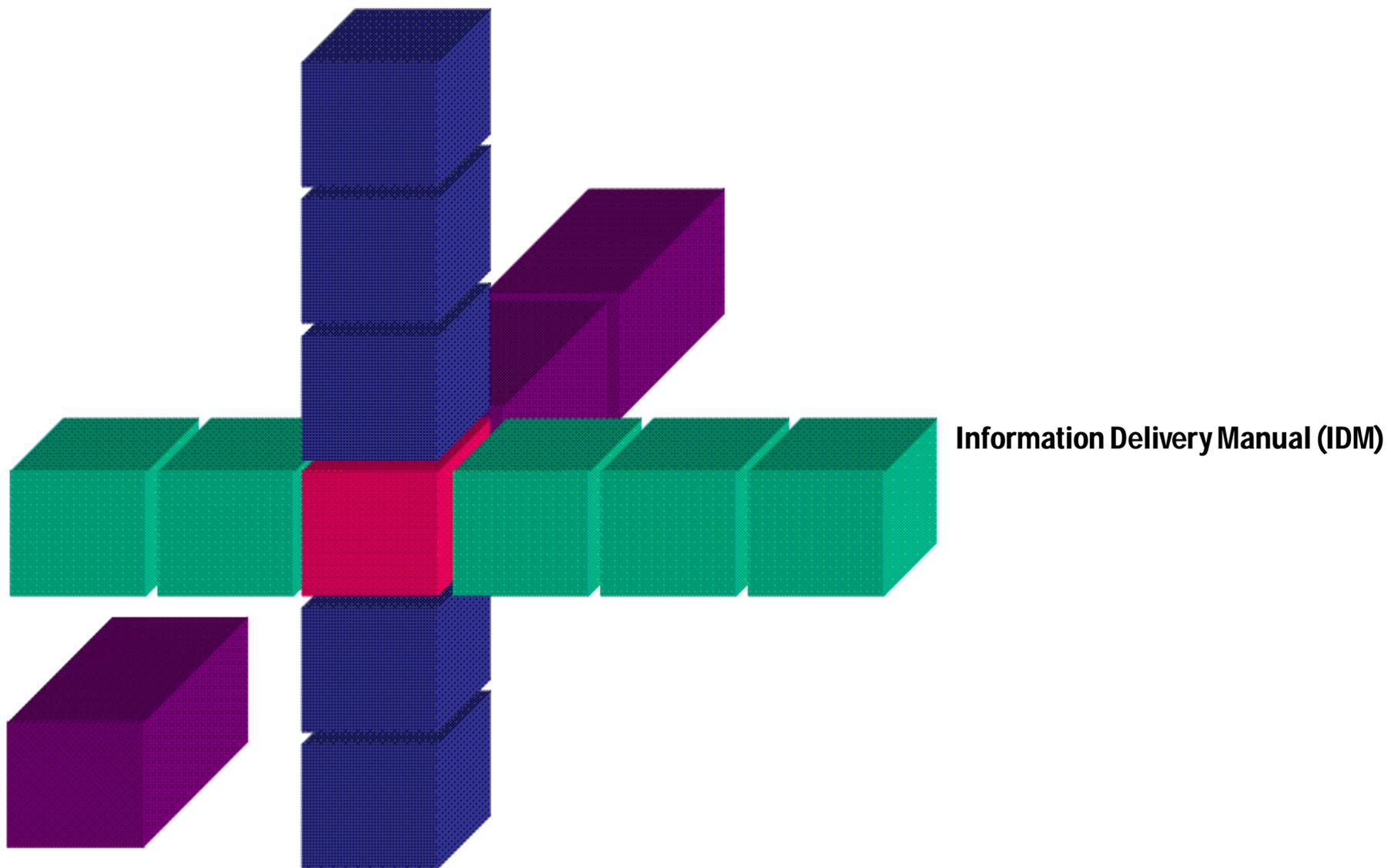
Labels: (None)

## Optional Information

---

1. Further accounting information may also be provided for the asset but is not a specific requirement for exchange at this point. A client may make accounting information a requirement through provision of a locally extended exchange requirement.
2. Additional cost information may also be provided through the general cost value assignment capability.
3. Whilst this exchange requirement identifies the specific set of information that may be contractually required at a point in the project, it should be borne in mind that information may be collected at earlier stages and information content within those earlier stages established as follows:

	<b>Design</b>	<b>Construction (pre-handover)</b>	<b>Construction (handover)</b>
	<b>stages 4,5,6,7</b>	<b>stage 8</b>	<b>O and M - stage 9</b>
Asset identifier	✔	✔	✔
Name	✔	✔	✔
Description	✔	✔	✔
Incorporation date		✔	✔
Original value		✔	✔
Current value			✔
Depreciated value			✔
Total replacement cost			✔
Owner	✔	✔	✔
User			✔
Person responsible			✔
Location	✔	✔	✔



## Exchange Asset Inventory (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 14, 2007 ([view change](#))

Labels: (None)

# Exchange Asset Inventory

---

## Project Stage

---

Stage	Name	Valid
0	Portfolio requirements	
1	Conception of need	
2	Outline feasibility	
3	Substantive feasibility	
4	Outline conceptual design	
5	Full conceptual design	
6	Coordinated design and procurement	
7	Production information	
8	Construction	
9	Operation and maintenance	
10	Disposal	

## Overview

---

The scope of this exchange requirement is to enable the exchange of information about an inventory that contains a list of assets. A requirement for the exchange of other inventory information may be defined elsewhere.

An inventory is simply a group of 'things' where the things are all of a particular type or from a restricted list of types.

- Information that can be included within an inventory includes:
- Identifier of the type of inventory (asset)
- The list of groups (assets) that comprise the inventory
- The organizational unit within whose jurisdiction the inventory falls
- The identity of the person whose responsibility it is to maintain the inventory
- The date on which the inventory was last updated
- The current value of the inventory (totalled from the value of the contents)
- The initial value of the inventory

## Information Requirements

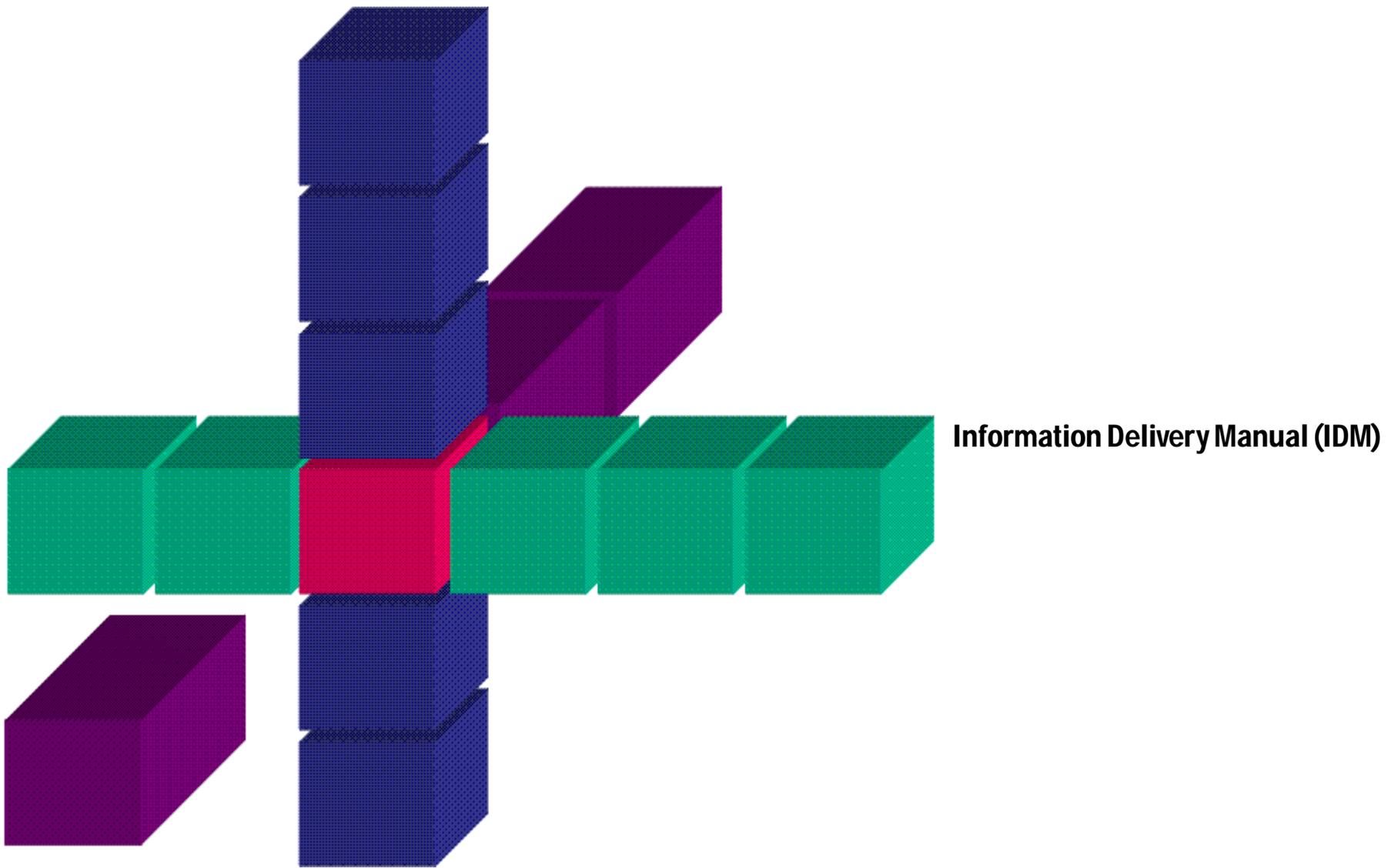
---

Context	Description
Precursor	All assets that are to be incorporated into the inventory are created and available. This means that the provisions of the exchange requirement <b>er_exchange_asset</b> have been met.
<a href="#">Inventory</a>	Creates the inventory in which assets will be recorded.
<a href="#">Asset to Inventory Assignment</a>	Provides for the assignment of assets to the inventory.

### Children [Hide Children](#) | [View in hierarchy](#)

---

-  [1 - Inventory- Exchange Asset Inventory\(ER\)](#) (Information Delivery Manual (IDM))
  -  [2 - Asset to Inventory Assignment- Exchange Asset Inventory \(ER\)](#) (Information Delivery Manual (IDM))
-



## 1 - Inventory- Exchange Asset Inventory(ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 14, 2007

Labels: (None)

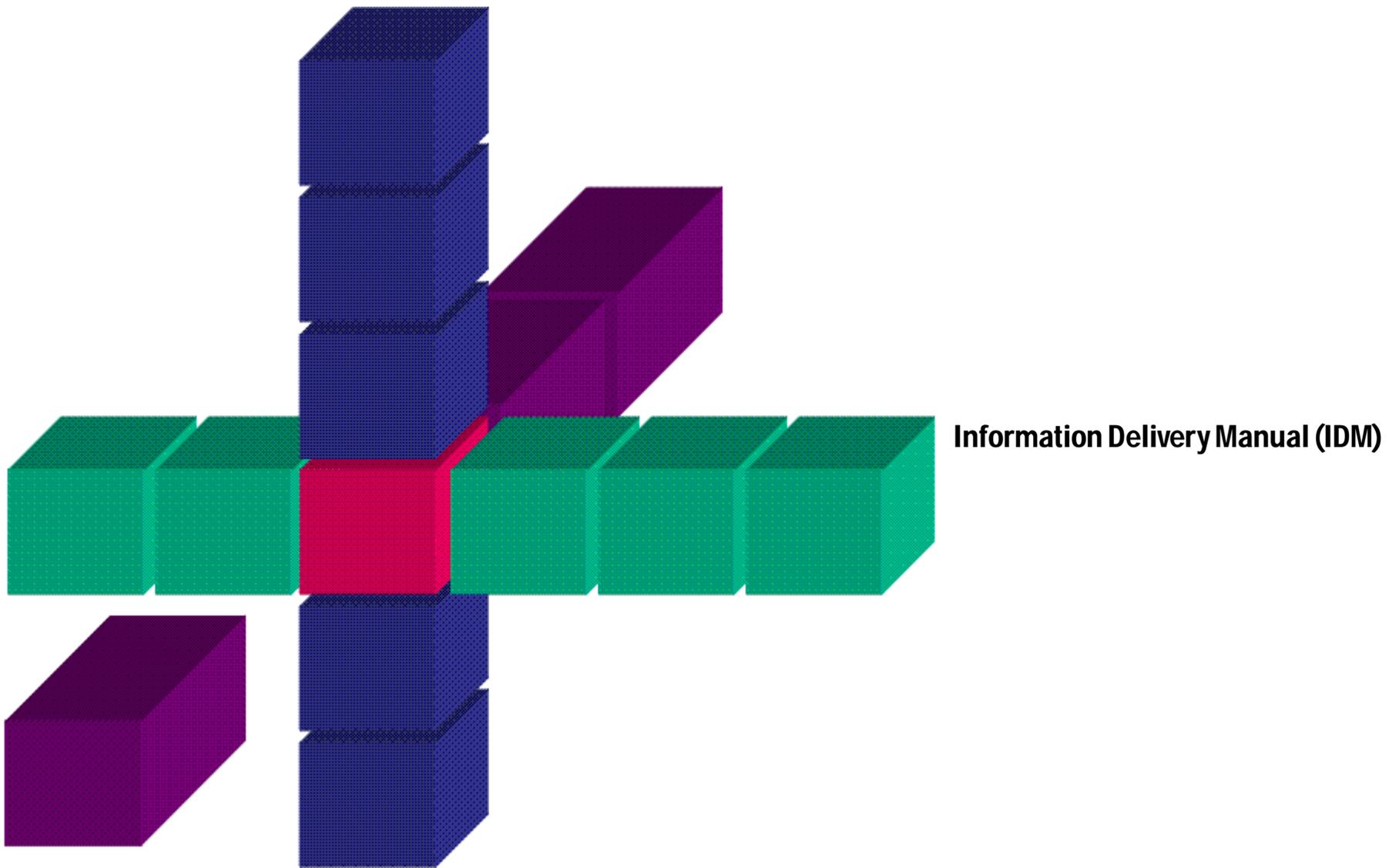
## Inventory

---

Creates the inventory in which assets will be recorded

For technical detail, refer to **fp\_model\_inventory(asset)**

	Define and name the inventory
	Set the globally unique identifier and owner history for the inventory. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Assert the type of inventory concerned to be an asset inventory.
	Define the organizational unit to which the inventory is applicable <i>Typically this is the organization or an organizational unit. It may however be a person.</i>
	Designate the person or persons who are responsible for the inventory.
	Record the date on which the last update of the inventory was carried out (day/month/year).
	Estimate the current cost value of the inventory.
	Estimate the original cost value of the inventory.



## 2 - Asset to Inventory Assignment- Exchange Asset Inventory (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 14, 2007 ([view change](#))

Labels: (None)

## Asset to Inventory Assignment

---

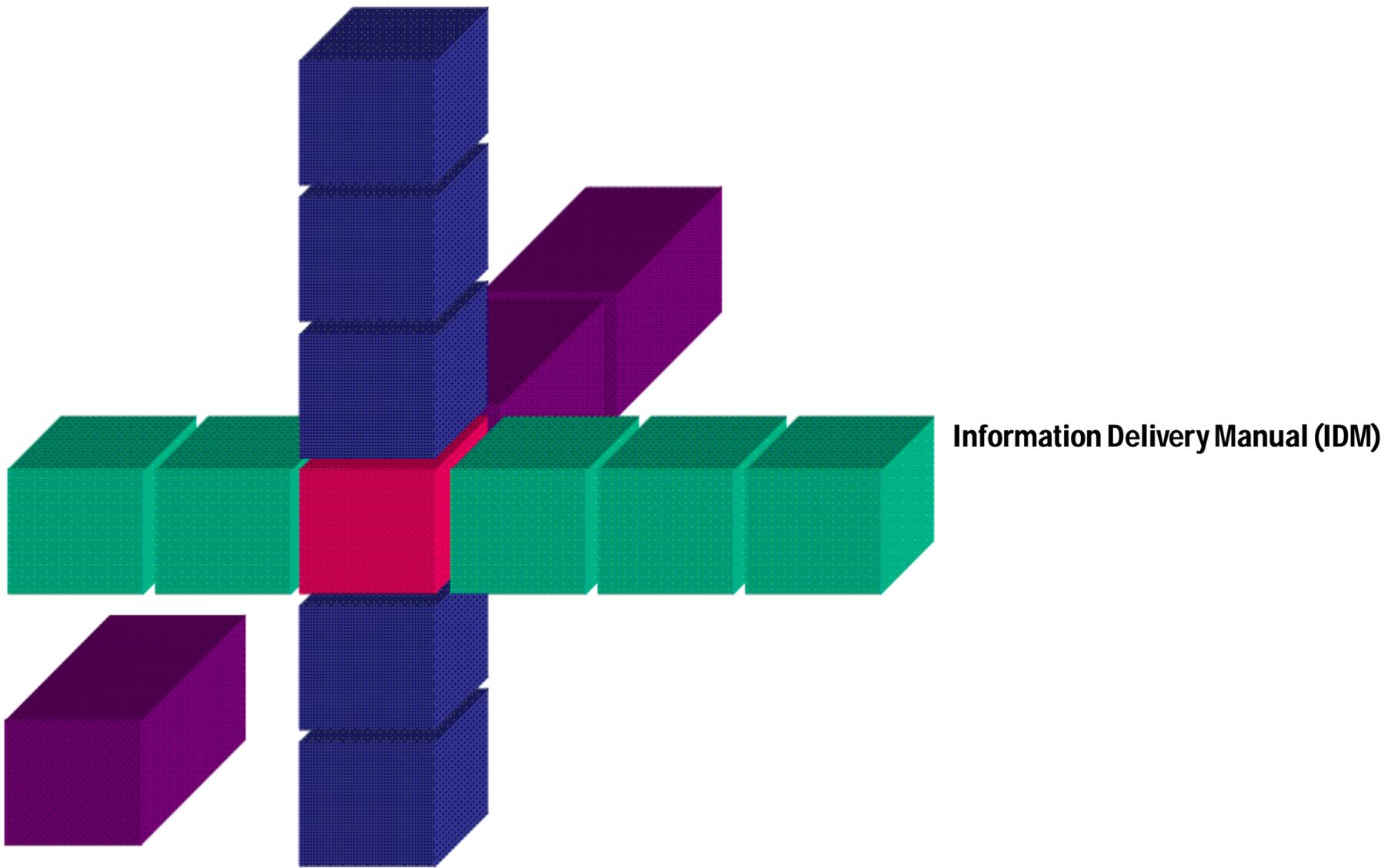
Provides for the assignment of assets to the inventory.

For technical detail, refer to **fp\_model\_inventory(asset)**

The asset(s) that are to be assigned to the inventory are already identified through the preconditions for this exchange requirement.

	Select the asset(s) that are to be recorded within the inventory.
	Set the globally unique identifier and owner history for the relationship between the asset(s) and the inventory. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Specify a name that can be used to identify the asset-inventory relationship.

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## Exchange Condition Assessment (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007 ([view change](#))

Labels: (None)

# Exchange Condition Assessment

---

## Project Stage

---

Stage	Name	Valid
0	Portfolio requirements	
1	Conception of need	
2	Outline feasibility	
3	Substantive feasibility	
4	Outline conceptual design	
5	Full conceptual design	
6	Coordinated design and procurement	
7	Production information	
8	Construction	
9	Operation and maintenance	
10	Disposal	

## Overview

---

The scope of this exchange requirement is to enable the exchange of information about the condition of an element or a group of elements (typically acting as an asset). Condition may be established to determine if an element requires maintenance in order to bring it to a required level of operating efficiency. Ultimately, the condition of an element may become so poor as to require replacement.

The condition of an element may be established by:

- instrumentation in place that enables operating parameters to be continuously monitored; the parameters being selected such that they indicate the current condition of the asset (e.g. vibration).
- periodic inspection that may be either visual, carried out with the assistance of instruments or a combination.

The condition of an element is considered to be determined by one or several condition criteria. Each criterion is a particular measured or assessed value that contributes to the overall condition of the element.

- A criterion that is measured using calibrated instruments and whose value is repeatable by different instruments is considered to be objective. An objective criterion will typically be a measured value.

- A criterion that is assessed by observation and whose value may not be repeatable between assessors is considered to be subjective. A subjective criterion will typically be measured against an arbitrary scale (e.g. 1 represents new condition whilst 10 represents poor condition requiring urgent replacement).

The actual condition of an element is considered to be determined by bringing together the various criteria within a group.

Particular target values may be established for maintenance or replacement operations to be carried out on an element. Such values are defined as constraints. When the condition of an element reaches or exceeds a value indicated by a constraint, it signifies that a particular action should be triggered. A condition based 'event' may be based on either objective or subjective criteria. For objective criteria, the potential exists for live reporting of condition data from fixed instrumentation.

Condition information about an element, group or asset needs to include:

- The element, group or asset whose condition is being assessed
- The condition criteria determining the overall condition including, for each criterion:
  - Identity and owner history
  - Name and description
  - Date and time at which the criterion is measured or assessed
  - The value of the measured or assessed criterion
- For the grouping of criteria as a condition assigned to the element, group or asset
  - Identity and owner history
  - Name and description of then group

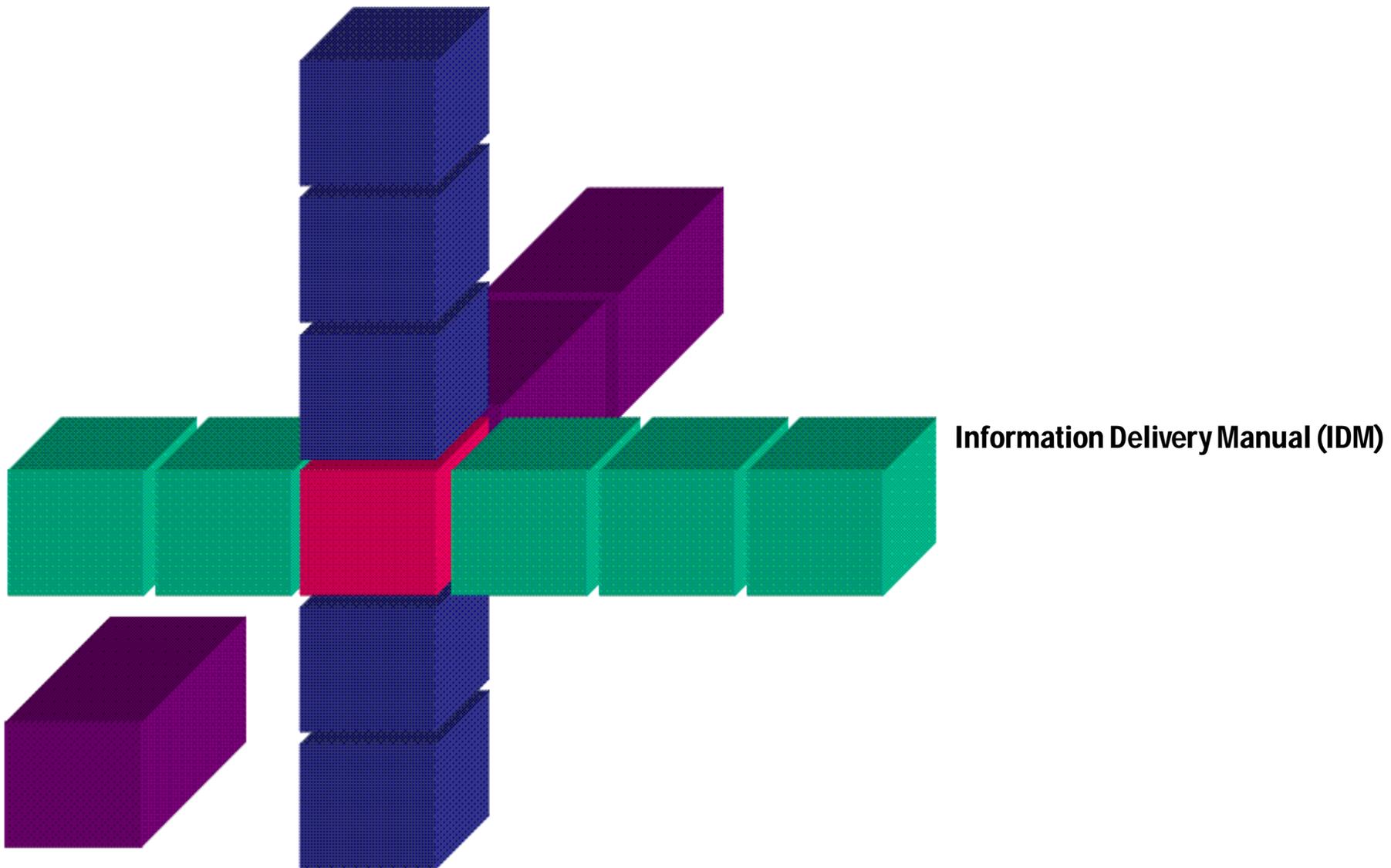
## Information Requirements

Context	Description
Precursor	All elements, groups or assets whose condition is to be assessed are created and available. This means that the provisions of the exchange requirement ' <b>er_exchange_asset</b> ', or ' <b>er_exchange_system</b> ' or ' <b>er_exchange_product</b> ' have been met.
<a href="#">Criteria</a>	Establishes the name, description, type and value of each of the criteria that are to contribute to the overall condition.
<a href="#">Criteria to Condition Grouping</a>	The criteria that have been assessed are grouped into a single condition group for assignment to the element, group or asset.
<a href="#">Condition to Product Assignment</a>	Assigns the condition group to the element, group or asset whose condition is being assessed.

**Children** [Hide Children](#) | [View in hierarchy](#)

-  [1 - Criteria - Exchange Condition Assessment \(ER\)](#) (Information Delivery Manual (IDM))
-  [2 - Criteria to Condition Grouping - Exchange Condition Assessment \(ER\)](#) (Information Delivery Manual (IDM))
-  [3 - Condition to Product Assignment - Exchange Condition Assessment \(ER\)](#) (Information Delivery Manual (IDM))

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## 1 - Criteria - Exchange Condition Assessment (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007

Labels: (None)

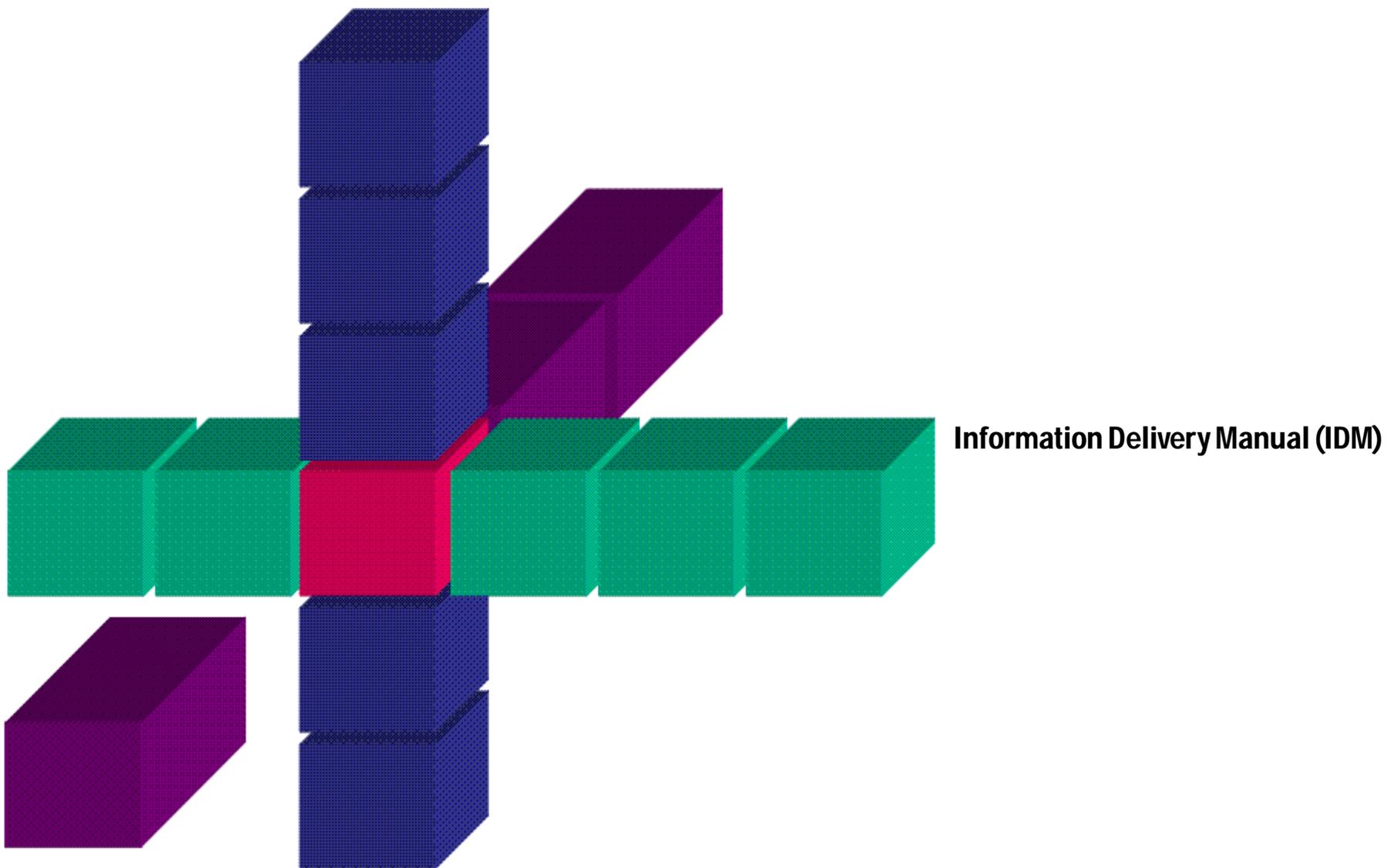
## Criteria

---

Establishes the name, description, type and value of each of the criteria that are to contribute to the overall condition. The following operations are applied to each operation.

For technical detail, refer to **fp\_model\_condition**

	Define and name the criterion <i>Note that it is generally recommended that criteria names should be defined in a list where their descriptions (definitions) can also be provided. Such a list entry would also define whether the condition should be assessed subjectively (in which case some guidance as to scales of measurement and meaning of values may be provided to secure accuracy of assessment between different assessors) or measured objectively (in which case the units of measure should also be identified).</i>
	Provide a description of the criterion in relation to the object whose condition is being assessed.
	Set the globally unique identifier and owner history for the criterion. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Set the date and time at which the condition assessment is made.
	Set the value of the criterion. <i>Note that if the value is objective, the value of the criterion should also include the units of measure.</i>



## 2 - Criteria to Condition Grouping - Exchange Condition Assessment (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007

Labels: (None)

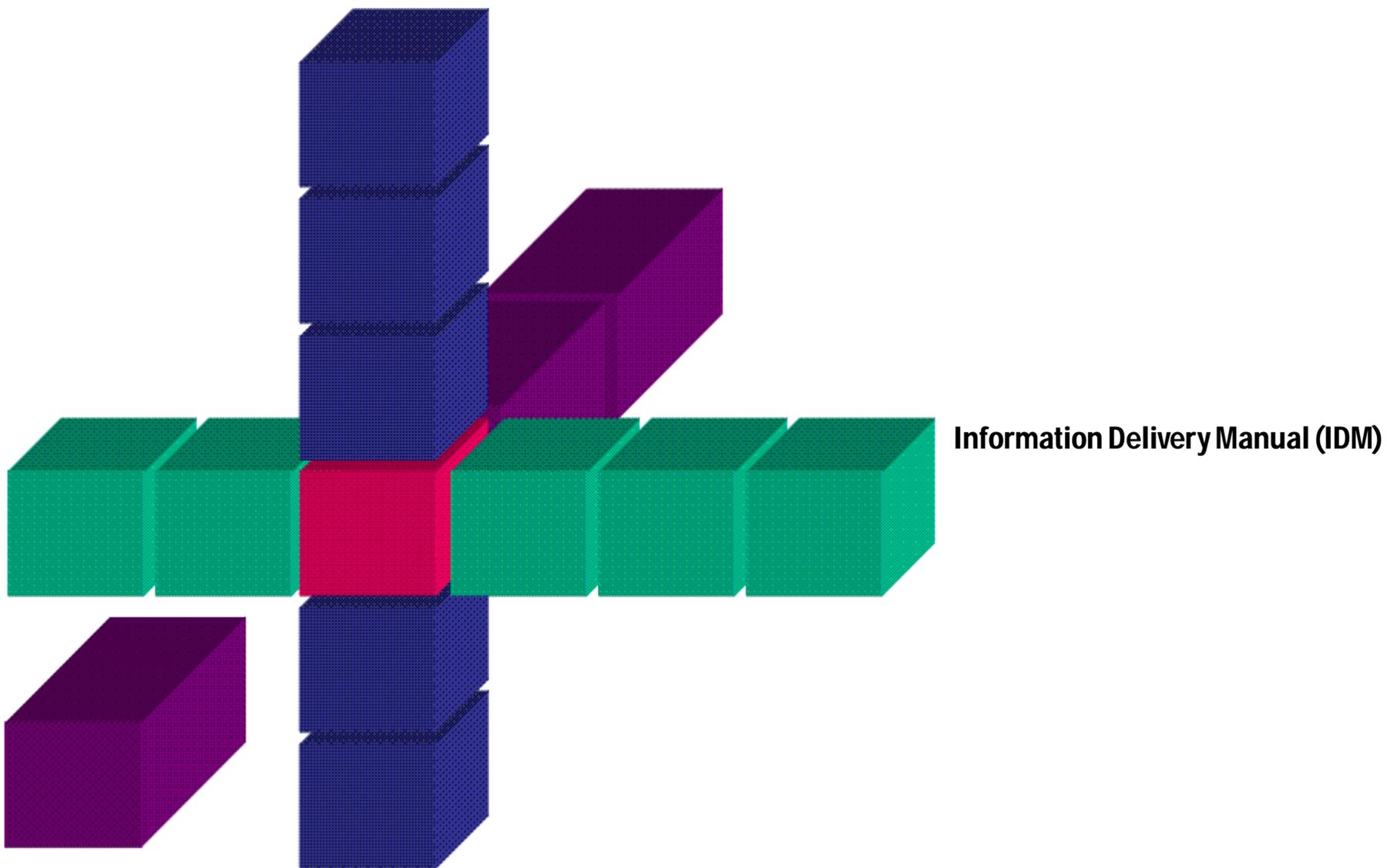
## Criteria to Condition Grouping

---

The criteria that have been assessed are grouped into a single condition group for assignment to the element, group or asset.

	Select the criteria that are to be grouped.
	Define the condition group to which the criteria are to be assigned.
	Set the globally unique identifier and owner history for the relationship between the criteria and the condition group. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Specify a name that can be used to identify the criteria-condition relationship.

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



### 3 - Condition to Product Assignment - Exchange Condition Assessment (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007

Labels: (None)

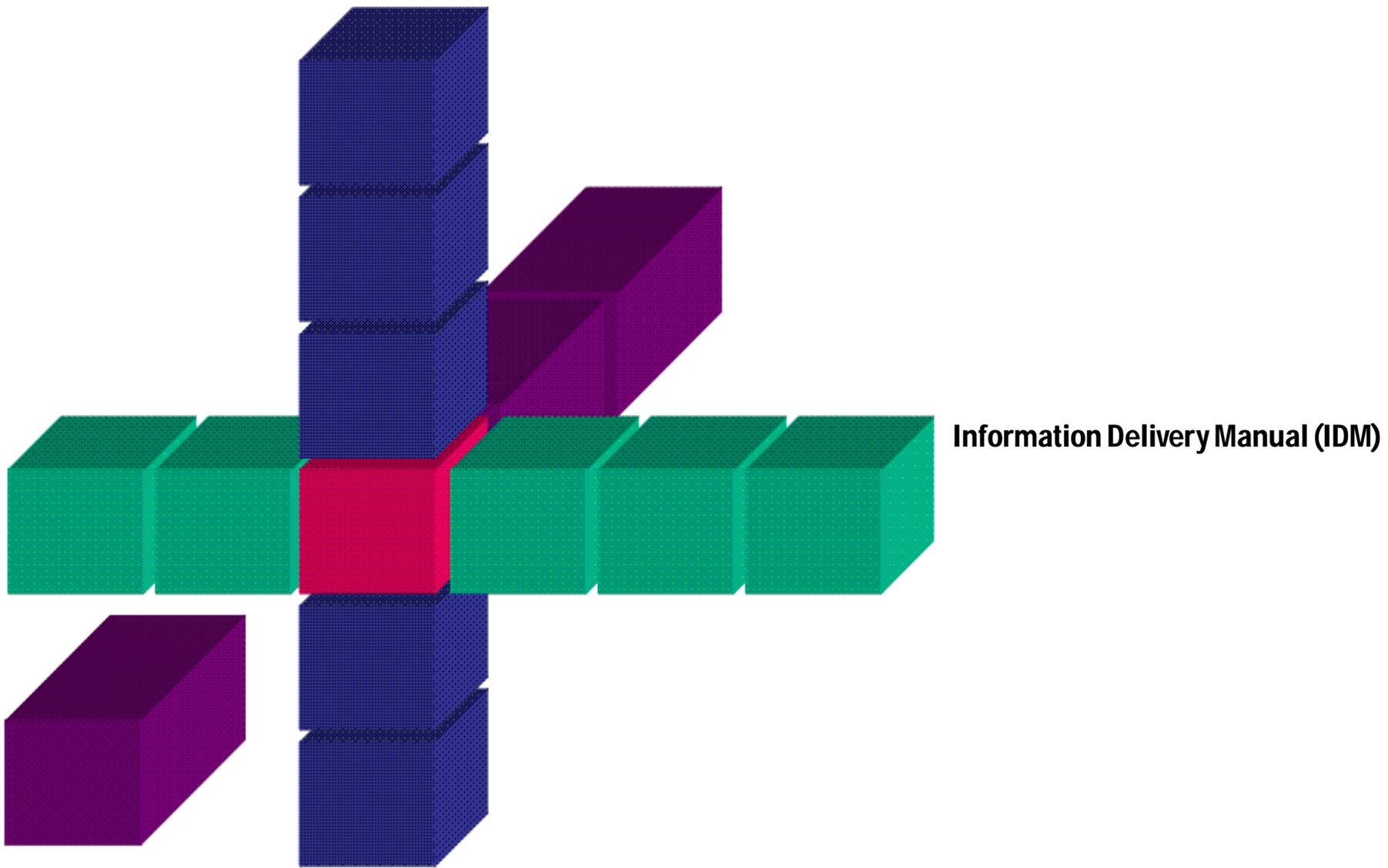
## Condition to Product Assignment

---

Assigns the condition group to the element, group or asset whose condition is being assessed.

	Set the globally unique identifier and owner history for the relationship between the criteria and the condition group. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Specify a name that can be used to identify the criteria-condition relationship.
	Select the item whose condition is being assessed and make it the product to which the condition is being assigned.
	Assign the condition group through the relationship to the product (element, group or asset).

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## Exchange Maintenance Plan (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007 ([view change](#))

Labels: (None)

# Exchange Maintenance Plan

---

## Project Stage

---

Stage	Name	Valid
0	Portfolio requirements	
1	Conception of need	
2	Outline feasibility	
3	Substantive feasibility	
4	Outline conceptual design	
5	Full conceptual design	
6	Coordinated design and procurement	✓
7	Production information	✓
8	Construction	✓
9	Operation and maintenance	✓
10	Disposal	

## Overview

---

The scope of this exchange requirement is to support the exchange of information concerning maintenance plans.

A maintenance plan is a type of work plan in which the work tasks that are planned are specifically related the assets that are to be maintained. That is, the maintenance plan will provide a listing of all the maintenance tasks that must be undertaken and the period that can elapse between successive performances of the same task. A maintenance plan does not identify the actual dates on which work should be undertaken. This is the function of a maintenance schedule and this is described elsewhere.. Additionally, a maintenance plan may define required resources to perform tasks in terms of the skills or capabilities required. It is not expected that a maintenance plan will identify actors by name.

For instance, the shaft bearings of a pump impeller may require lubrication once each year with the work being undertaken by a Class 1 Lubrication Specialist. This is a work description that can form the basis of a maintenance plan.

For a maintenance plan, the following information will be needed:

- The maintenance tasks that are to be undertaken with, for each task:
  - Identity of the task (using the general form of identification or breakdown structure adopted for the project)
  - The name of the task (i.e. how it will be shown in a chart)
  - Any further useful description (which may be provided if required).

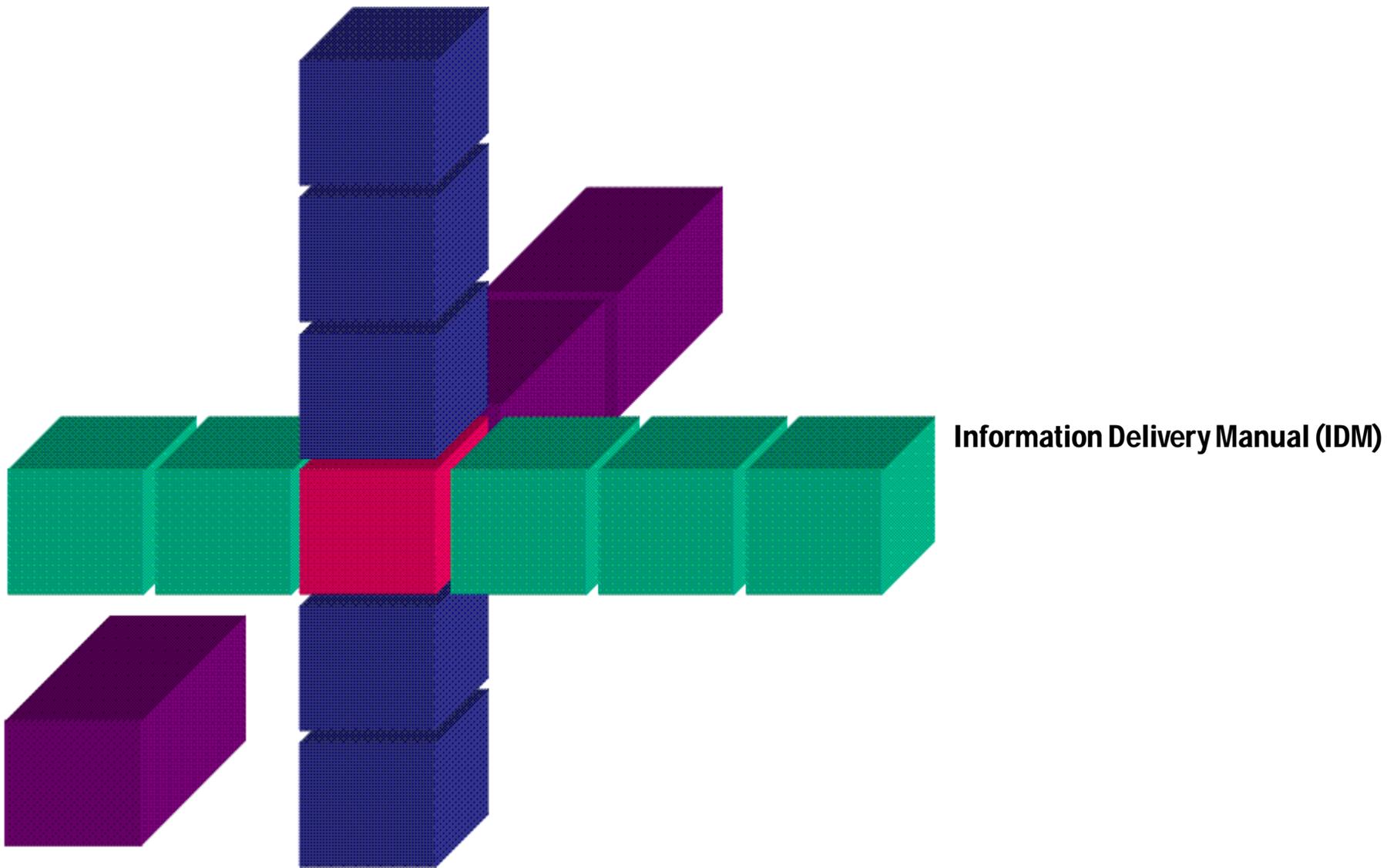
- Indication that this is not a milestone task (note that a task is required to be identified as to whether or not it is a milestone; this is not considered to be appropriate for a maintenance plan and the value should therefore be set to FALSE by default).
- The assets on which the work is done
- For each task on a particular asset, the period of time between occasions when the work is carried out.

## Information Requirements

Context	Description
Precursor	Entities that are to be subject to maintenance planning are already defined as assets. This means that the <b>er_exchange_asset</b> exchange requirement provisions will have been satisfied.
<a href="#">Maintenance Plan</a>	Provides for the capture of all information that defines the maintenance plan.
<a href="#">Nest</a>	Allows tasks to be contained (nested) within other tasks.
<a href="#">Resource</a>	Defines resources that may be associated with a task in a work plan.

### Children [Hide Children](#) | [View in hierarchy](#)

-  [1 - Maintenance Plan - Exchange Maintenance Plan \(ER\)](#) (Information Delivery Manual (IDM))
-  [2 - Nest - Exchange Maintenance Plan \(ER\)](#) (Information Delivery Manual (IDM))
-  [3 - Resource - Exchange Maintenance Plan \(ER\)](#) (Information Delivery Manual (IDM))



## 1 - Maintenance Plan - Exchange Maintenance Plan (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007 ([view change](#))

Labels: (None)

## Maintenance Plan

---

Provides for the capture of all information that defines the maintenance plan.

For technical detail, refer to **fp\_control\_maintenance\_plan**.

All of the following information can be asserted through the functional part but is identified in sub-groups for convenience.

### Task

	Set the globally unique identifier and owner history for the task. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Specify a name for the task that can be used as its representation in a plan or schedule.
	Set the value of the task identifier. <i>This is locally unique (within the maintenance plan), that may be used as a sequence number in a list of tasks 1,2,3,4 etc. or as work breakdown structure identifier.</i>
	Identify if the task is a milestone. <i>Although this value must be asserted, it is not relevant to the maintenance plan. To ensure that it is not taken into consideration, it must be asserted as FALSE for a maintenance plan.</i>
<b>X</b>	The status of a task MUST NOT be used in a maintenance plan.
<b>X</b>	The method of work for a task MUST NOT be used in a maintenance plan.
<b>X</b>	The relative priority of a task MUST NOT be used in a maintenance plan.

### Plan Period

	Set the unit time measure to be applied to the period between occurrences of tasks in the workplan <i>e.g. the time measure for lubrication might be 'monthly'.</i>
	Set the number of units of the time measure that specifies the period between recurring occurrences of the task <i>e.g. '2' for 2 months.</i>

### Asset-Task Assignment

The asset(s) to which tasks are to be assigned are already identified through the preconditions for this exchange requirement.

	Select the asset(s) to which the task is to be assigned.
	Select the tasks that are to be assigned to the asset.

	Set the globally unique identifier and owner history for the relationship between the asset(s) and the task. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Specify a name that can be used to identify the asset-task relationship. <i>For example 'Change filter in AHU1' gives the task and the name of the asset on which it is carried out.</i>
<b>X</b>	The quantity of the an object specific for the operation by this task MUST NOT be used in a maintenance plan.

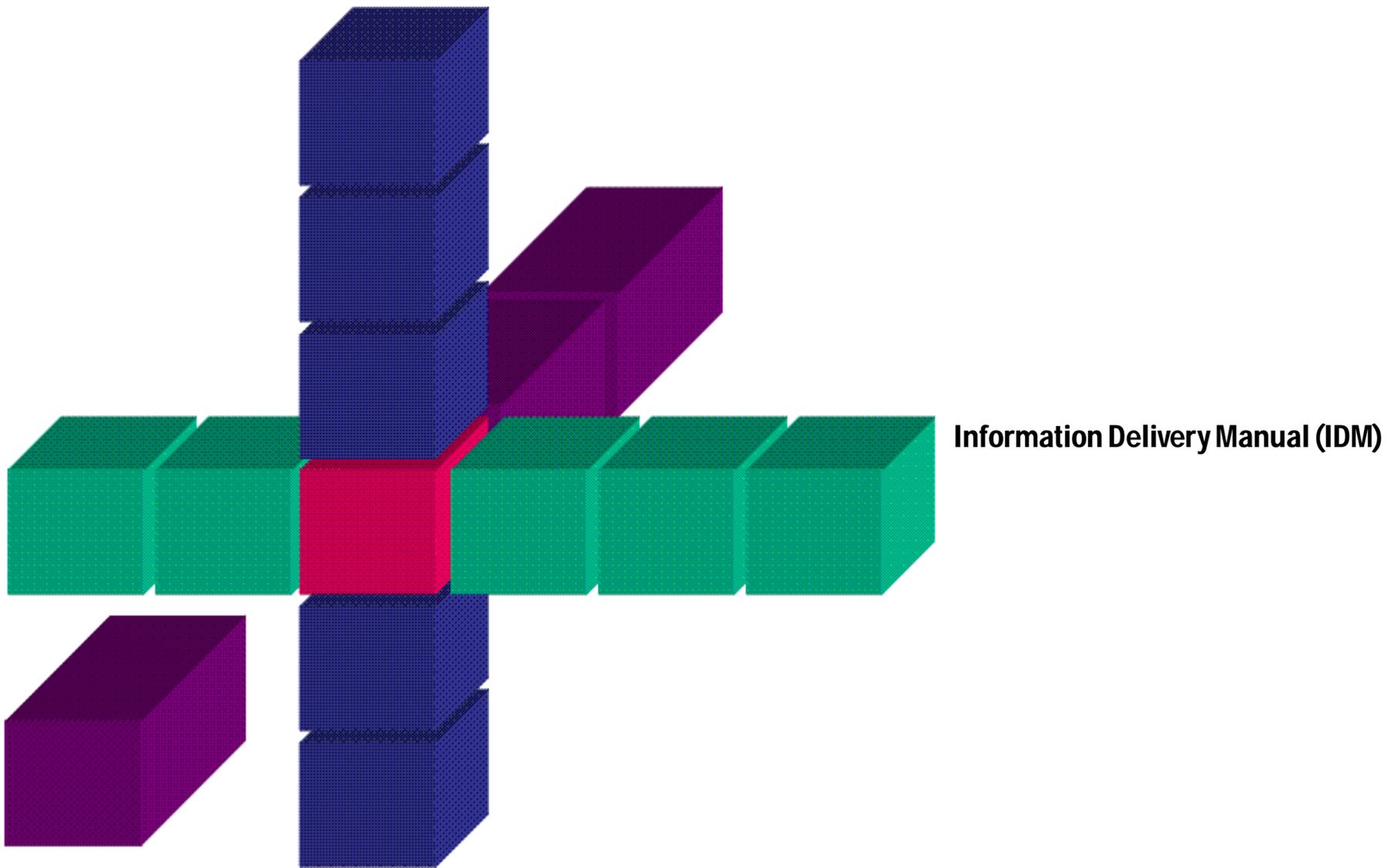
## WorkPlan

	Set the globally unique identifier and owner history for the work plan. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Specify a name for the work plan that provides a 'human' readable identification.
	Set the value of the work plan identifier. <i>This may be used in for 'human' readable identification in conjunction with the name.</i>
	Set the date on which the work plan was created or modified.
	Set the name of the authors of the work plan.
	Decide if the work control type is baseline, planned or actual (or some other user defined value in which case, the user defined value must also be specified).
<b>X</b>	The total duration of the entire work schedule MUST NOT be used in a maintenance plan.
<b>X</b>	The total time float of the entire work schedule MUST NOT be used in a maintenance plan.
<b>X</b>	The start time of the schedule MUST NOT be used in a maintenance plan.
<b>X</b>	The finish time of the schedule MUST NOT be used in a maintenance plan.

## WorkPlan -Task Assignment

	Select the work plan to which the tasks are to be assigned.
	Select the tasks that are to be assigned to the work plan.
	Set the particular type of the work plan/task assignment relationship which must be PROCESS.
	Set the globally unique identifier and owner history for the relationship between the work plan and the tasks. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Specify a name that can be used to identify the work plan-task relationship.
<b>X</b>	The time related information for the task MUST NOT be used in a maintenance plan.

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## 2 - Nest - Exchange Maintenance Plan (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007

Labels: (None)

## Nest

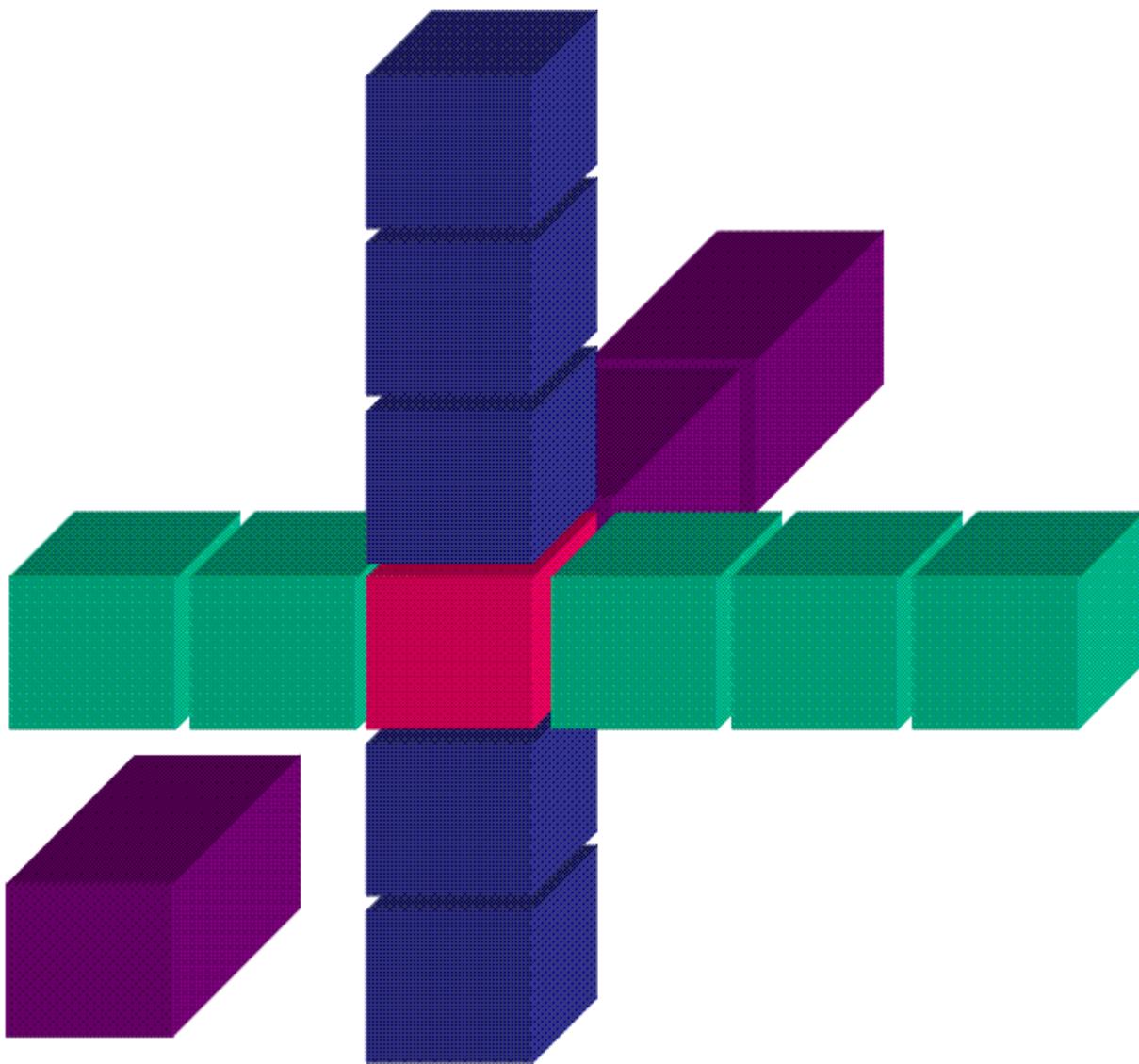
---

Allows tasks to be contained (nested) within other tasks.

For technical detail, refer to **fp\_nests (task)**

	Collect together the tasks that are to be nested (child tasks).
	Define or set the parent task into which the child tasks are to be nested and ensure that it has a globally unique identifier and owner history.
	A relationship object will be created that handles the nesting of the tasks. This relationship will have its own globally unique identifier and owner history.

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



**Information Delivery Manual (IDM)**

### 3 - Resource - Exchange Maintenance Plan (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007

Labels: (None)

## Resource

---

Define resources that may be associated with a task in a work plan

For technical detail, refer to **fp\_assign\_construction\_resource\_to\_task**

Different types of resource are handled through different concepts in this functional part. Each type resource or concept is identified by type below. However, globally unique identification and owner history is set for each resource as:

	Set the globally unique identifier and owner history for the relationship between the construction resource and the task. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Identify the resource.
	Identify the resource group (type) <i>e.g. as plant hire, rental etc.</i>
	Set the extent of the resource consumption <i>e.g. wholly/partially/not occupied.</i>
	Set the base quantity in which the equipment resource is measured.

### Construction Equipment Resource

- Resource may be identified as article number, RFID tag etc.
- Resource group may be identified as plant hire, rental etc.
- Consumption should be wholly/partially/not occupied

### Construction Material Resource

- Resource may be identified as article number, buying code etc.
- Resource group may be identified as sand, gravel etc.
- Consumption should be wholly/partially/not consumed

### Construction Product Resource

- Resource may be identified as article number, buying code etc.
- Resource group may be referenced from the product catalogue etc.
- Consumption should be wholly/partially/not consumed or occupied (depending on the product)

### Labor Resource

- Resource may be identified by payroll number, social security number etc.
- Resource group may be identified by the general actor function such as carpenter, plumber etc. (may be similar to the actor role specification)
- Consumption should be wholly/partially/not occupied

Additional attributes include:

 Specify the skillset used by the labor resource.

### Subcontract Resource

- Resource may be identified by contractor identifier etc.
- Resource group may be identified by the type of subcontract resource (bar bending, insulation etc.)
- Consumption should be wholly/partially/not occupied or consumed
- Base quantity for this purpose is the subcontract unit and is therefore likely to be 1.

Additional attributes include:

 Identify the actor performing the role of the subcontracted resource.

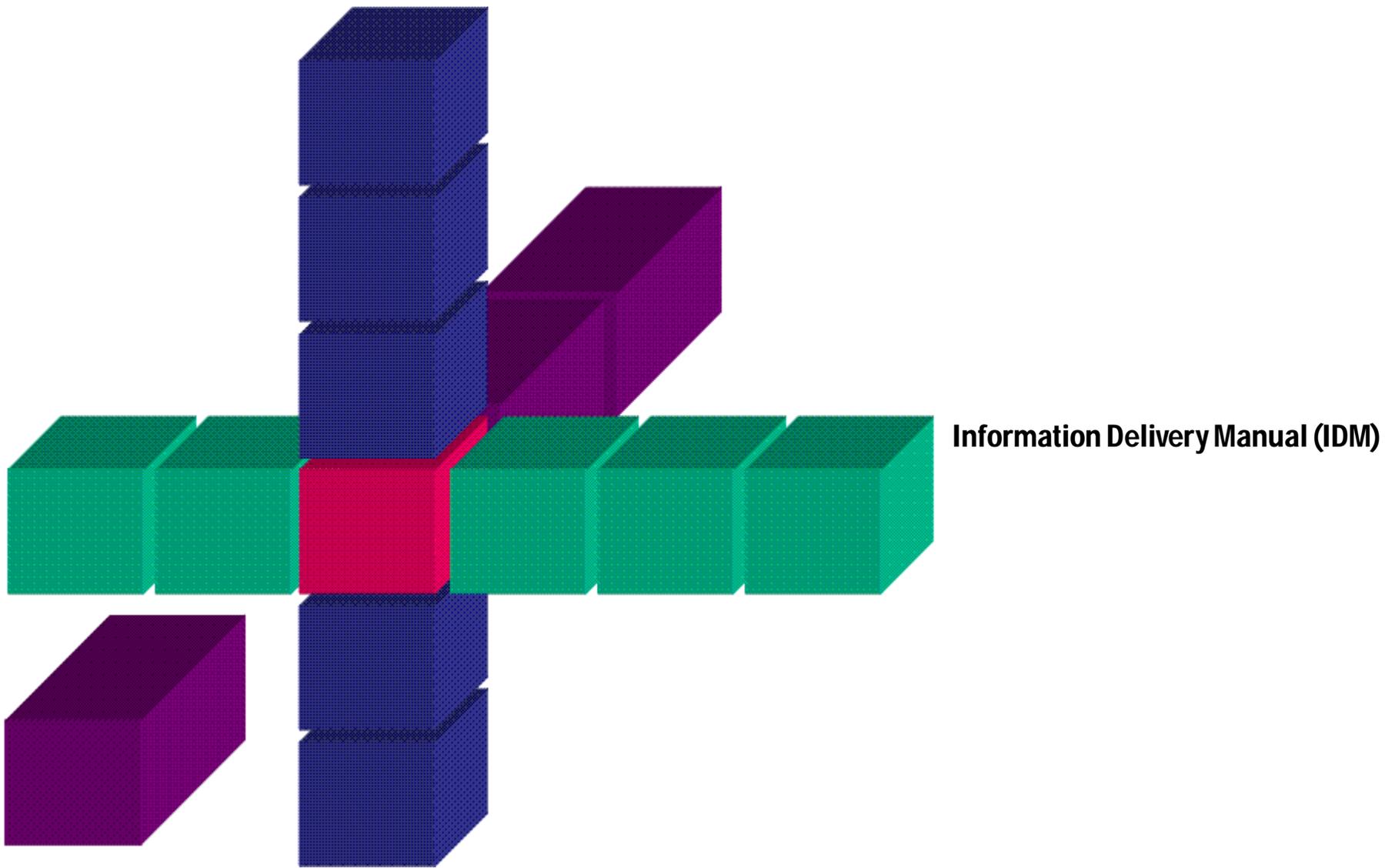
 Describe the jobs that the subcontract should complete.

### Crew Resource

- Resource may be identified by whatever means are appropriate etc.
- Resource group may be identified by whatever means are appropriate
- Consumption should be wholly/partially/not occupied or consumed
- Base quantity for this purpose is the crew unit and is therefore likely to be 1.

Additional attributes include:

 The aggregation of the construction resources that make up the crew (typically labor and equipment).



## Exchange Maintenance Schedule (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007 ([view change](#))

Labels: (None)

# Exchange Maintenance Schedule

---

## Project Stage

---

Stage	Name	Valid
0	Portfolio requirements	
1	Conception of need	
2	Outline feasibility	
3	Substantive feasibility	
4	Outline conceptual design	
5	Full conceptual design	
6	Coordinated design and procurement	
7	Production information	
8	Construction	
9	Operation and maintenance	
10	Disposal	

## Overview

---

The scope of this exchange requirement is to support the exchange of information concerning maintenance schedules.

A maintenance schedule is a type of work schedule in which the work tasks that are scheduled are specifically related to the assets that are to be maintained. That is, the maintenance schedule will describe all the maintenance tasks that must be undertaken and the actual dates on which work is scheduled to be done.

Additionally, a maintenance schedule may define required resources to perform tasks in terms of the skills or capabilities required and/or in terms of specific actors.

For a maintenance schedule, the following information will be needed:

- The maintenance tasks that are to be undertaken with, for each task:
  - Identity of the task (using the general form of identification or breakdown structure adopted for the project)
  - The name of the task (i.e. how it will be shown in a chart)
  - Any further useful description (which may be provided if required).
  - Whether the task is a milestone or has a scheduled duration.
  - Time related information about the task including earliest/latest start dates, earliest/latest finish dates, scheduled start/finish dates, actual start/finish dates.  
The extent of time related information specified depends on whether the baseline schedule or actual concept schedule is applied.
  - Available float times for the task.

- The assets on which the work is done

For schedule time, two schedule concepts are identified. These relate to respectively a planned or baseline schedule in which schedule times are proposed and an actual schedule in which times actually taken are retrospectively entered into the schedule. The concepts are:

#### **Concept 1: Baseline Schedule**

Includes earliest, latest and scheduled start and finish times. Float times may also be included. Actual start and finish times are excluded together with other values that are dependant on start or finish time.

#### **Concept 2: Actual Schedule**

Actual start and finish times are included, earliest, latest and scheduled start/finish times may be adjusted according to circumstances. Float times may also be adjusted. Remaining time for a task may be identified.

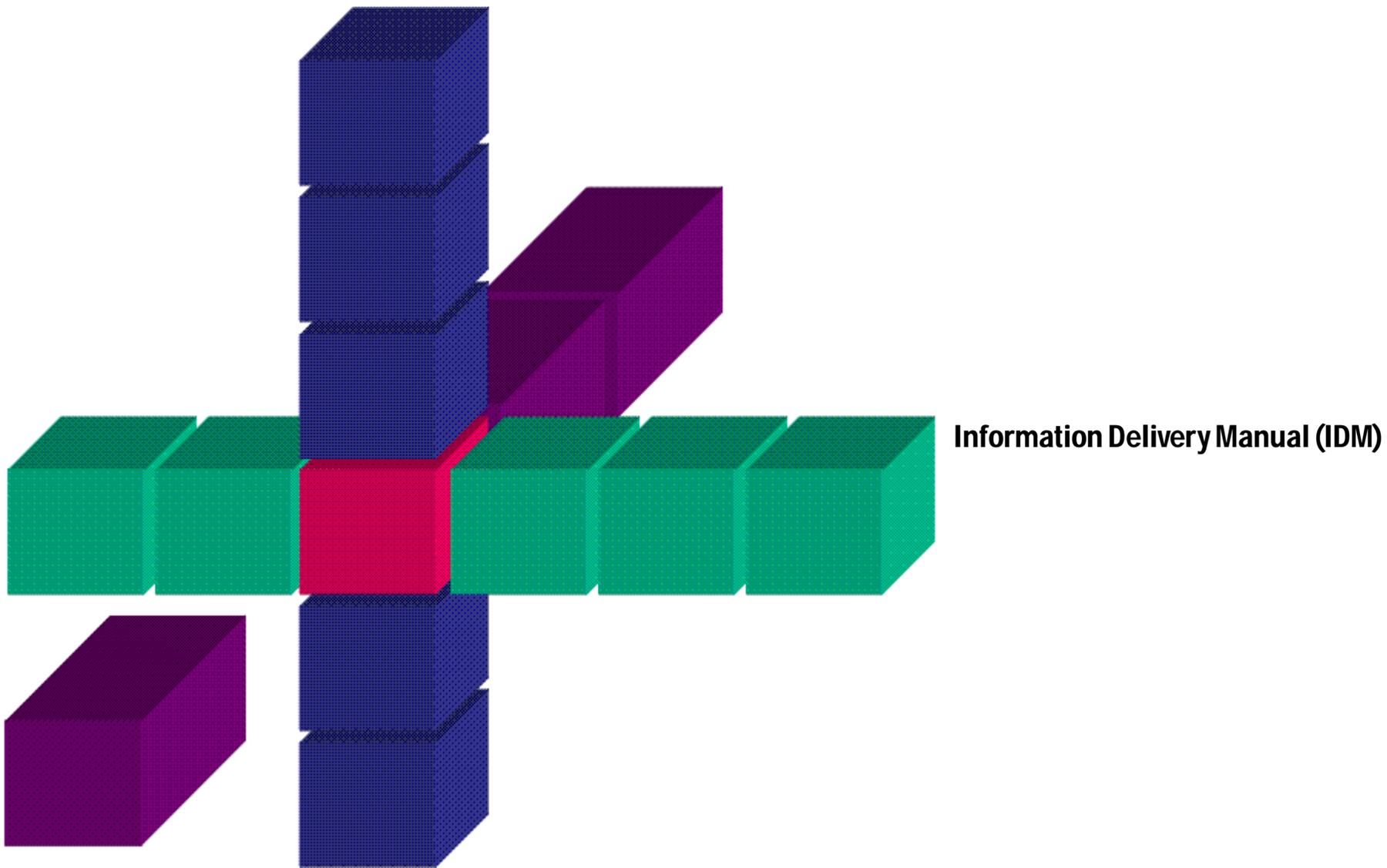
Note that it is expected that a baseline schedule will have been created before the actual schedule is composed so that there can be a comparison of actual versus proposed performance.

## Information Requirements

Context	Description
Precursor	Entities that are to be subject to maintenance scheduling are already defined as assets. This means that the <b>er_exchange_asset</b> exchange requirement provisions will have been satisfied. It is also expected that a maintenance plan will also have been defined through the exchange requirement <b>er_exchange_maintenance_plan</b> . This will facilitate prior identification of tasks. Where this has been done, tasks identified in the maintenance plan can be promoted to the maintenance schedule. However, it is not a mandatory precondition of a maintenance schedule that a maintenance plan should have been defined.
<a href="#">Maintenance Schedule</a>	Provides for the capture of all information that defines the maintenance schedule.
<a href="#">Nest</a>	Allows tasks to be contained (nested) within other tasks.
<a href="#">Resource</a>	Defines resources that may be associated with a task in a work schedule.

### Children [Hide Children](#) | [View in hierarchy](#)

-  [1 - Maintenance Schedule - Exchange Maintenance Schedule \(ER\)](#) (Information Delivery Manual (IDM))
-  [2 - Nest - Exchange Maintenance Schedule \(ER\)](#) (Information Delivery Manual (IDM))
-  [3 - Resource - Exchange Maintenance Schedule \(ER\)](#) (Information Delivery Manual (IDM))



## 1 - Maintenance Schedule - Exchange Maintenance Schedule (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007 ([view change](#))

Labels: (None)

## Maintenance Schedule

---

Provides for the capture of all information that defines the maintenance schedule.

For technical detail, refer to **fp\_control\_maintenance\_plan**.

All of the following information can be asserted through the functional part but is identified in sub-groups for convenience.

### Task

	Set the globally unique identifier and owner history for the task. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Specify a name for the task that can be used as its representation in a plan or schedule.
	Set the value of the task identifier. <i>This is locally unique (within the maintenance plan), that may be used as a sequence number in a list of tasks 1,2,3,4 etc. or as work breakdown structure identifier.</i>
	Identify if the task is a milestone. <i>Although this value must be asserted, it is not relevant to the maintenance plan. To ensure that it is not taken into consideration, it must be asserted as FALSE for a maintenance plan.</i>
	Set the current status of the task.
	Specify the method of work to be used in carrying out the task.
	Set the relative priority of the task (in comparison to the priorities of other tasks).

### Asset-Task Assignment

The asset(s) to which tasks are to be assigned are already identified through the preconditions for this exchange requirement.

	Select the asset(s) to which the task is to be assigned.
	Select the tasks that are to be assigned to the asset.
	Set the globally unique identifier and owner history for the relationship between the asset(s) and the task. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Specify a name that can be used to identify the asset-task relationship. <i>For example 'Change filter in AHU1' gives the task and the name of the asset on which it is carried out.</i>
	The quantity of the an object specific for the operation by this task MUST NOT be used in a maintenance plan.

## Work Schedule

✔	Set the globally unique identifier and owner history for the work schedule. <i>These should be handled directly by the software application and should not require action by the user.</i>
✔	Specify a name for the work schedule that provides a 'human' readable identification.
✔	Set the value of the work schedule identifier. <i>This may be used in for 'human' readable identification in conjunction with the name.</i>
✔	Set the date on which the work schedule was created or modified.
✔	Set the name of the authors of the work schedule.
✔	Decide if the work control type is baseline, planned or actual (or some other user defined value in which case, the user defined value must also be specified).
✔	Set the total duration of the entire work schedule.
✔	Set the total time float of the entire work schedule.
✔	Set the start time of the schedule.
✔	Set the finish time of the schedule.

## Work Schedule - Task Assignment

✔	Select the work schedule to which the tasks are to be assigned.
✔	Select the tasks that are to be assigned to the work schedule.
✔	Set the particular type of the work schedule/task assignment relationship which must be PROCESS.
✔	Set the globally unique identifier and owner history for the relationship between the work schedule and the tasks. <i>These should be handled directly by the software application and should not require action by the user.</i>
✔	Specify a name that can be used to identify the work schedule-task relationship.
✔	Set the time related information for either the BASELINE or the ACTUAL schedule as set out below.

## Baseline Schedule

✔	Set the earliest date on which a task can be started.
✔	Set the latest date on which a task can be started.
✔	Set the date on which a task is scheduled to be started.
✔	Set the earliest date on which a task can be finished.

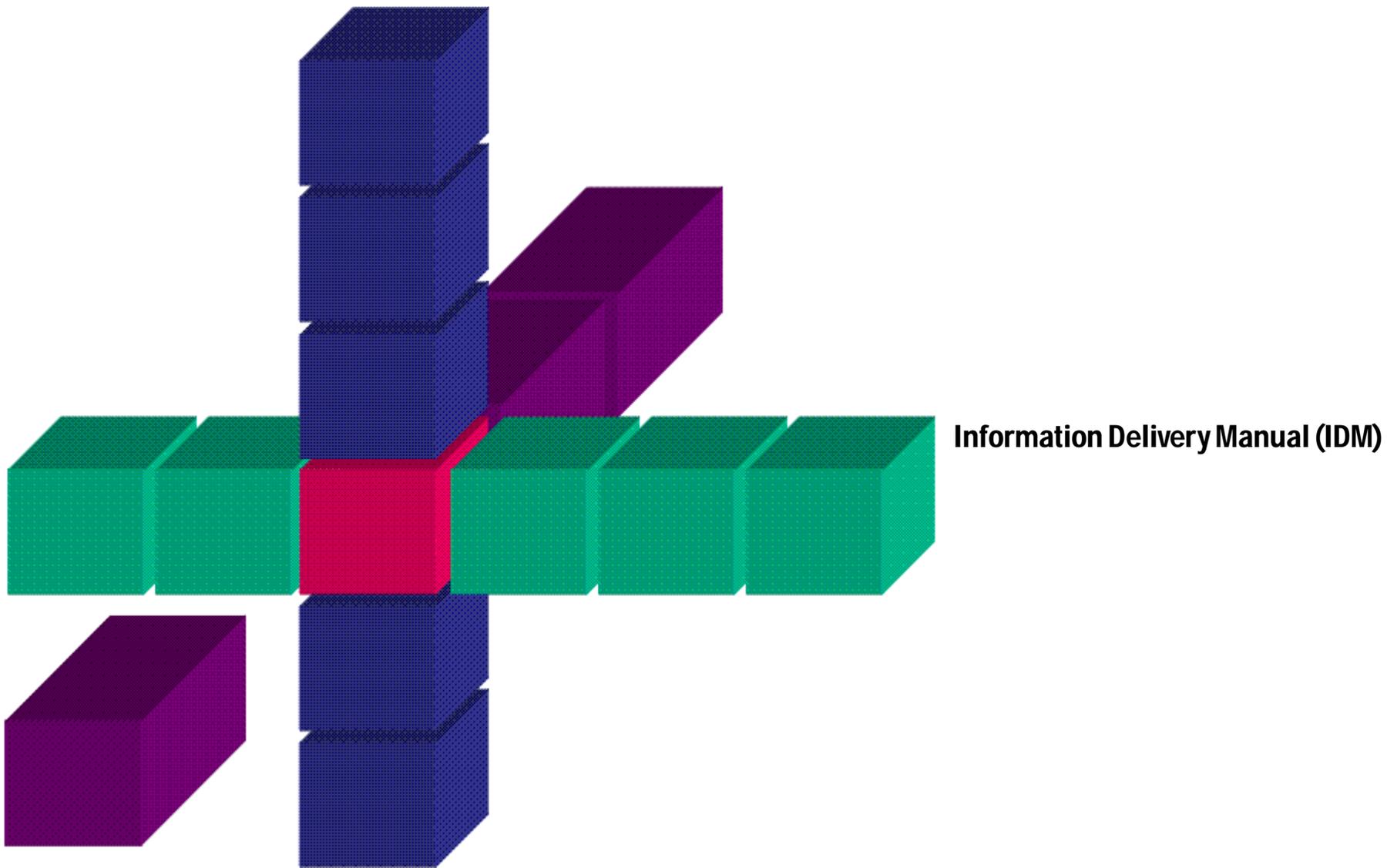
<input checked="" type="checkbox"/>	Set the latest date on which a task can be finished.
<input checked="" type="checkbox"/>	Set the date on which a task is scheduled to be finished.
<input checked="" type="checkbox"/>	Set the amount of time which is scheduled for completion of a task.
<input checked="" type="checkbox"/>	Set the amount of time remaining to complete a task.
<input checked="" type="checkbox"/>	Set the amount of time during which the start or finish of a task may be varied without any effect on the overall programme of work.
<input checked="" type="checkbox"/>	Set the difference between the duration available to carry out a task and the scheduled duration of the task.
<input checked="" type="checkbox"/>	Set a flag which identifies whether a scheduled task is a critical item within the program.
<input checked="" type="checkbox"/>	Set the difference between the late start and early start of a task.
<input checked="" type="checkbox"/>	Set the difference between the late finish and early finish of a task.
<input checked="" type="checkbox"/>	Set the extent of completion expressed as a ratio or percentage. <i>This value must be set to 0 for a baseline schedule.</i>
<input type="checkbox"/>	The date on which a task is actually started CANNOT be set for a baseline maintenance schedule.
<input type="checkbox"/>	The date on which a task is actually finished CANNOT be set for a baseline maintenance schedule.
<input type="checkbox"/>	The actual duration of a task is actually started CANNOT be set for a baseline maintenance schedule.
<input type="checkbox"/>	The date or time at which the status of the tasks within the schedule is analyzed CANNOT be set for a baseline maintenance schedule.

## Actual Schedule

Note that values that have previously been set in a baseline schedule may be subject to variation in an actual schedule.

<input checked="" type="checkbox"/>	Set the earliest date on which a task can be started.
<input checked="" type="checkbox"/>	Set the latest date on which a task can be started.
<input checked="" type="checkbox"/>	Set the date on which a task is scheduled to be started.
<input checked="" type="checkbox"/>	Set the earliest date on which a task can be finished.
<input checked="" type="checkbox"/>	Set the latest date on which a task can be finished.
<input checked="" type="checkbox"/>	Set the date on which a task is scheduled to be finished.
<input checked="" type="checkbox"/>	Set the amount of time which is scheduled for completion of a task.
<input checked="" type="checkbox"/>	Set the amount of time remaining to complete a task.
<input checked="" type="checkbox"/>	Set the amount of time during which the start or finish of a task may be varied without any effect on the overall programme of work.

	Set the difference between the duration available to carry out a task and the scheduled duration of the task.
	Set a flag which identifies whether a scheduled task is a critical item within the program.
	Set the difference between the late start and early start of a task.
	Set the difference between the late finish and early finish of a task.
	Set the extent of completion expressed as a ratio or percentage. <i>This value must be set to 0 for a baseline schedule.</i>
	Set the date on which a task is actually started.
	Set the date on which a task is actually finished.
	Set the actual duration of a task is actually started.
	Set the date or time at which the status of the tasks within the schedule is analyzed.



## 2 - Nest - Exchange Maintenance Schedule (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007

Labels: (None)

## Nest

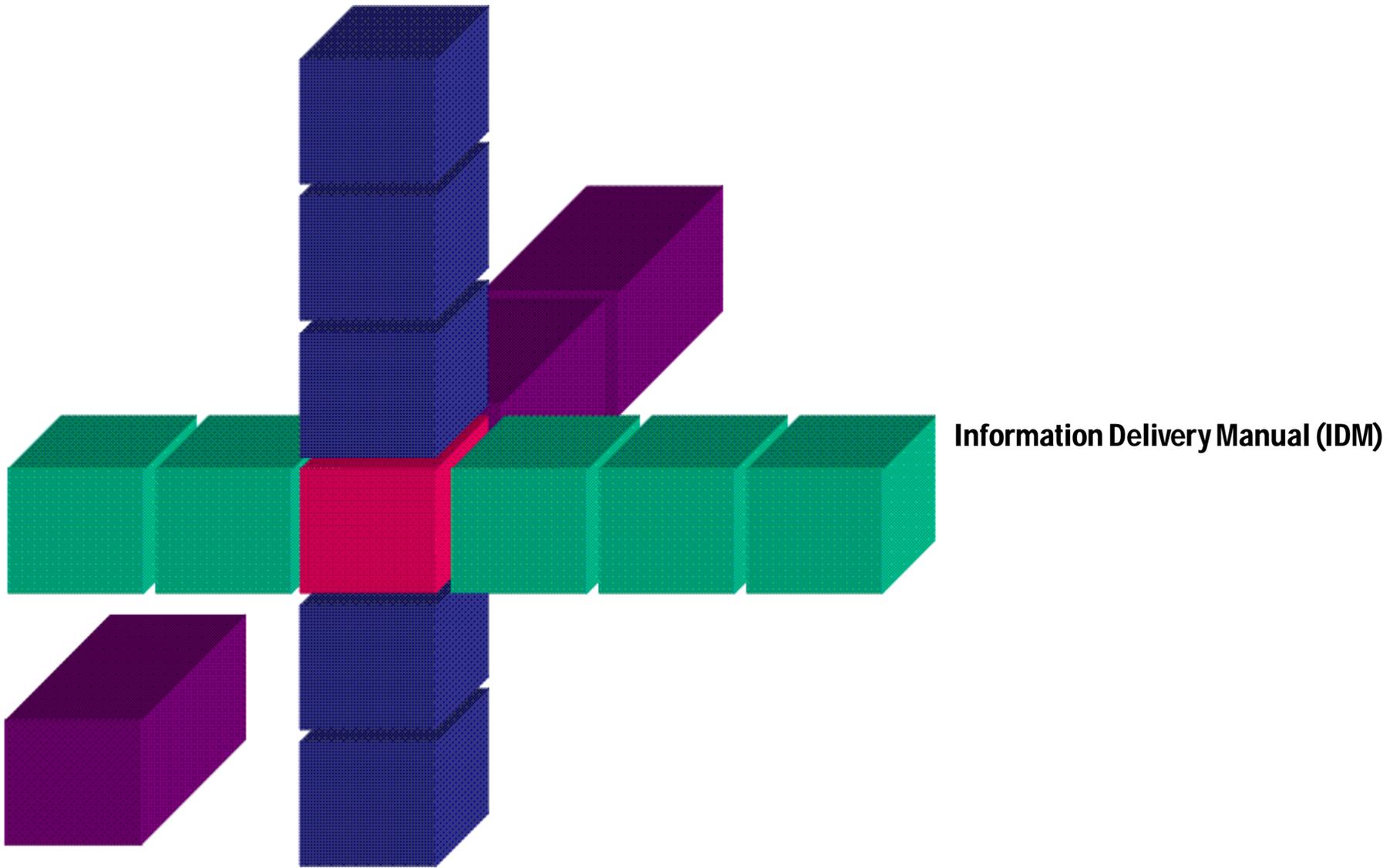
---

Allows tasks to be contained (nested) within other tasks.

For technical detail, refer to **fp\_nests (task)**

	Collect together the tasks that are to be nested (child tasks).
	Define or set the parent task into which the child tasks are to be nested and ensure that it has a globally unique identifier and owner history.
	A relationship object will be created that handles the nesting of the tasks. This relationship will have its own globally unique identifier and owner history.

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



### 3 - Resource - Exchange Maintenance Schedule (ER)

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Mar 15, 2007

Labels: (None)

## Resource

---

Define resources that may be associated with a task in a work schedule

For technical detail, refer to **fp\_assign\_construction\_resource\_to\_task**

Different types of resource are handled through different concepts in this functional part. Each type resource or concept is identified by type below. However, globally unique identification and owner history is set for each resource as:

	Set the globally unique identifier and owner history for the relationship between the construction resource and the task. <i>These should be handled directly by the software application and should not require action by the user.</i>
	Identify the resource.
	Identify the resource group (type) <i>e.g. as plant hire, rental etc.</i>
	Set the extent of the resource consumption <i>e.g. wholly/partially/not occupied.</i>
	Set the base quantity in which the equipment resource is measured.

### Construction Equipment Resource

- Resource may be identified as article number, RFID tag etc.
- Resource group may be identified as plant hire, rental etc.
- Consumption should be wholly/partially/not occupied

### Construction Material Resource

- Resource may be identified as article number, buying code etc.
- Resource group may be identified as sand, gravel etc.
- Consumption should be wholly/partially/not consumed

### Construction Product Resource

- Resource may be identified as article number, buying code etc.
- Resource group may be referenced from the product catalogue etc.
- Consumption should be wholly/partially/not consumed or occupied (depending on the product)

### Labor Resource

- Resource may be identified by payroll number, social security number etc.
- Resource group may be identified by the general actor function such as carpenter, plumber etc. (may be similar to the actor role specification)
- Consumption should be wholly/partially/not occupied

Additional attributes include:

 Specify the skillset used by the labor resource.

### Subcontract Resource

- Resource may be identified by contractor identifier etc.
- Resource group may be identified by the type of subcontract resource (bar bending, insulation etc.)
- Consumption should be wholly/partially/not occupied or consumed
- Base quantity for this purpose is the subcontract unit and is therefore likely to be 1.

Additional attributes include:

 Identify the actor performing the role of the subcontracted resource.

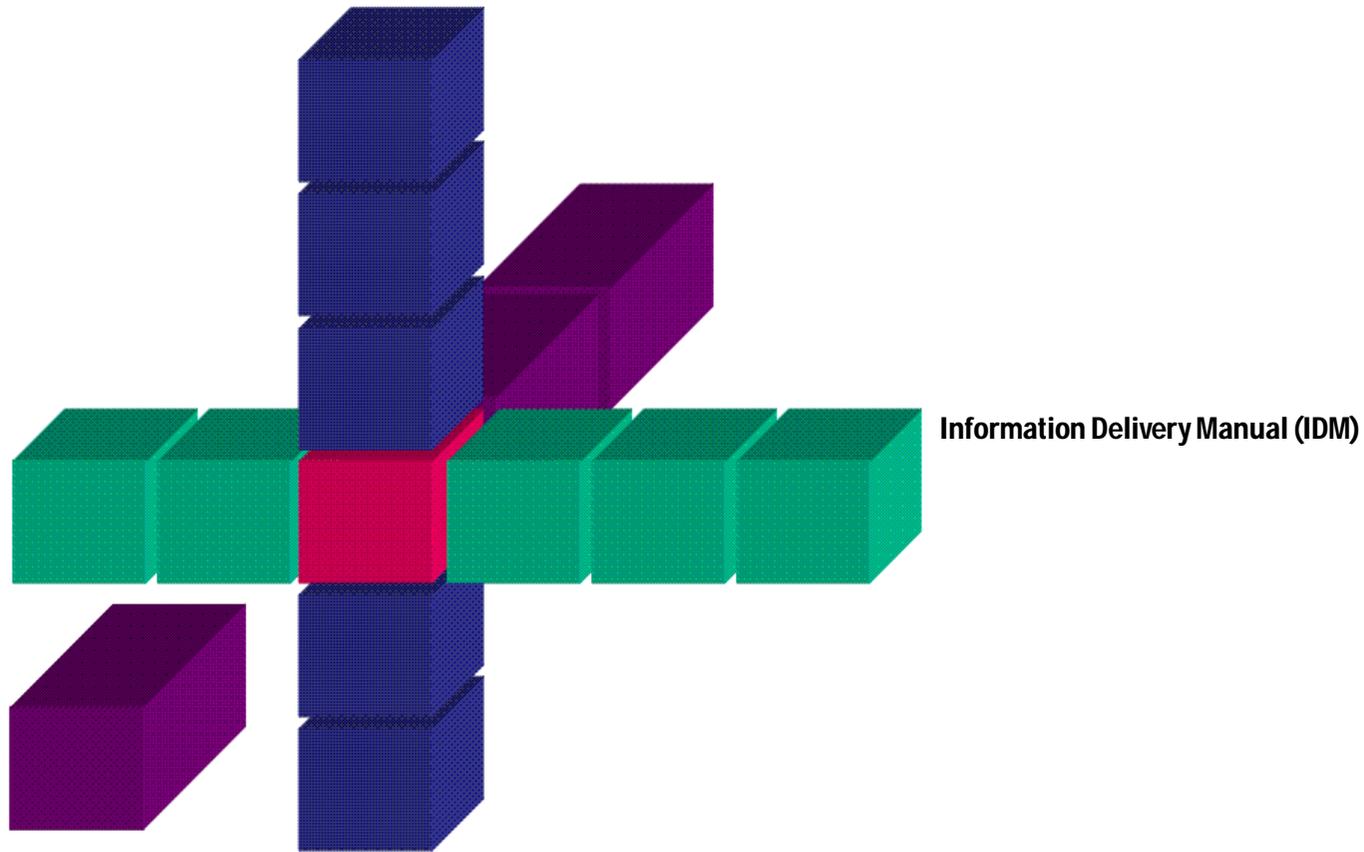
 Describe the jobs that the subcontract should complete.

### Crew Resource

- Resource may be identified by whatever means are appropriate etc.
- Resource group may be identified by whatever means are appropriate
- Consumption should be wholly/partially/not occupied or consumed
- Base quantity for this purpose is the crew unit and is therefore likely to be 1.

Additional attributes include:

 The aggregation of the construction resources that make up the crew (typically labor and equipment).



## Assign Construction Resource To Task (FP)

Added by Jeffrey Wix, last edited by Jeffrey Wix on Jan 14, 2007 ([view change](#))

Labels: (None)

## Assign Resource To Task

### Overview

This functional part describes how construction resources are defined and how they are then assigned to tasks.

Within the IFC schema, a construction resource is considered to be something that is used to assist completion of a process or something that is consumed within a process. There are six separate subtypes of construction resource identified and these are:

- Construction Equipment Resource  
Construction equipment represents resources that may be used to assist in the act of construction and that are wholly or partially occupied (i.e. used) in the performance of construction. Typically, it may be a resource such as a piece of machinery that is used in carrying out a particular process. It might be a piece of earth moving equipment used to form the site prior to construction or a pipe bending machine or special heavy duty drilling or piling equipment. The resource in this case may need to be specifically allocated to a task and, where a schedule time resource indicates start and finish times, the time for which the equipment is allocated may be known.
- Construction Material Resource

A construction material resource represents a general resource that is made available for use on a project without necessarily being specific as to how it is used. However, when it is used for a task, some indication may need to be given as to approximately how much is used.

Care should be taken to distinguish the usage of construction material resources and occurrences of IfcMaterial. A construction material resource can have a material assigned. For instance, stone chippings may be a construction material resource but flint might be the material designation.

Care should also be taken in making a distinction between material and product resources.

- **Construction Product Resource**

Generally, a construction product resource is a product that may have been constructed as part of the project work but whose primary purpose is to support, be used in or be consumed by a task. For instance, formwork used in concrete work may have been the subject of a prior task 'construct formwork. In the 'pour concrete' task, the formwork will then be used as a construction product to assist the shaping of the concrete and may be left in place or removed after completion of the task.

- **Crew Resource**

A crew resource may be used as an aggregation of other resources into a single resource. For instance, it may be applied where an equipment resource is used in conjunction with specific labor and/or sub-contract resources.

- **Labor Resource**

A labor resource may be used to identify a particular skill or craft capability that needs to be applied. In this sense, it may be considered to apply to multiple actors that may assist in the performance of a task. A labor resource is not the same thing as an actor although an actor may be considered as being able to fulfil the requirements of a required labor resource.

- **Sub-Contract Resource**

Represents the requirement for sub-contracted resources (a sub-contractor being an organization with whom a contractual relationship is defined rather than defining the relationship with specific actors.

For each type of resource to be used, the following information may need to be determined:

? A name for the resource used.

? The identity of the resource which may be a code or article number used for the purposes of buying

? Specification of the resource type (carpenter, foreman, electrician etc.). This is equivalent to the use of actor role in relation to an actor.

? How the resource is used or consumed within a task

? The base quantity used for measuring the resource (if appropriate); this may also be identified using element quantities.

Once the resources have been defined, they can be assigned to the task in which they will be used or consumed.

Reference to a specific construction resource concept should be by declaring it as a parameter to the functional part e.g. fp\_assign\_construction\_resource\_to\_tak (IfcLaborResource)

## Results

- Construction resources are identified and assigned to the task in which they will be used.

### Suggested modifications to IFC model and documentation

\* IfcResourceConsumptionEnum has a USERDEFINED value. However, the attribute name is IfcConstructionResource.ResourceConsumption which does not have a specific pointer to an entity/attribute that can hold a user defined value. It is recommended that an optional attribute 'UserDefinedResourceConsumption' is added.

Description	Entity/Pset/Functional Part	MAN	REC	OPT
<b>For all construction resources, general attributes should be asserted</b>				
<i>In the description below, the entity is shown as IfcConstructionResource[subtypel. For normal usage, this should be replaced with the actual entity being used e.g. IfcConstructionEquipmentResource.</i>				
Assert the identity of the construction resource	IfcConstructionResource[subtypel.GlobalId::IfcGloballyUniqueId	✓		
Assert the owner history of the construction resource	IfcConstructionResource[subtypel.OwnerHistory::fp_apply_owner_history	✓		
Specify a name of the construction resource	IfcConstructionResource[subtypel.Name::IfcLabel		✓	

<p><i>This should be a generally useful name in human identification of the type and specific reference for the resource e.g. screwing machine #3.</i></p>				
<p>Specify a description of the construction resource.</p> <p><i>A description should be asserted to give as much information as possible about the object in question.</i></p>	IfcConstructionResource[subtype].Description:: IfcLabel			
<p>Specify the object type for the resource</p>	IfcConstructionResource[subtype].ObjectType:: IfcLabel			
<p><b>Concept of Construction Equipment Resource.</b></p> <p><i>Subtype is IfcConstructionEquipmentResource</i></p>				
<p>Identify the construction resource</p> <p><i>Identification may be in the form of a code or ID used as a general indicator for ordering or buying purposes. This may be an article number, RFID tag etc.</i></p> <p><i>For a construction equipment resource, it is anticipated that the identification will be in the form of a suppliers marking or RFID tag.</i></p>	IfcConstructionEquipmentResource. ResourceIdentifier:: IfcIdentifier			
<p>Identify the resource group</p> <p><i>For construction equipment, this value may be plant hire, rental or some other form of grouping reference.</i></p>	IfcConstructionEquipmentResource.ResourceGroup:: IfcLabel			
<p>Identify the resource consumption.</p> <p><i>This should be selected from the predefined enumeration list which includes</i></p> <ul style="list-style-type: none"> <li>*CONSUMED</li> <li>*PARTIALLYCONSUMED</li> <li>*NOTCONSUMED</li> <li>*OCCUPIED</li> <li>*PARTIALLYOCCUPIED</li> <li>*NOTOCCUPIED</li> <li>*USERDEFINED</li> </ul>	IfcConstructionEquipmentResource.ResourceConsumption:: IfcResourceConsumptionEnum			

<p>*NOTDEFINED_</p> <p><i>For equipment, the values OCCUPIED, PARTIALLYOCCUPIED or NOTOCCUPIED should be used. Consumption values should not be used</i></p>				
<p>Set the base quantity</p> <p>_This is the default, or recommended unit that should be used for measuring the volume (or amount) of the resource and the basic quantity of the resource fully or partially consumed.</p> <p>Generally, for equipment, the base quantity may be omitted as equipment will be measured in 'counted' units._</p>	IfcConstructionEquipmentResource.BaseQuantity::fp_measure_with_unit			✓
<p><b>Concept of Construction Material Resource.</b></p> <p><i>Subtype is IfcConstructionMaterialResource</i></p>				
<p>Identify the construction resource</p> <p><i>Identification is expected to be in the form of an article number or buying code for a material resource.</i></p>	IfcConstructionMaterialResource.ResourceIdentifier::IfcIdentifier		✓	
<p>Identify the resource group</p> <p><i>For construction material, this value may be sand, gravel etc.</i></p>	IfcConstructionMaterialResource.ResourceGroup::IfcLabel		✓	
<p>Identify the resource consumption.</p> <p>_ *See predefined list above</p> <p><i>For material, the values CONSUMED, PARTIALLY CONSUMED or NOTCONSUMED should be used. Occupied values should not be used</i></p>	IfcConstructionMaterialResource.ResourceConsumption::IfcResourceConsumptionEnum		✓	
<p>Set the base quantity</p> <p><i>This is the default, or recommended unit that should be used for measuring the volume (or amount) of the resource and the basic quantity of the resource fully or partially consumed.</i></p>	IfcConstructionMaterialResource.BaseQuantity::fp_measure_with_unit		✓	
<p><b>Concept of Construction Product Resource.</b></p> <p><i>Subtype is IfcConstructionProductResource</i></p>				
<p>Identify the construction resource</p>	IfcConstructionProductResource.ResourceIdentifier::IfcIdentifier		✓	

<p>  <i>Identification is expected to be in the form of an article number or buying code for a material resource.</i></p>				
<p>  Identify the resource group</p> <p>  <i>For construction product, this value may be related to the product catalogue</i></p>	IfcConstructionProductResource.ResourceGroup: : IfcLabel		✓	
<p>  Identify the resource consumption.</p> <p>  <i>_See predefined list above</i></p> <p>  <i>For products, any of the values in the predefined list may be appropriate_</i></p>	IfcConstructionProductResource.ResourceConsumption: : IfcResourceConsumptionEnum		✓	
<p>  Set the base quantity</p> <p>  <i>This is the default, or recommended unit that should be used for measuring the volume (or amount) of the resource and the basic quantity of the resource fully or partially consumed.</i></p>	IfcConstructionProductResource.BaseQuantity: : fp_measure_with_unit		✓	
<p><b>Concept of Labor Resource.</b></p> <p><i>Subtype is IfcLaborResource</i></p>				
<p>Identify the construction resource</p> <p><i>Identification may be in the form of a reference number used for human resource, payroll or social security purposes.</i></p>	IfcLaborResource.ResourceIdentifier: : IfcIdentifier		✓	
<p>Identify the resource group</p> <p><i>For a labor resource, this value may be equivalent to a personal role (but applied in general terms) such as carpenter, foreman, electrician etc</i></p>	IfcLaborResource.ResourceGroup: : IfcLabel		✓	
<p>Identify the resource consumption.</p> <p><i>_ *See predefined list above</i></p> <p>For labor, the values OCCUPIED, PARTIALLYOCCUPIED or NOTOCCUPIED should be used. Consumption values should not be used_</p>	IfcLaborResource.ResourceConsumption: : IfcResourceConsumptionEnum		✓	

<p>Set the base quantity</p> <p>_This is the default, or recommended unit that should be used for measuring the volume (or amount) of the resource and the basic quantity of the resource fully or partially consumed.</p> <p>Generally, for labor, the base quantity may be omitted._</p>	IfcLaborResource.BaseQuantity::fp_measure_with_unit			
<p>Specify the skillset</p> <p>_This provides an indication of the skill por craft possessed by the labor resource and which may be used to determine the potential to perform a particular task which is otherwise constraint in terms of the skills required. _</p>	IfcLaborResource.Skillset:			
<p><b>Concept of Subcontract Resource.</b></p> <p><i>Subtype is IfcSubcontractResource</i></p>				
<p>  Identify the construction resource</p>	IfcSubcontractResource.ResourceIdentifier::IfcIdentifier			
<p>  Identify the resource group</p>	IfcSubcontractResource.ResourceGroup::IfcLabel			
<p>  Identify the resource consumption.</p> <p>  *_See predefined list above</p> <p>For sub-contracts, any of the values in the predefined list may be appropriate._</p>	IfcSubcontractResource.ResourceConsumption::IfcResourceConsumptionEnum			
<p>  Set the base quantity</p> <p>  _This is the default, or recommended unit that should be used for measuring the volume (or amount) of the resource and the basic quantity of the resource fully or partially consumed.</p> <p>  Generally, for sub-contracts, the base quantity may be omitted._</p>	IfcSubcontractResource.BaseQuantity::fp_measure_with_unit			
<p>  Set the identity of the actor performing the role of the subcontracted resource</p>	IfcSubcontractResource.SubContractor::fp_select_actor			

Describe the jobs that this subcontract should complete	IfcSubcontractResource.JobDescription::IfcText			
<b>Concept of Crew Resource.</b> <i>Subtype is IfcCrewResource</i>				
Identify the construction resource	IfcCrewResource.ResourceIdentifier::IfcIdentifier			
Identify the resource group	IfcCrewResource.ResourceGroup::IfcLabel			
Identify the resource consumption.  _See predefined list above  For crews, any of the values in the predefined list may be appropriate_	IfcCrewResource.ResourceConsumption::IfcResourceConsumptionEnum			
Set the base quantity  _This is the default, or recommended unit that should be used for measuring the volume (or amount) of the resource and the basic quantity of the resource fully or partially consumed.  Generally, for crews, the base quantity may be omitted._	IfcCrewResource.BaseQuantity::fp_measure_with_unit			
Determine the resources that are aggregated within the crew resource.  <i>Any type of resource may be aggregated within a crew resource. Generally however, it is anticipated that crews will comprise equipment and labor resources.</i>	fp_aggregates			
<b>Specify the task to which the construction resource(s) are to be assigned</b>  <i>Note that details of the attribute settings for tasks is given in other functional parts notably fp_control_maintenance_plan and fp_control_maintenance_schedule. Specific information on setting of attributes is not given in this functional part</i>	IfcRelAssignsToProcess.RelatingProcess::IfcTask			
Specify the construction resources that are to be assigned to the task.	IfcRelAssignsToProcess.RelatedObjects::IfcConstructionResource			

<p>Note that specific subtypes of construction resource as indicated below should be used as the target datatype for the related objects attribute.</p>				
<p>Set the quantity of the object specific for the operation by this process.</p> <p>This attribute is not relevant to the assigning of resources to tasks.</p>	IfcRelAssignsToProcess.QuantityInProcessIfcMeasureWithUnit	X	X	X
<p>Assert the identity of the relationship</p>	IfcRelAssignsToProcess.GlobalId::IfcGloballyUniqueId			
<p>Assert the owner history of the relationship</p>	IfcRelAssignsToProcess.OwnerHistory::fp_apply_owner_history			
<p>Specify a name of the relationship</p>	IfcRelAssignsToProcess.Name::IfcLabel			
<p>Specify a description of the relationship</p>	IfcRelAssignsToProcess.Description::IfcLabel			

## IFC Entities Required

- ? IfcConstructionResource
- ? IfcConstructionEquipmentResource
- ? IfcConstructionMaterialResource
- ? IfcConstructionProductResource
- ? IfcCrewResource
- ? IfcLaborResource
- ? IfcObject
- ? IfcObjectDefinition
- ? IfcProcess
- ? IfcRelationship
- ? IfcRelAssigns
- ? IfcRelAssignsToProcess

? IfcResource

? IfcRoot

? IfcSubcontractResource

? IfcTask

## IFC Datatypes Required

---

? IfcGloballyUniqueId

? IfcIdentifier

? IfcLabel

? IfcObjectTypeEnum

? IfcRatioMeasure

? IfcResourceConsumptionEnum

? IfcText

## IFC Functions Required

---

? -

## IFC Property Sets Required

---

? -

## IDM Functional Parts Required

---

? fp\_aggregates

? fp\_apply\_owner\_history

? fp\_define\_by\_properties

? fp\_measure\_with\_unit

? fp\_select\_actor

## EXPRESS-G

---

## EXPRESS Schema

---

```
SCHEMA FP_ASSIGN_CONSTRUCTION_RESOURCE_TO_TASK;  
  
  TYPE IfcGloballyUniqueId = STRING (22) FIXED;  
  END_TYPE;  
  
  TYPE IfcIdentifier = STRING;  
  END_TYPE;  
  
  TYPE IfcLabel = STRING;  
  END_TYPE;  
  
  TYPE IfcRatioMeasure = REAL;
```

```
END_TYPE;

TYPE IfcText = STRING;
END_TYPE;

TYPE IfcObjectTypeEnum = ENUMERATION OF
(PRODUCT,
PROCESS,
CONTROL,
RESOURCE,
ACTOR,
GROUP,
PROJECT,
NOTDEFINED);
END_TYPE;

TYPE IfcResourceConsumptionEnum = ENUMERATION OF
(CONSUMED,
PARTIALLYCONSUMED,
NOTCONSUMED,
OCCUPIED,
PARTIALLYOCCUPIED,
NOTOCCUPIED,
USERDEFINED,
NOTDEFINED);
END_TYPE;

ENTITY IfcObject
ABSTRACT SUPERTYPE OF (ONEOF(IfcProcess, IfcResource))
SUBTYPE OF(IfcObjectDefinition);
ObjectType : OPTIONAL IfcLabel;
WHERE
WR1 : SIZEOF(QUERY(temp <* IsDefinedBy | 'IFC2X2_FINAL.IFCRELDEFINESBYTYPE' IN TYPEOF(temp))) <= 1;
END_ENTITY;

ENTITY IfcRoot
ABSTRACT SUPERTYPE OF (ONEOF(IfcRelationship, IfcObjectDefinition));
GlobalId      : IfcGloballyUniqueId;
Name          : OPTIONAL IfcLabel;
Description   : OPTIONAL IfcText;
OwnerHistory  : fp_apply_owner_history;
UNIQUE
URL : GlobalId;
END_ENTITY;

ENTITY IfcRelationship
ABSTRACT SUPERTYPE
SUBTYPE OF(IfcRoot);
END_ENTITY;

ENTITY IfcRelAssigns
ABSTRACT SUPERTYPE
SUBTYPE OF(IfcRelationship);
RelatedObjects      : SET [1:?] OF IfcObject;
RelatedObjectsType : OPTIONAL IfcObjectTypeEnum;
WHERE
WR1 : IfcCorrectObjectAssignment(RelatedObjectsType, RelatedObjects);
END_ENTITY;

ENTITY IfcProcess
ABSTRACT SUPERTYPE
SUBTYPE OF(IfcObject);
END_ENTITY;

ENTITY IfcTask
SUBTYPE OF(IfcProcess);
TaskId      : IfcIdentifier;
Status     : OPTIONAL IfcLabel;
WorkMethod : OPTIONAL IfcLabel;
```

```
IsMilestone : BOOLEAN;
Priority : OPTIONAL INTEGER;
WHERE
WR1 : SIZEOF(QUERY(temp <* SELF\IfcObject.Decomposes | NOT('IFC2X2_FINAL.IFCRELNESTS' IN TYPEOF(temp)))) = 0;
WR2 : SIZEOF(QUERY(temp <* SELF\IfcObject.IsDecomposedBy | NOT('IFC2X2_FINAL.IFCRELNESTS' IN TYPEOF(temp)))) = 0;
WR3 : EXISTS(SELF\IfcRoot.Name);
END_ENTITY;

ENTITY IfcResource
ABSTRACT SUPERTYPE
SUBTYPE OF(IfcObject);
END_ENTITY;

ENTITY IfcConstructionResource
ABSTRACT SUPERTYPE OF (ONEOF(IfcConstructionEquipmentResource, IfcConstructionMaterialResource, IfcConstructionProductResource, IfcCrewResource, IfcLaborResource, IfcSubContractResource))
SUBTYPE OF(IfcResource);
ResourceIdentifier : OPTIONAL IfcIdentifier;
ResourceGroup : OPTIONAL IfcLabel;
ResourceConsumption : OPTIONAL IfcResourceConsumptionEnum;
BaseQuantity : fp_measure_with_unit;
END_ENTITY;

ENTITY IfcConstructionEquipmentResource
SUBTYPE OF(IfcConstructionResource);
END_ENTITY;

ENTITY IfcConstructionMaterialResource
SUBTYPE OF(IfcConstructionResource);
UsageRatio : OPTIONAL IfcRatioMeasure;
Suppliers : OPTIONAL SET [1:?] OF fp_select_actor;
WHERE
WR1 : SIZEOF(SELF\IfcResource.ResourceOf) <=1;
WR2 : NOT(EXISTS(SELF\IfcResource.ResourceOf[1])) OR
(SELF\IfcResource.ResourceOf[1].RelatedObjectsType = IfcObjectTypeEnum.PRODUCT);
END_ENTITY;

ENTITY IfcConstructionProductResource
SUBTYPE OF(IfcConstructionResource);
WHERE
WR1 : SIZEOF(SELF\IfcResource.ResourceOf) <=1;
WR2 : NOT(EXISTS(SELF\IfcResource.ResourceOf[1])) OR
(SELF\IfcResource.ResourceOf[1].RelatedObjectsType = IfcObjectTypeEnum.PRODUCT);
END_ENTITY;

ENTITY IfcCrewResource
SUBTYPE OF(IfcConstructionResource);
END_ENTITY;

ENTITY IfcLaborResource
SUBTYPE OF(IfcConstructionResource);
SkillSet : OPTIONAL IfcText;
END_ENTITY;

ENTITY IfcSubContractResource
SUBTYPE OF(IfcConstructionResource);
JobDescription : OPTIONAL IfcText;
SubContractor : OPTIONAL fp_select_actor;
END_ENTITY;

ENTITY IfcRelAssignsToProcess
SUBTYPE OF(IfcRelAssigns);
RelatingProcess : IfcProcess;
QuantityInProcess : fp_measure_with_unit;
WHERE
WR1 : SIZEOF(QUERY(Temp <* SELF\IfcRelAssigns.RelatedObjects | RelatingProcess := Temp)) = 0;
END_ENTITY;

ENTITY IfcObjectDefinition
SUBTYPE OF(IfcRoot);
```

```
INVERSE
  HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
END_ENTITY;

ENTITY fp_aggregates;
END_ENTITY;

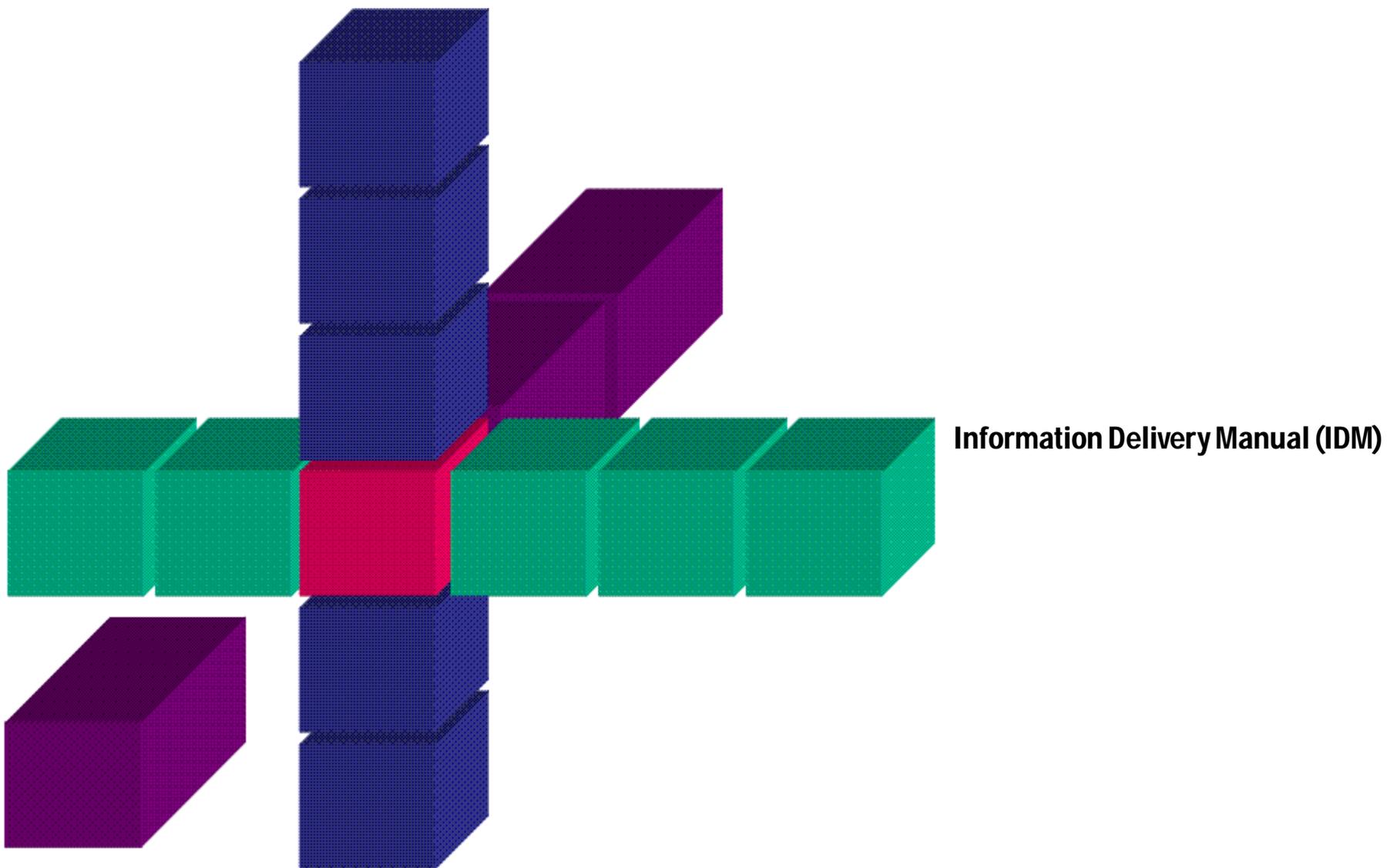
ENTITY fp_apply_owner_history;
END_ENTITY;

ENTITY fp_define_by_properties;
END_ENTITY;

ENTITY fp_measure_with_unit;
END_ENTITY;

ENTITY fp_select_actor;
END_ENTITY;

END_SCHEMA;
```



## Assigns To Group (FP)

Added by [Brendan McFarlane](#), last edited by [Brendan McFarlane](#) on Sep 27, 2006 ([view change](#))

Labels: (None)

# Assigns To Group (FP)

---

## Overview

Describes a grouping relationship in which the parent object (the group) is made from the assignment of a set of other objects.

Grouping allows for any arbitrary collection of objects to be brought together into a designated, named group. Objects can still act independently of the group or can be operated upon within the context of the whole group.

An object can belong to more than one group concurrently. For instance, it may belong to one group that is a system, to a second group for cost scheduling purposes, and to a third group for scheduling purposes.

Certain specific types of group are defined within the IFC model. These include:

- Asset
- Condition
- Inventory
- Structural load group
- Structural result group
- System
  - Electrical Circuit (type of system)
- Zone

In a group assignment relationship, the group is referred to as the 'relating group' whilst the assigned objects are referred to as the 'related objects'. There is always exactly one relating group and there may be one or many related objects. An object may be assigned only once to each group in which it participates.

The group assignment relationship is described at a high level in the IFC model as between an occurrence of `IfcGroup` and occurrences of `IfcObject`. Practically, the relationship will be between instances of subtypes of `IfcObject` and these inherit the relationship.

With exchange requirements that call upon the use of this functional part, reference to the type of group to which objects are assigned may be defined using a parameter as:

```
fp_assigns_to_group ()      -> describes an arbitrary collection of objects into a named group
fp_assigns_to_group (system) -> describes the assignment of objects to a system group
fp_assigns_to_group (asset) -> describes the assignment of objects to an identified asset group
etc.
```

Use of the parameter recognizes that certain rules may apply to the collection of objects into particular group types.

## Results

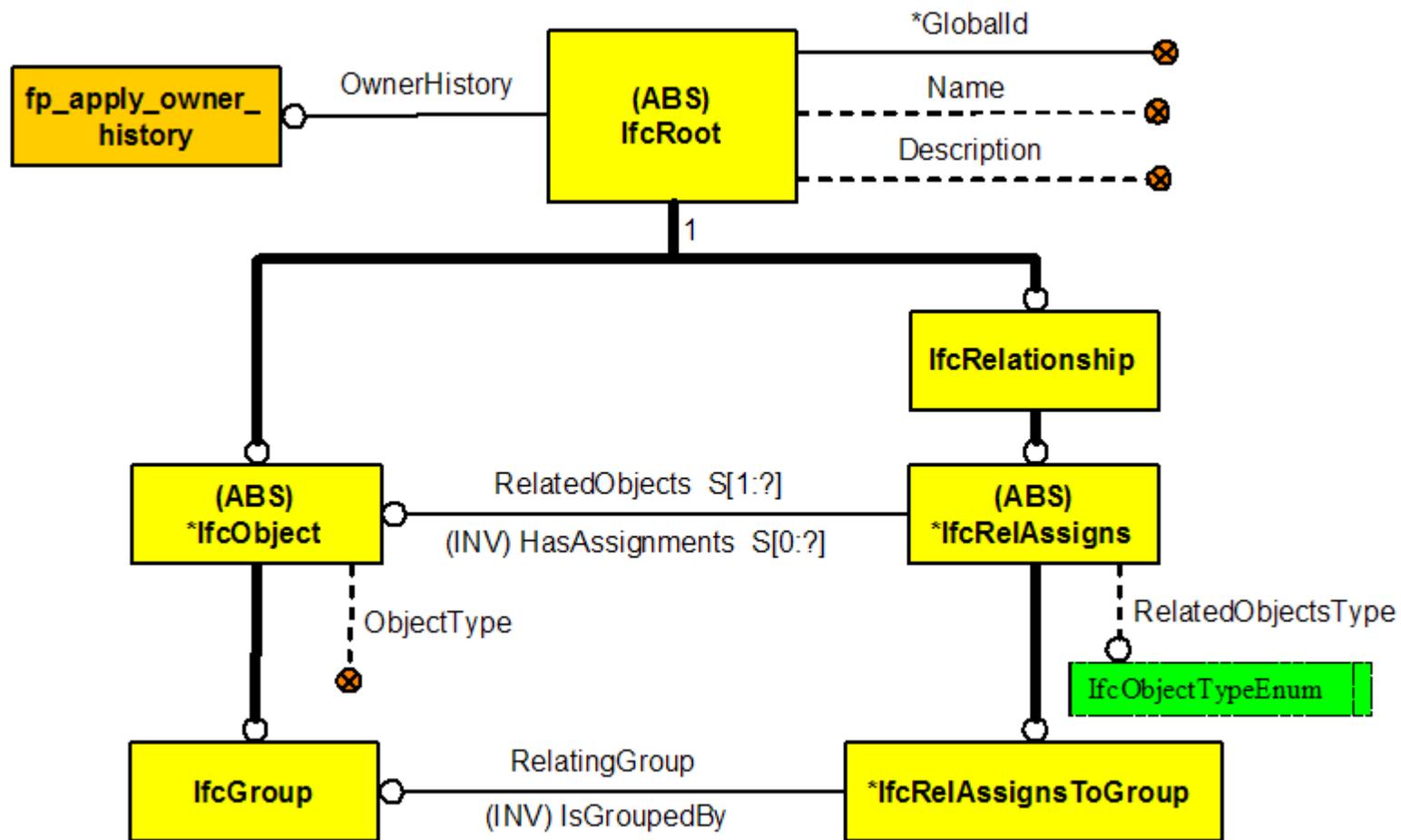
Declares the assignment of a collection of objects into a group

Description	Entity/Pset/Functional Part	MAN	REC	OPT
<b>Create the group to which entities will be related and the entities.</b> <i>Note that groups are modeled explicitly elsewhere.</i>				
Set the occurrence of the group in the relationship	IfcGroup	✓		
Set the occurrences of the child entities in the relationship	IfcObject <child subtypes>	✓		
<b>Assert the attributes of the relationship</b>				
Assert the parent relationship	IfcRelAssignsToGroup.RelatingGroup -> IfcGroup	✓		
Assert the child relationship	IfcRelAssignsToGroup.RelatedObjects -> IfcObject	✓		
Assert the type of related objects from the predefined list if required <i>This value can be used to identify the parent supertype of related objects as e.g. product, group, process, control etc. If the related objects have mixed supertypes, then the value .NOTDEFINED. should be used.</i>	IfcRelAssignsToGroup.RelatedObjectType -> IfcObjectTypeEnum			✓
Assert the identity of the relationship	IfcRelAssignsToGroup.GlobalId -> IfcGloballyUniqueId	✓		
Assert the owner history of the relationship	IfcRelAssignsToGroup.OwnerHistory -> fp_apply_owner_history	✓		
Specify a name of the relationship	IfcRelAssignsToGroup.Name		✓	
Specify a description of the relationship	IfcRelAssignsToGroup.Description			✓

- IFC Entities Required
  - IfcGroup
  - IfcObject
  - IfcRelationship
  - IfcRelAssigns
  - IfcRelAssignsToGroup
  - IfcRoot
- IFC Datatypes Required
  - IfcGloballyUniqueId
  - IfcLabel
  - IfcObjectTypeEnum
  - IfcText

- IFC Functions Required
  - -
- IFC Property Sets Required
  - -
- IDM Functional Parts Required
  - fp\_apply\_owner\_history

**EXPRESS-G**



## EXPRESS Schema

```
SCHEMA FP_ASSIGNNS_TO_GROUP;

TYPE IfcGloballyUniqueId = STRING (22) FIXED;
END_TYPE;

TYPE IfcLabel = STRING;
END_TYPE;

TYPE IfcText = STRING;
END_TYPE;

TYPE IfcObjectTypeEnum = ENUMERATION OF
  (PRODUCT,
   PROCESS,
   CONTROL,
   RESOURCE,
   ACTOR,
   GROUP,
   PROJECT,
   NOTDEFINED);
END_TYPE;

ENTITY IfcObject
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcRoot);
  ObjectType      : OPTIONAL IfcLabel;
  INVERSE
  HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
  WHERE
  WR1 : SIZEOF(QUERY(temp <* IsDefinedBy | 'IFC2X2_FINAL.IFCRELDEFINESBYTYPE' IN TYPEOF(temp))) <= 1;
END_ENTITY;

ENTITY IfcRoot
  ABSTRACT SUPERTYPE OF (ONEOF(IfcObject, IfcRelationship));
  GlobalId      : IfcGloballyUniqueId;
  Name          : OPTIONAL IfcLabel;
  Description   : OPTIONAL IfcText;
  OwnerHistory : fp_apply_owner_history;
  UNIQUE
  UR1 : GlobalId;
END_ENTITY;

ENTITY IfcRelAssigns
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcRelationship);
  RelatedObjects      : SET [1:?] OF IfcObject;
  RelatedObjectsType : OPTIONAL IfcObjectTypeEnum;
  WHERE
  WR1 : IfcCorrectObjectAssignment(RelatedObjectsType, RelatedObjects);
```

```
END_ENTITY;

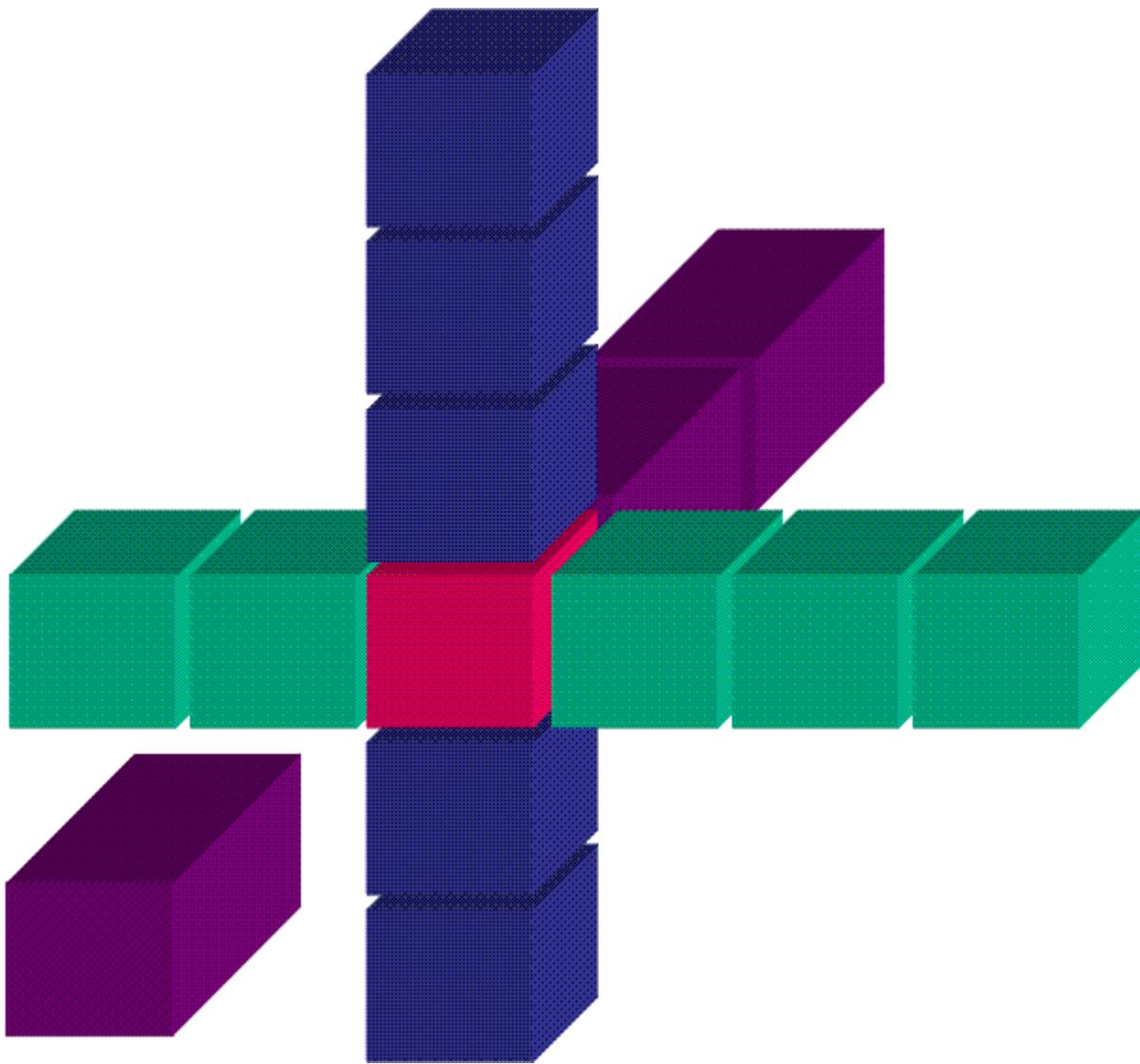
ENTITY IfcGroup
  SUBTYPE OF(IfcObject);
  INVERSE
    IsGroupedBy : IfcRelAssignsToGroup FOR RelatingGroup;
END_ENTITY;

ENTITY IfcRelAssignsToGroup
  SUBTYPE OF(IfcRelAssigns);
  RelatingGroup : IfcGroup;
  WHERE
    WR1 : SIZEOF(QUERY(Temp <* SELF\IfcRelAssigns.RelatedObjects | RelatingGroup == Temp)) = 0;
END_ENTITY;

ENTITY IfcRelationship
  SUBTYPE OF(IfcRoot);
END_ENTITY;

ENTITY fp_apply_owner_history;
END_ENTITY;

END_SCHEMA;
```



**Information Delivery Manual (IDM)**

## **Control Maintenance Plan (FP)**

Added by [Jeffrey Wix](#), last edited by [Jeffrey Wix](#) on Jan 20, 2007 ([view change](#))

Labels: (None)

# Control Maintenance Plan

---

## Overview

---

A maintenance plan is a type of work plan in which the work tasks that are planned are specifically related the assets that are to be maintained. That is, the maintenance plan will provide a listing of all the maintenance tasks that must be undertaken and the period that can elapse between successive performances of the same task. A maintenance plan does not identify the actual dates on which work should be undertaken. This is the function of a maintenance schedule which is described in fp\_control\_maintenance\_schedule.

Additionally, a maintenance plan may define required resources to perform tasks in terms of the skills or capabilities required. It is not expected that a maintenance plan will identify actors by name.

For instance, the shaft bearings of a pump impeller may require lubrication once each year with the work being undertaken by a Class 1 Lubrication Specialist. This is a work description that can form the basis of a maintenance plan.

For a maintenance plan, the following information will be needed:

The maintenance tasks that are to be undertaken with, for each task:

Identity of the task (using the general form of identification or breakdown structure adopted for the project)

The name of the tasks (i.e. how it will be shown in a chart)

Any further useful description (which may be provided if required).

Indication that this is not a milestone task (note that a task is required to be identified as to whether or not it is a milestone; this is not considered to be appropriate for a maintenance plan and the value should therefore be set to FALSE by default.

The assets on which the work is done

For each task on a particular asset, the period of time between occasions when the work is carried out.

### Suggested modifications to IFC model and documentation

? IfcTimeMeasure is specified as a real value that is normally measured in seconds. Many time measures may be measured in much larger units (months/years for a typical building or building component). This fact has to be either dealt with in software (currently as indicated) or an alternative approach to measuring time measure determined for IFC.

? To deal with periods between tasks in a work plan, a property set (Pset\_WorkPlanPeriod) is introduced. This should be considered to act as a placeholder for a more serious reconsideration of the time series capability. The property set has two properties. The first is a time measure property which measures the basic unit (day, week, month, year). The second value is an integer value that specifies how many units of time measure are to pass between planned tasks being done. Thus, for work planned to be done on a quarterly basis, the PlanUnitTimeMeasure would be 'MONTH' and the PlanTaskPeriod would be 3

## Results

---

Maintenance plan is established in which all maintenance tasks are defined for assets that are to be maintained and the period at which they should be undertaken is

indicated.

Information Needed	Entity/Pset/Functional Part	MAN	REC	OPT
<p><b>CONCEPT: Establish a maintenance plan defining the maintenance work to be undertaken on a selection of assets.</b></p> <p>Note that whilst it would be possible to look at the maintenance of products, it is assumed that such products are grouped together within an asset for the purposes of maintenance. Individual elements of the asset are identifiable within the asset group through the general grouping mechanism.</p>				
<p><b>Determine the maintenance tasks to be undertaken. Perform the following actions for each task to be included in the maintenance plan.</b></p>				
<p><b>Assert general attributes for the task</b></p>				
<p>Set the global unique identifier</p>	IfcTask.GlobalId::fcGloballyUniqueId			
<p>Assert the owner history of the task</p>	IfcTask.OwnerHistory::fp_apply_owner_history			
<p><i>Specify a name for the task</i></p> <p><i>Note that this is the primary presentation of the task; it is the text entry in a plan or schedule.</i></p> <p><i>This attribute must be asserted for a task</i></p>	IfcTask.Name::IfcLabel			
<p>Specify a description for the task</p> <p><i>Specification of a description is optional but can provide</i></p>	IfcTask.Description::IfcLabel			

<i>valuable further information</i>				
<b>Special information about the task</b>				
<p>Assert special information that may relate to a task</p> <p><i>Examples that may be used include fixed duration, fixed unit, fixed work etc.</i></p>	IfcTask.ObjectType:: IfcLabel			
<b>Assert specific task related attributes that are relevant to the maintenance plan</b>				
<i>Note that certain attributes of a task are relevant only to use of a task within a schedule and therefore should not be used for a maintenance plan</i>				
<p>Set the value of the Task identifier.</p> <p><i>Note that this is a reference value given to the task in the context of the maintenance plan. It must be locally unique (within the maintenance plan), but is not a globally unique identifier. It may be used as a sequence number in a list of tasks 1,2,3,4 etc. or as work breakdown structure identifier as a lightweight alternative to the use of the classification capability in IFC.</i></p>	IfcTask.TaskId :: IfcIdentifier			
<p>Identify if the task is a milestone.</p> <p><i>Although this value must be asserted, it is not relevant to the maintenance plan. To ensure that it is taken completely out of consideration in this context, it is a RULE that the value must be asserted as FALSE for a maintenance plan.</i></p>	IfcTask.IsMilestone:: BOOLEAN = FALSE			
Set the current status of the task	IfcTask.Status :: IfcLabel	X	X	X

<p><i>This is not relevant for a maintenance plan and therefore MUST not be asserted</i></p>				
<p>Specify the method of work to be used in carrying out the task</p> <p><i>This is not relevant for a maintenance plan and therefore MUST not be asserted</i></p>	IfcTask.WorkMethod :: IfcLabel	X	X	X
<p>Set the relative priority of the task (in comparison to the priorities of other tasks).</p> <p><i>This is not relevant for a maintenance plan and therefore MUST not be asserted</i></p>	IfcTask.Priority :: INTEGER	X	X	X
<p><b>A task may nest other tasks, only the parent task being related to the asset. For each sub-task, the procedure for assertion of attributes as defined above must be followed. Once the tasks are defined, they can be brought together into a nesting relationship.</b></p> <p><i>Note that whilst the possibility of nested tasks is allowed for, it is not expected that this capability will be widely used in developing a maintenance plan</i></p>	fp_nests (IfcTask)			
<p><b>Define the period between planned work actions</b></p>				
<p>Set the unit time measure to be applied to the task</p> <p><i>Note that this value must be asserted for the specification of a maintenance plan.</i></p>	Pset_WorkPlanPeriod.PlanUnitTimeMeasure :: IfcPropertyEnumeratedValue :: IfcLabel			

Set the number of units of the time measure that specifies the period between recurring occurrences of the task  <i>Note also that this value must be asserted to qualify the meaning of the time measure above.</i>	Pset_WorkPlanPeriod.PlanTaskPeriod::IfcPropertySingleValue::IfcInteger			
<b>Define the property set for the task</b>				
Define each occurrence of task to which the work plan period is to apply	fp_define_by_properties			
<b>Define resources that may be associated with a task in a work plan</b>				
Resources are allocated through the functional part	fp_assign_construction_resource_to_task			
For a construction equipment resource	fp_assign_construction_resource_to_task (IfcConstructionEquipmentResource)			
For a construction material resource	fp_assign_construction_resource_to_task (IfcConstructionMaterialResource)			
For a construction product resource	fp_assign_construction_resource_to_task (IfcConstructionProductResource)			
For a crew resource	fp_assign_construction_resource_to_task (IfcCrewResource)			
For a labor resource	fp_assign_construction_resource_to_task (IfcLaborResource)			
For a sub-contract resource	fp_assign_construction_resource_to_task (IfcSubContractResource)			
<b>Identify the asset</b>				

<p>Identify the asset to which the task will be assigned</p> <p><i>Identifying the asset through the functional part also allows for all of the elements that are grouped within the asset to be determined.</i></p>	fp_model_asset			
<p><b>Associate the asset with the task</b></p> <p>_In this functional part, the asset is assigned to the task rather than vice versa. This is done because the asset is a group rather than a product.</p> <p>Note also that the identification of the asset is not specifically referenced by the work control. Reference to the asset is considered to be implicit through the task_</p>				
<p>Specify the relating task</p>	IfcRelAssignsToProcess.RelatingProcess::IfcTask			
<p>Specify the related objects</p> <p>_Note here that specific reference is made to IfcAsset although access to the IfcAsset specification is through the fp_model_asset functional part.</p> <p>A given task may be relevant for multiple assets. This can be managed through the fact that the IfcRelAssignsToProcess allows for multiple related objects._</p>	IfcRelAssignsToProcess.RelatedObjects::IfcAsset			
<p>Set the quantity of the object specific for the operation by this process.</p>	IfcRelAssignsToProcess.QuantityInProcess::IfcMeasureWithUnit	X	X	X

<i>This attribute is not relevant to the assigning of tasks to the work plan and should not be asserted for this purpose.</i>				
Assert the identity of the relationship	IfcRelAssignsToProcess.GlobalId::IfcGloballyUniqueId	✓		
Assert the owner history of the relationship	IfcRelAssignsToProcess.OwnerHistory::fp_apply_owner_history	✓		
Specify a name of the relationship	IfcRelAssignsToProcess.Name::IfcLabel		✓	
Specify a description of the relationship	IfcRelAssignsToProcess.Description::IfcLabel			✓
<b>Define the work plan to which the tasks will be assigned</b>				
Assert the identity of the work plan	IfcWorkPlan.GlobalId::IfcGloballyUniqueId	✓		
Assert the owner history of the work plan	IfcWorkPlan.OwnerHistory::fp_apply_owner_history	✓		
Specify a name of the work plan	IfcWorkPlan.Name::IfcLabel		✓	
Specify a description of the work plan	IfcWorkPlan.Description::IfcLabel			✓
Set identifier of the work plan	IfcWorkPlan.Identifier::IfcIdentifier	✓		
Set the date that the plan is created	IfcWorkPlan.CreationDate::fp_select_date_time	✓		

## IFC Entities Required

---

- IfcControl
- IfcGroup
- IfcObject
- IfcObjectDefinition
- IfcProcess
- IfcProduct
- IfcRelationship
- IfcRelAssigns
- IfcRelAssignsTasks
- IfcRelAssignsToControl
- IfcRelAssignsToProcess
- IfcRoot
- IfcScheduleTimeControl
- IfcTask
- IfcWorkControl
- IfcWorkPlan

## IFC Datatypes Required

---

- IfcGloballyUniqueId
- IfcIdentifier
- IfcLabel
- IfcObjectTypeEnum
- IfcPositiveRatioMeasure
- IfcRatioMeasure
- IfcText
- IfcTimeMeasure
- IfcWorkControlTypeEnum

## IFC Functions Required

---

- -

## IFC Property Sets Required

---

- Pset\_WorkPlanPeriod

## IDM Functional Parts Required

---

- fp\_apply\_owner\_history;
- fp\_assign\_construction\_resource\_to\_task
- fp\_define\_by\_properties;
- fp\_measure\_with\_unit;
- fp\_model\_asset
- fp\_nests
- fp\_place\_object;
- fp\_represent\_product;
- fp\_select\_date\_time;

## EXPRESS Schema

---

```
SCHEMA FP_CONTROL_MAINTENANCE_PLAN;  
  
TYPE IfcGloballyUniqueId = STRING (22) FIXED;  
END_TYPE;  
  
TYPE IfcIdentifier = STRING;  
END_TYPE;  
  
TYPE IfcLabel = STRING;  
END_TYPE;  
  
TYPE IfcRatioMeasure = REAL;  
END_TYPE;  
  
TYPE IfcPositiveRatioMeasure = IfcRatioMeasure;  
WHERE  
WR1 : SELF > 0;  
END_TYPE;  
  
TYPE IfcText = STRING;  
END_TYPE;  
  
TYPE IfcTimeMeasure = REAL;  
END_TYPE;  
  
TYPE IfcObjectTypeEnum = ENUMERATION OF  
( PRODUCT,  
  PROCESS,  
  CONTROL,  
  RESOURCE,  
  ACTOR,  
  GROUP,  
  PROJECT,
```

```
NOTDEFINED);
END_TYPE;

TYPE IfcWorkControlTypeEnum = ENUMERATION OF
(ACTUAL,
BASELINE,
PLANNED,
USERDEFINED,
NOTDEFINED);
END_TYPE;

ENTITY IfcProduct
ABSTRACT SUPERTYPE
SUBTYPE OF(IfcObject);
Representation : OPTIONAL fp_represent_product;
ObjectPlacement : OPTIONAL fp_place_object;
WHERE
WR1 : (EXISTS(Representation) AND EXISTS(ObjectPlacement))
OR (EXISTS(Representation) AND
(NOT('IFC2X2_FINAL.IFCPRODUCTDEFINITIONSHAPE' IN TYPEOF(Representation))))
OR (NOT(EXISTS(Representation)));
END_ENTITY;

ENTITY IfcObject
ABSTRACT SUPERTYPE OF (ONEOF(IfcControl, IfcProcess, IfcProduct))
SUBTYPE OF(IfcObjectDefinition);
ObjectType : OPTIONAL IfcLabel;
WHERE
WR1 : SIZEOF(QUERY(temp < \* IsDefinedBy \ | 'IFC2X2_FINAL.IFCRELDEFINESBYTYPE' IN TYPEOF(temp))) <= 1;
END_ENTITY;

ENTITY IfcObjectDefinition
SUBTYPE OF(IfcRoot);
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
END_ENTITY;

ENTITY IfcRoot
ABSTRACT SUPERTYPE OF (ONEOF(IfcRelationship, IfcObjectDefinition));
GlobalId : IfcGloballyUniqueId;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
OwnerHistory : fp_apply_owner_history;
UNIQUE
UR1 : GlobalId;
END_ENTITY;

ENTITY IfcRelationship
ABSTRACT SUPERTYPE
SUBTYPE OF(IfcRoot);
END_ENTITY;
```

```
ENTITY IfcRelAssigns
ABSTRACT SUPERTYPE OF (ONEOF(IfcRelAssignsToControl, IfcRelAssignsToProcess))
SUBTYPE OF(IfcRelationship);
RelatedObjects : SET [1:?] OF IfcObject;
RelatedObjectsType : OPTIONAL IfcObjectTypeEnum;
WHERE
WR1 : IfcCorrectObjectAssignment(RelatedObjectsType, RelatedObjects);
END_ENTITY;

ENTITY IfcRelAssignsToControl
SUBTYPE OF(IfcRelAssigns);
RelatingControl : IfcControl;
WHERE
WR1 : SIZEOF(QUERY(Temp <\< * SELF\IfcRelAssigns.RelatedObjects \ | RelatingControl ::= Temp)) = 0;
END_ENTITY;

ENTITY IfcRelAssignsTasks
SUBTYPE OF(IfcRelAssignsToControl);
TimeForTask : OPTIONAL IfcScheduleTimeControl;
WHERE
WR1 : HIINDEX(SELF\IfcRelAssigns.RelatedObjects) = 1;
WR2 : 'IFC2X2_FINAL.IFCTASK' IN TYPEOF(SELF\IfcRelAssigns.RelatedObjects[1]);
WR3 : 'IFC2X2_FINAL.IFCWORKCONTROL' IN TYPEOF(SELF\IfcRelAssignsToControl.RelatingControl);
END_ENTITY;

ENTITY IfcScheduleTimeControl
SUBTYPE OF(IfcControl);
ActualStart : OPTIONAL fp_select_date_time;
EarlyStart : OPTIONAL fp_select_date_time;
LateStart : OPTIONAL fp_select_date_time;
ScheduleStart : OPTIONAL fp_select_date_time;
ActualFinish : OPTIONAL fp_select_date_time;
EarlyFinish : OPTIONAL fp_select_date_time;
LateFinish : OPTIONAL fp_select_date_time;
ScheduledFinish : OPTIONAL fp_select_date_time;
ScheduleDuration : OPTIONAL IfcTimeMeasure;
ActualDuration : OPTIONAL IfcTimeMeasure;
RemainingTime : OPTIONAL IfcTimeMeasure;
FreeFloat : OPTIONAL IfcTimeMeasure;
TotalFloat : OPTIONAL IfcTimeMeasure;
IsCritical : OPTIONAL BOOLEAN;
StatusTime : OPTIONAL fp_select_date_time;
StartFloat : OPTIONAL IfcTimeMeasure;
FinishFloat : OPTIONAL IfcTimeMeasure;
Completion : OPTIONAL IfcPositiveRatioMeasure;
INVERSE
ScheduleTimeControlAssigned : IfcRelAssignsTasks FOR TimeForTask;
END_ENTITY;

ENTITY IfcControl
ABSTRACT SUPERTYPE OF (ONEOF(IfcScheduleTimeControl, IfcWorkControl))
SUBTYPE OF(IfcObject);
```

```
INVERSE
Controls : SET OF IfcRelAssignsToControl FOR RelatingControl;
END_ENTITY;

ENTITY IfcWorkControl
ABSTRACT SUPERTYPE
SUBTYPE OF(IfcControl);
Identifier : IfcIdentifier;
CreationDate : fp_select_date_time;
Purpose : OPTIONAL IfcLabel;
Duration : OPTIONAL IfcTimeMeasure;
TotalFloat : OPTIONAL IfcTimeMeasure;
StartTime : fp_select_date_time;
FinishTime\_ : fp_select_date_time;
WorkControlType : OPTIONAL IfcWorkControlTypeEnum;
UserDefinedControlType : OPTIONAL IfcLabel;
WHERE
WR1 : (WorkControlType <> IfcWorkControlTypeEnum.USERDEFINED) OR
((WorkControlType = IfcWorkControlTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcWorkControl.UserDefinedControlType));
END_ENTITY;

ENTITY IfcWorkPlan
SUBTYPE OF(IfcWorkControl);
END_ENTITY;

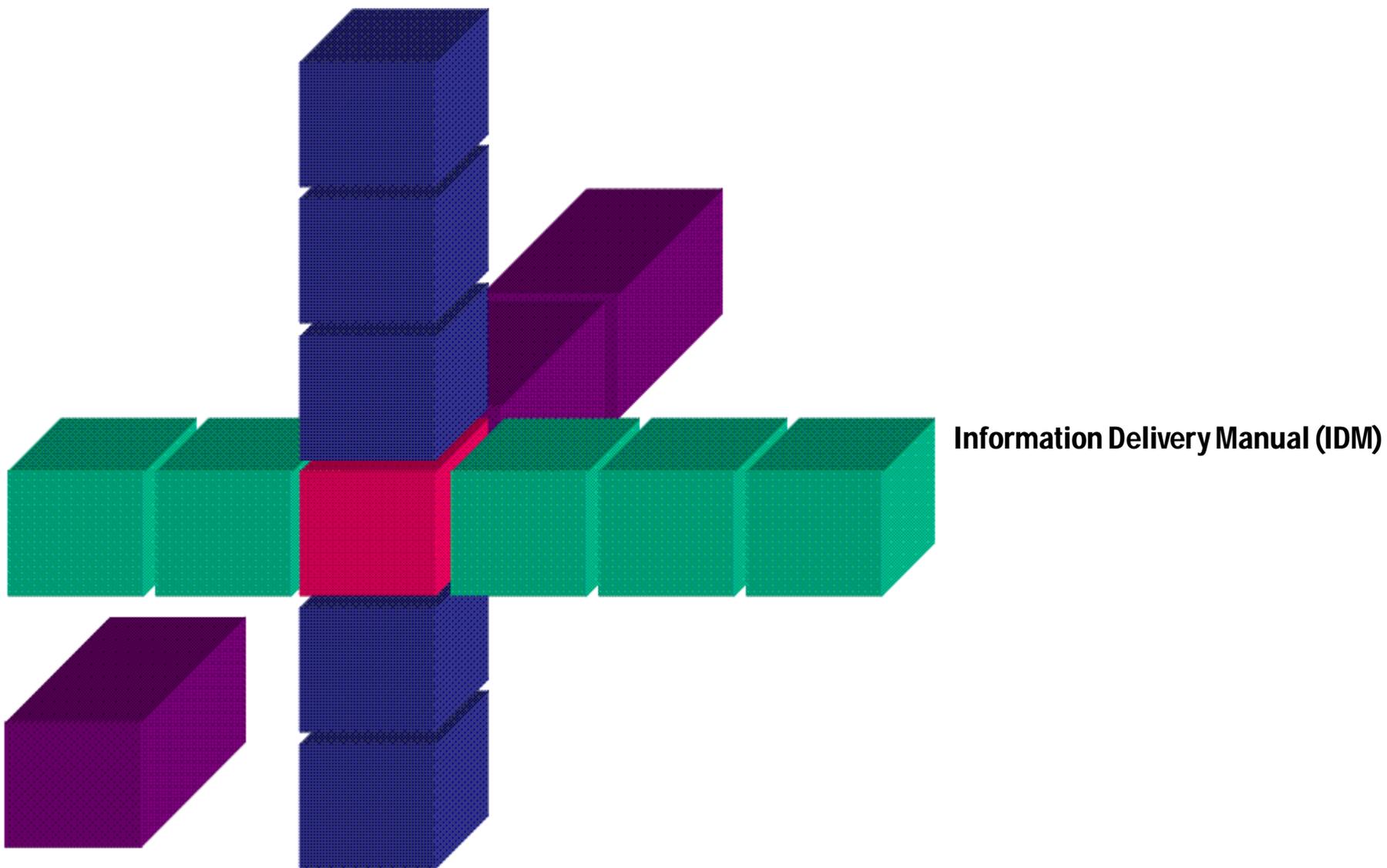
ENTITY IfcRelAssignsToProcess
SUBTYPE OF(IfcRelAssigns);
RelatingProcess : IfcProcess;
QuantityInProcess : fp_measure_with_unit;
WHERE
WR1 : SIZEOF(QUERY(Temp <\* SELF\IfcRelAssigns.RelatedObjects \ | RelatingProcess ::= Temp)) = 0;
END_ENTITY;

ENTITY IfcProcess
ABSTRACT SUPERTYPE
SUBTYPE OF(IfcObject);
END_ENTITY;

ENTITY IfcTask
SUBTYPE OF(IfcProcess);
TaskId : IfcIdentifier;
Status : OPTIONAL IfcLabel;
WorkMethod : OPTIONAL IfcLabel;
IsMilestone : BOOLEAN;
Priority : OPTIONAL INTEGER;
WHERE
WR1 : SIZEOF(QUERY(temp <\* SELF\IfcObject.Decomposes \ | NOT('IFC2X2_FINAL.IFCRELNESTS' IN TYPEOF(temp)))) = 0;
WR2 : SIZEOF(QUERY(temp <\* SELF\IfcObject.IsDecomposedBy \ | NOT('IFC2X2_FINAL.IFCRELNESTS' IN TYPEOF(temp)))) = 0;
WR3 : EXISTS(SELF\IfcRoot.Name);
END_ENTITY;

ENTITY fp_apply_owner_history;
```

```
END_ENTITY;  
  
ENTITY fp_assign_construction_resource_to_task;  
END_ENTITY;  
  
ENTITY fp_define_by_properties;  
END_ENTITY;  
  
ENTITY fp_measure_with_unit;  
END_ENTITY;  
  
ENTITY fp_model_asset;  
END_ENTITY;  
  
ENTITY fp_nests;  
END_ENTITY;  
  
ENTITY fp_place_object;  
END_ENTITY;  
  
ENTITY fp_represent_product;  
END_ENTITY;  
  
ENTITY fp_select_date_time;  
END_ENTITY;  
  
END_SCHEMA;
```



## Model Asset (FP)

Added by [Janice Wix](#), last edited by [Janice Wix](#) on Jan 30, 2007 ([view change](#))

Labels: (None)

# Model Asset

---

## Overview

---

Provides the information concerning assets where an asset is a grouping of elements that are identified together for financial or operational or maintenance purposes. Any type of physical element or product may be grouped into an asset.

Items grouped within an asset may be individual elements of groups of elements that are already grouped into an asset but that are treated as being further grouped within a 'super' asset. Such contained assets are described below as sub assets but are recognised within the schema as assets in their own right.

### Assets

An asset can be either a single object or may be a group of objects upon which a single maintenance action is to be specified. For instance, the following scenarios can be envisaged for asset identification: -

- A large element such as an air handling unit, boiler or transformer will probably be uniquely identified as an asset.
- An asset may be an 'assembled group' where several individual elements are physically assembled. An example of a physical association in the illustration is the pump set which is brings together the two pumps and four valves shown (it may also include other elements such as strainers). In this case, the asset may be defined as the grouping of the multiple elements or as a group containing the assembly alone (selection of the approach is a strategic decision by the Facilities Manager and is not relevant to the schema)
- An asset may be a 'group' where several items are spatially associated. Spatial association means that the individual elements within the group remain distinct but that they are all treated as though they were a single element for maintenance purposes. An example of a spatial association in the illustration is the luminaires, each of which is physically distinct but which act together to illuminate the plant room space and are maintained together e.g. for relamping.
- An asset may be declared as a group of elements with a system identity such as the steam traps within a system (or subsystem) where a maintenance action can only be carried out when the system is closed down.
- An asset may be declared as a group of elements that may be geographically located in different buildings or even on different sites within the same project but that are subject to maintenance at the same time. For instance, for an asset that contains specialist equipment, it might be appropriate to arrange for preventive maintenance to be carried out on all items of such equipment within a particular geographical region when the skilled human resource available to perform the maintenance work is available.

Note that for the purposes of this functional part, maintenance is always carried out on an asset and not on an element. Even where an element is sufficiently large or complex to warrant a maintenance action in its own right, it is considered that it should be declared and identified as an asset for the owner as well as being identified as a product by its manufacturer.

### Sub-Assets

Although work is carried out against the asset, information may need to be subdivided to take into account smaller element groupings within the asset. These smaller element groupings are referred to here as sub assets.

Sub-assets, and their constituent elements, may have a different life cycle than the asset as a whole. For instance, an air handling unit asset may have a fan motor and drive set specified as a sub-asset. The belts used to drive the fan may be consumable components i.e. they wear out and require replacing far more frequently than the air handling unit.

Sub assets within an air handling unit asset may include:

- Fan (motor[s], impeller, drive wheels and belts)
- Filter
- Preheater coil
- Reheater coil
- Cooling coil
- Dampers (fresh air and recirculation)
- Moisture eliminator
- Humidifier

#### Information Provision

The information that should, or may be provided directly about an asset includes:

- The specific identifier given to the asset which is unique within the project or enterprise to which the asset is registered
- Cost values for the asset including original cost (or value) current cost, replacement cost, depreciated cost
- Actors associated with the asset including owner, user, responsible person

Information that may be provided indirectly about an asset through the provision of property sets includes accounting, tax and insurance type identification.

#### Suggested modifications to IFC model and documentation

\* Documentation for Pset\_Asset.AssetAccountingType refers to type of risk and not to accounting type. Change definition.

\* In the documentation for Pset\_Asset.AssetTaxType, the predefined type 'Capitalised' is spelt in UK form and not in US form as mandated through IFC. Change this to 'Capitalized'

## Results

Specification of relevant information concerning the asset and the elements and/or sub-assets from which it is composed.

Description	Entity/Pset/Functional Part	MAN	REC	OPT
<b>Establish and identify the asset</b>				
<p>Specify the asset occurrence in which elements and/or sub assets will participate</p> <p><i>An asset is directly specified as an occurrence. It does not have a defining type entity.</i></p>	IfcAsset	✔		

<p><b>Assert general attributes for the asset</b></p>				
<p>Set the global unique identifier</p>	IfcAsset.GlobalId: IfcGloballyUniqueId	✓		
<p>Assert the owner history</p>	IfcAsset.OwnerHistory: fp_apply_owner_history	✓		
<p>Specify the name of the asset.</p> <p><i>Assets are not generally specified by name but by the unique asset identifier (see below). However, it may be useful to assert a name for human identification purposes</i></p>	IfcAsset.Name: IfcLabel			✓
<p>Specify a description for the asset</p> <p><i>Whilst the description does not add value to the semantics of the asset, it can provide significant information for use in the asset register and thus should be asserted.</i></p>	IfcAsset.Description: IfcText		✓	
<p><b>Assert the locally unique identifier of the asset</b></p>				
<p>Set the locally unique identifier of the asset</p> <p><i>An asset identifier is a locally unique identification assigned to an asset that enables its differentiation from other assets within an asset register. It differs from the globally unique id assigned to the instance of an entity populating a</i></p>	IfcAsset.AssetID: IfcIdentifier	✓		

<p>database</p>				
<p><b>Assert the date of incorporation of the asset</b></p>				
<p>Set the date on which the asset was incorporated into the project.</p> <p><i>Incorporation is taken here to mean the date from which the asset started depreciating and will generally be considered to be the same as the date at which construction/ erection/ installation was completed.</i></p>	<p>IfcAsset.IncorporationDate::IfcCalendarDate</p>			
<p>Set the year value</p> <p><i>Note that a year component is a mandatory requirement of a calendar date and therefore must be set even though it is not a requirement in this context.</i></p>	<p>IfcCalendarDate.YearComponent::IfcYearNumber</p>			
<p>Set the month value</p>	<p>IfcCalendarDate.MonthComponent::IfcMonthInYearNumber</p>			
<p>Set the day value</p>	<p>IfcCalendarDate.DayComponent::IfcDayInMonthNumber</p>			
<p><b>Assert cost information about the asset</b></p> <p><i>Note that all information asserted below is for costs that are explicitly attributable to an asset and that comprise the key cost information within an asset register. Other costs may be associated with an asset through the IfcRelAssociatesAppliedValue relationship.</i></p>				
<p>Define the original value of the asset</p>	<p>IfcAsset.OriginalValue::IfcCostValue</p>			

<p><i>Original value should be asserted as the value or cost of the asset at the time when the initial asset register entry was created. Generally, the term value should be used for an asset as opposed to the term cost which is more applicable to the individual elements within the asset. Value is broadly determined here as an amount that would be returned on disposal of the asset.</i></p> <p><i>If all components of the asset were new at initiation, it is probable that value would be equal to the sum of the costs of a given type of all the elements in the asset group. However this is not a rule that can be applied within this functional part.</i></p>				
<p>Define the current value of the asset</p> <p><i>Current value should be asserted as the value or cost of the asset if purchased at the present time. The value may be shown as an actual current cost or may be related to a baseline date through accounting methods (DCF, NPV etc). The discount factor applied may be visible through the IfcCostValue.CostType = "Discount" assertion.</i></p>	IfcAsset.OriginalValue::IfcCostValue	✔		
<p>Define the total replacement cost of the asset</p> <p><i>Total replacement cost should be asserted as the cost of completely replacing the asset and including all work associated with such replacement. This cost should take account of any factors that are likely to inhibit replacement once a building is complete such location of physical constructions or services or other obstacles that may make removal and replacement uncertain, difficult, unsafe or impossible.</i></p> <p><i>Note that the cost value of total replacement cost should be given in the same</i></p>	IfcAsset.TotalReplacementCost::IfcCostValue	✔		

<p><i>terms as current value.</i></p>				
<p>Define the depreciated value of the asset</p> <p><i>Depreciated value should be asserted as the value or cost of the asset related to the date of incorporation. The value may be shown as related to a baseline date through accounting methods (DCF, NPV etc). The discount factor applied may be visible through the IfcCostValue.CostType = "Discount" assertion</i></p>	IfcAsset.DepreciatedValue::IfcCostValue	✓		
<p><b>Assert actor information about the asset</b></p> <p><i>Note that all information asserted below is for actor information that is explicitly attributable to an asset and that comprise the key actor information within an asset register.</i></p>				
<p>Assert ownership of the asset</p> <p><i>Owner may be defined either as a person or an organization through the actor selection functional part</i></p>	IfcAsset.Owner::fp_select_actor	✓		
<p>Assert the user of the asset</p> <p><i>The user of the asset may be a lessee, tenant or other person or organization obtaining beneficial use of an asset. The user may be distinct from the owner.</i></p>	IfcAsset.User::fp_select_actor	✓		
<p>Assert the person responsible for the asset</p>	IfcAsset.ResponsiblePerson::IfcPerson	✓		

<p>The person responsible may be defined as anyone having a particular fixed duty towards the asset. It may be e.g. a keyholder, someone who is authorized to switch an asset on/off etc.</p> <p>The person responsible is defined as a PERSON and not as an organization.</p>				
<p><b>Set accounting information for the asset</b></p>				
<p>Set the type of risk to be applied.</p> <p>Type of risk is taken from a predefined selection that includes:</p> <ul style="list-style-type: none"> <li>bq_* Fixed_</li> <li>bq_* NonFixed_</li> <li>bq_* Other_</li> <li>bq_* NotKnown_</li> <li>bq_* Unset_</li> </ul>	<p>Pset_Asset.AssetAccountingType: : IfcPropertyEnumeratedValue</p>			
<p>Set the type of taxation group to which the asset belongs.</p> <p><b>IFC Entities Required</b></p> <ul style="list-style-type: none"> <li>● IfcAppliedValue</li> <li>● IfcAsset</li> <li>● IfcCalendarDate</li> <li>● IfcCostValue</li> <li>● IfcElement</li> <li>● IfcGroup</li> <li>● IfcObject</li> <li>● IfcProduct</li> <li>● IfcProxy</li> </ul> <ul style="list-style-type: none"> <li>bq_* Capitalized_</li> <li>bq_* Expensed_</li> <li>bq_* Other_</li> <li>bq_* NotKnown_</li> <li>bq_* Unset_</li> </ul>				

- IfcRelationship
- IfcRelAssociates
- IfcRelAssociatesAppliedValue
- IfcRoot
- IfcSystem

## IFC Datatypes Required

---

- IfcAppliedValueSelect
- IfcDayInMonthNumber
- IfcGloballyUniqueId
- IfcIdentifier
- IfcLabel
- IfcMonetaryMeasure
- IfcMonthInYearNumber
- IfcObjectTypeEnum
- IfcText
- IfcYearNumber

## IFC Functions Required

---

- IfcValidCalendarDate

## IFC Property Sets Required

---

- Pset\_Asset

## IDM Functional Parts Required

---

- fp\_apply\_owner\_history
- fp\_define\_by\_properties
- fp\_place\_object
- fp\_measure\_with\_unit
- fp\_represent\_product
- fp\_select\_actor
- fp\_select\_date\_time

## EXPRESS-G

---

## EXPRESS Schema

```
SCHEMA FP_MODEL_ASSET;

TYPE IfcDayInMonthNumber = INTEGER;
END_TYPE;

TYPE IfcGloballyUniqueId = STRING (22) FIXED;
END_TYPE;

TYPE IfcIdentifier = STRING;
END_TYPE;

TYPE IfcLabel = STRING;
END_TYPE;

TYPE IfcMonetaryMeasure = REAL;
END_TYPE;

TYPE IfcMonthInYearNumber = INTEGER;
WHERE
  WR1 : { 1<= SELF <= 12 };
END_TYPE;

TYPE IfcText = STRING;
END_TYPE;

TYPE IfcYearNumber = INTEGER;
END_TYPE;

TYPE IfcObjectTypeEnum = ENUMERATION OF
  (PRODUCT,
   PROCESS,
   CONTROL,
   RESOURCE,
   ACTOR,
   GROUP,
   PROJECT,
   NOTDEFINED);
END_TYPE;

TYPE IfcAppliedValueSelect = SELECT
  (IfcMonetaryMeasure,
   fp_measure_with_unit);
END_TYPE;

ENTITY IfcCalendarDate;
  DayComponent      : IfcDayInMonthNumber;
  MonthComponent    : IfcMonthInYearNumber;
  YearComponent      : IfcYearNumber;
```

```
WHERE
  WR21 : IfcValidCalendarDate (SELF);
END_ENTITY;

ENTITY IfcCostValue
  SUBTYPE OF(IfcAppliedValue);
  CostType : IfcLabel;
  Condition : OPTIONAL IfcText;
END_ENTITY;

ENTITY IfcAppliedValue
  ABSTRACT SUPERTYPE;
  Name : OPTIONAL IfcLabel;
  Description : OPTIONAL IfcText;
  AppliedValue : OPTIONAL IfcAppliedValueSelect;
  UnitBasis : fp_measure_with_unit;
  ApplicableDate : fp_select_date_time;
  FixedUntilDate : fp_select_date_time;
  WHERE
    WR1 : EXISTS (AppliedValue) OR
          EXISTS (ValueOfComponents);
END_ENTITY;

ENTITY IfcProduct
  ABSTRACT SUPERTYPE OF (ONEOF(IfcElement, IfcProxy))
  SUBTYPE OF(IfcObject);
  Representation : OPTIONAL fp_represent_product;
  ObjectPlacement : OPTIONAL fp_place_object;
  WHERE
    WR1 : (EXISTS(Representation) AND EXISTS(ObjectPlacement))
          OR (EXISTS(Representation) AND
              (NOT('IFC2X2_FINAL.IFCPRODUCTDEFINITIONSHAPE' IN TYPEOF(Representation))))
          OR (NOT(EXISTS(Representation)));
END_ENTITY;

ENTITY IfcObject
  ABSTRACT SUPERTYPE OF (ONEOF(IfcGroup, IfcProduct))
  SUBTYPE OF(IfcRoot);
  ObjectType : OPTIONAL IfcLabel;
  INVERSE
  HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
  WHERE
    WR1 : SIZEOF(QUERY(temp <* IsDefinedBy | 'IFC2X2_FINAL.IFCRELDEFINESBYTYPE' IN TYPEOF(temp))) <= 1;
END_ENTITY;

ENTITY IfcRoot
  ABSTRACT SUPERTYPE OF (ONEOF(IfcObject, IfcRelationship));
  GlobalId : IfcGloballyUniqueId;
  Name : OPTIONAL IfcLabel;
  Description : OPTIONAL IfcText;
  OwnerHistory : fp_apply_owner_history;
  UNIQUE
```

```
    URI : GlobalId;
END_ENTITY;

ENTITY IfcRelationship
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcRoot);
END_ENTITY;

ENTITY IfcElement
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcProduct);
  Tag : OPTIONAL IfcIdentifier;
END_ENTITY;

ENTITY IfcGroup
  SUBTYPE OF(IfcObject);
END_ENTITY;

ENTITY IfcAsset
  SUBTYPE OF(IfcGroup);
  AssetId          : IfcIdentifier;
  OriginalValue    : IfcCostValue;
  CurrentValue     : IfcCostValue;
  TotalReplacementCost : IfcCostValue;
  Owner            : fp_select_actor;
  User             : fp_select_actor;
  IncorporationDate : IfcCalendarDate;
  DepreciatedValue : IfcCostValue;
  WHERE
    WR1 : SIZEOF(QUERY(Temp <* SELF\IfcGroup.IsGroupedBy.RelatedObjects | NOT('IFC2X2_FINAL.IFCELEMENT' IN TYPEOF(Temp)))) = 0;
END_ENTITY;

ENTITY IfcRelAssociates
  SUBTYPE OF(IfcRelationship);
  RelatedObjects : SET [1:?] OF IfcRoot;
  WHERE
    WR1 : SIZEOF(QUERY(temp <* RelatedObjects |
      NOT(('IFC2X2_FINAL.IFCOBJECT' IN TYPEOF(temp))
      OR ('IFC2X2_FINAL.IFCPROPERTYDEFINITION' IN TYPEOF(temp)))
    )) = 0;
END_ENTITY;

ENTITY IfcRelAssociatesAppliedValue
  SUBTYPE OF(IfcRelAssociates);
  RelatingAppliedValue : IfcAppliedValue;
END_ENTITY;

ENTITY IfcProxy
  SUBTYPE OF(IfcProduct);
  ProxyType : IfcObjectTypeEnum;
  Tag       : OPTIONAL IfcLabel;
  WHERE
```

```
        WR1 : EXISTS(SELF\IfcRoot.Name);
END_ENTITY;

ENTITY fp_apply_owner_history;
END_ENTITY;

ENTITY fp_represent_product;
END_ENTITY;

ENTITY fp_measure_with_unit;
END_ENTITY;

ENTITY fp_place_object;
END_ENTITY;

ENTITY fp_define_by_properties;
END_ENTITY;

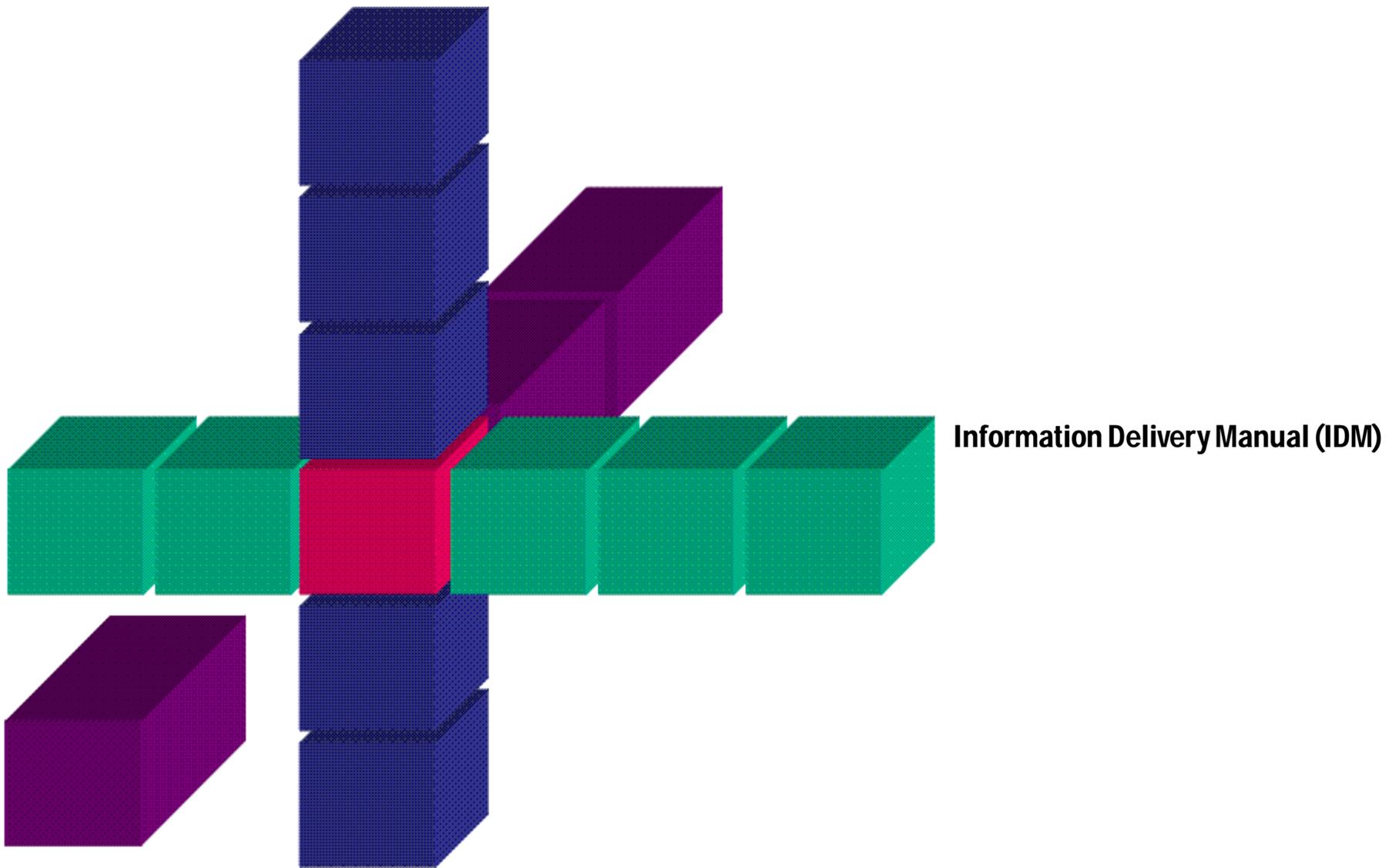
ENTITY fp_select_actor;
END_ENTITY;

ENTITY fp_select_date_time;
END_ENTITY;

FUNCTION IfcValidCalendarDate
(Date : IfcCalendarDate)
    : LOGICAL;
IF NOT ({1 <= Date.DayComponent <= 31}) THEN
    RETURN(FALSE);
END_IF;
CASE Date.MonthComponent OF
    4      : RETURN({ 1<= Date.DayComponent <= 30});
    6      : RETURN({ 1<= Date.DayComponent <= 30});
    9      : RETURN({ 1<= Date.DayComponent <= 30});
    11     : RETURN({ 1<= Date.DayComponent <= 30});
    2      :
BEGIN
    IF (IfcLeapYear(Date.YearComponent)) THEN
        RETURN({ 1<= Date.DayComponent <= 29});
    ELSE
        RETURN({ 1<= Date.DayComponent <= 28});
    END_IF;
END;
    OTHERWISE : RETURN(TRUE);
END_CASE;
END_FUNCTION;

END_SCHEMA;
```

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## Model Condition (FP)

Added by [Janice Wix](#), last edited by [Janice Wix](#) on Jan 31, 2007 ([view change](#))

Labels: (None)

# Model Condition

---

## Overview

---

Provides the information concerning the condition of an element or a group of elements

Condition may be established to determine if an element requires maintenance in order to bring it to a required level of operating efficiency. This may be brought about either by the absolute condition or by the rate of change of condition. Ultimately, the condition of an element may become so poor as to require replacement

The condition of an element may be established by:

- instrumentation in place that enables operating parameters to be continuously monitored; the parameters being selected such that they indicate the current condition of the asset (e.g. vibration).
- periodic inspection that may be either visual, carried out with the assistance of instruments or a combination.

The condition of an element is considered to be determined by one or several condition criteria. Each criterion is a particular measured or assessed value that contributes to the overall condition of the element.

- A criterion that is measured using calibrated instruments and whose value is repeatable by different instruments is considered to be objective. An objective criterion will typically be a measured value.
- A criterion that is assessed by observation and whose value may not be repeatable between assessors is considered to be subjective. A subjective criterion will typically be measured against an arbitrary scale (e.g. 1 represents new condition whilst 10 represents poor condition requiring urgent replacement).  
The actual condition of an element is considered to be determined by bringing together the various criteria within a group.  
Particular target values may be established for maintenance or replacement operations to be carried out on an element. Such values are defined as constraints. When the condition of an element reaches or exceeds a value indicated by a constraint, it signifies that a particular action should be triggered. A condition based 'event' may be based on either objective or subjective criteria. For objective criteria, the potential exists for live reporting of condition data from fixed instrumentation. Through the use of condition criteria, IFC can therefore offer the potential for 'expert' automated decision making in maintenance.

### Suggested modifications to IFC model and documentation

\* Date and time values are currently held for each condition criterion individually and not for condition. Therefore, for tracking condition history, each of the criterion dates within a condition must be the same so that they can also be applied to the condition. It would be better if a date and time could also be explicitly applied to the condition as well as to the condition criteria.

## Results

---

Specification of relevant information concerning the asset and the elements and/or sub-assets from which it is composed.

Description	Entity/Pset/Functional Part	MAN	REC	OPT
-------------	-----------------------------	-----	-----	-----

<p><b>Establish Condition Criteria</b></p> <p><i>The actual condition of an element may need to be determined by reference to several condition criteria. Each criterion may measure a different aspect of the condition and may be either a subjective or objective assessment.</i></p> <p><i>Each condition criterion should be established in turn.</i></p>				
<p><b>Assert general attributes for the criterion</b></p>				
<p>Set the global unique identifier</p>	IfcConditionCriterion.GlobalId: : IfcGloballyUniqueId			
<p>Assert the owner history</p>	IfcConditionCriterion.OwnerHistory: : fp_apply_owner_history			
<p>Specify the name of the criterion.</p> <p><i>Each condition criterion must have a name that identifies it.</i></p>	IfcConditionCriterion.Name: : IfcLabel			
<p>Specify a description for the criterion</p> <p><i>The description of a condition criterion may be used to provide further information if required. However, if the name of the criterion conforms to an agreed standard that also incorporates a description of what the name means, it may be unnecessary to include a description with every criterion.</i></p>	IfcConditionCriterion.Description: : IfcText			
<p><b>Set the date and time</b></p>				
<p>Set the date and time at which the condition assessment is made.</p>	IfcConditionCriterion.OwnerHistory: : fp_select_date_time			

<p><i>Date and time values are required to establish the point at which the assessment is made. It may also be used as the basis for historical condition recording within which knowledge of the dates of condition assessment can be used to determine the rate of degradation; information which can be useful in predicting actual service life values (see fp_control_service_life)</i></p> <p><i>Note that for a set of condition criteria that are grouped to define an element condition, the date and time value asserted for each criterion should be the same.</i></p>				
<p><b>Select whether a condition criterion will be subjective or objective</b></p> <p><i>A subjective criterion is defined as a condition that is assessed by human inspection (visual, audio, smell, touch etc)</i></p> <p><i>An objective criterion is defined as a condition that can be determined from the values of particular parameters found by reliable instruments</i></p>				
<p>If the criterion is subjective, assess the condition criterion value against an available scale</p> <p><i>For subjective condition criteria, a scale of values should be established that can be easily understood and interpreted by subsequent condition investigators.</i></p>	IfcConditionCriterion.Criterion::IfcLabel			
<p>If the criterion is objective, determine the value and units</p>	IfcConditionCriterion.Criterion::fp_measure_with_unit			
<p><b>Assert the relationship that assigns the condition criteria that will establish the condition</b></p>	fp_assigns_to_group (condition)			

<p><i>This is defined in a separate functional part. The relating group in this functional part will be the entity IfcCondition. The related objects will be the instances of IfcConditionCriterion that establish the condition.</i></p> <p><i>Note the rule that requires the date and time value for each condition criterion within the group to have the same value.</i></p>				
<p><b>Establish and identify the element whose condition is to be assessed</b></p>				
<p>Specify the element or group whose condition is to be assessed</p> <p><i>Any subtype of IfcProduct can be identified for this purpose. Practically, certain subtypes may be excluded including site, building storey and space. However, this is not considered to form a general business rule since it may be appropriate to use e.g. space as a contained for a general condition assessment of all the elements that are contained within it.</i></p> <p><i>Occurrences of IfcSystem (including occurrences of IfcElectricalCircuit) and IfcAsset may also be subject to condition assessment.</i></p>	<p>IfcProduct &lt;subtypes&gt; OR IfcSystem &lt;subtypes&gt; OR IfcAsset</p>			
<p><b>Assert the relationship that assigns the condition to the product or group</b></p>				
<p>If the condition is being assigned to an element The relating product is the product subtype selected and the related objects is the condition that is being assigned.</p>	<p>fp_assigns_to_product</p>			
<p>If the condition is being assigned to an asset or system The relating group is the asset, system or subtype of system selected and the related objects is the condition that is being assigned.</p>	<p>fp_assigns_to_group</p>			

## IFC Entities Required

- IfcAppliedValue
- IfcCalendarDate
- IfcCondition
- IfcConditionCriterion
- IfcControl
- IfcCostValue

- IfcGroup
- IfcObject
- IfcProduct
- IfcRoot
- IfcSystem

## IFC Datatypes Required

---

- IfcAppliedValueSelect
- IfcConditionCriterionSelect
- IfcDayInMonthNumber
- IfcGloballyUniqueId
- IfcLabel
- IfcMonetaryMeasure
- IfcMonthInYearNumber
- IfcText
- IfcYearNumber

## IFC Functions Required

---

- -

## IFC Property Sets Required

---

- -

## IDM Functional Parts Required

---

- fp\_apply\_owner\_history
- fp\_assigns\_to\_group
- fp\_assigns\_to\_product
- fp\_measure\_with\_unit
- fp\_place\_object
- fp\_represent\_product
- fp\_select\_actor
- fp\_select\_date\_time

## EXPRESS-G

---

## EXPRESS Schema

```
SCHEMA FP_MODEL_CONDITION;  
  
  TYPE IfcDayInMonthNumber = INTEGER;  
  END_TYPE;  
  
  TYPE IfcGloballyUniqueId = STRING (22) FIXED;  
  END_TYPE;  
  
  TYPE IfcLabel = STRING;  
  END_TYPE;  
  
  TYPE IfcMonthInYearNumber = INTEGER;  
  WHERE  
    WR1 : { 1<= SELF <= 12 };  
  END_TYPE;  
  
  TYPE IfcText = STRING;  
  END_TYPE;  
  
  TYPE IfcYearNumber = INTEGER;  
  END_TYPE;  
  
  TYPE IfcConditionCriterionSelect = SELECT  
    (IfcLabel,  
     fp_measure_with_unit);  
  END_TYPE;  
  
  TYPE IfcMonetaryMeasure = REAL;  
  END_TYPE;  
  
  TYPE IfcAppliedValueSelect = SELECT  
    (IfcMonetaryMeasure,  
     fp_measure_with_unit);  
  END_TYPE;  
  
  ENTITY IfcProduct  
    ABSTRACT SUPERTYPE  
    SUBTYPE OF(IfcObject);  
    Representation : OPTIONAL fp_represent_product;  
    ObjectPlacement : OPTIONAL fp_place_object;  
    WHERE  
      WR1 : (EXISTS(Representation) AND EXISTS(ObjectPlacement))  
            OR (EXISTS(Representation) AND  
                (NOT('IFC2X2_FINAL.IFCPRODUCTDEFINITIONSHAPE' IN TYPEOF(Representation))))  
            OR (NOT(EXISTS(Representation)));  
  END_ENTITY;  
  
  ENTITY IfcObject
```

```
ABSTRACT SUPERTYPE OF (ONEOF(IfcControl, IfcGroup, IfcProduct))
SUBTYPE OF(IfcRoot);
  ObjectType      : OPTIONAL IfcLabel;
WHERE
  WR1 : SIZEOF(QUERY(temp <* IsDefinedBy | 'IFC2X2_FINAL.IFCRELDEFINESBYTYPE' IN TYPEOF(temp))) <= 1;
END_ENTITY;

ENTITY IfcRoot
ABSTRACT SUPERTYPE OF (ONEOF(IfcObject, IfcRelationship));
  GlobalId       : IfcGloballyUniqueId;
  Name           : OPTIONAL IfcLabel;
  Description    : OPTIONAL IfcText;
  OwnerHistory  : fp_apply_owner_history;
UNIQUE
  UR1 : GlobalId;
END_ENTITY;

ENTITY IfcControl
ABSTRACT SUPERTYPE
SUBTYPE OF(IfcObject);
END_ENTITY;

ENTITY IfcConditionCriterion
SUBTYPE OF(IfcControl);
  Criterion       : IfcConditionCriterionSelect;
  CriterionDateTime : fp_select_date_time;
WHERE
  WR1 : EXISTS(SELF\IfcRoot.Name);
END_ENTITY;

ENTITY IfcSystem
SUBTYPE OF(IfcGroup);
WHERE
  WR1 : SIZEOF (QUERY (temp <* SELF\IfcGroup.IsGroupedBy.RelatedObjects | NOT('IFC2X2_FINAL.IFCELEMENT' IN TYPEOF(temp)))) = 0;
END_ENTITY;

ENTITY IfcGroup
SUPERTYPE OF (ONEOF(IfcCondition, IfcSystem, IfcAsset))
SUBTYPE OF(IfcObject);
END_ENTITY;

ENTITY IfcCondition
SUBTYPE OF(IfcGroup);
END_ENTITY;

ENTITY IfcCalendarDate;
  DayComponent   : IfcDayInMonthNumber;
  MonthComponent : IfcMonthInYearNumber;
  YearComponent  : IfcYearNumber;
WHERE
  WR21 : IfcValidCalendarDate (SELF);
END_ENTITY;
```

```
ENTITY IfcCostValue
  SUBTYPE OF(IfcAppliedValue);
END_ENTITY;

ENTITY IfcAppliedValue
  ABSTRACT SUPERTYPE;
  AppliedValue      : OPTIONAL IfcAppliedValueSelect;
  UnitBasis         : fp_measure_with_unit;
  ApplicableDate    : fp_select_date_time;
  FixedUntilDate    : fp_select_date_time;
  WHERE
    WR1 : EXISTS (AppliedValue) OR
          EXISTS (ValueOfComponents);
END_ENTITY;

ENTITY IfcAsset
  SUBTYPE OF(IfcGroup);
  OriginalValue      : IfcCostValue;
  CurrentValue       : IfcCostValue;
  TotalReplacementCost : IfcCostValue;
  Owner              : fp_select_actor;
  User               : fp_select_actor;
  IncorporationDate  : IfcCalendarDate;
  DepreciatedValue   : IfcCostValue;
  WHERE
    WR1 : SIZEOF(QUERY(Temp <* SELF\IfcGroup.IsGroupedBy.RelatedObjects | NOT('IFC2X2_FINAL.IFCELEMENT' IN TYPEOF(Temp)))) = 0;
END_ENTITY;

ENTITY fp_apply_owner_history;
END_ENTITY;

ENTITY fp_assigns_to_group;
END_ENTITY;

ENTITY fp_assigns_to_product;
END_ENTITY;

ENTITY fp_measure_with_unit;
END_ENTITY;

ENTITY fp_place_object;
END_ENTITY;

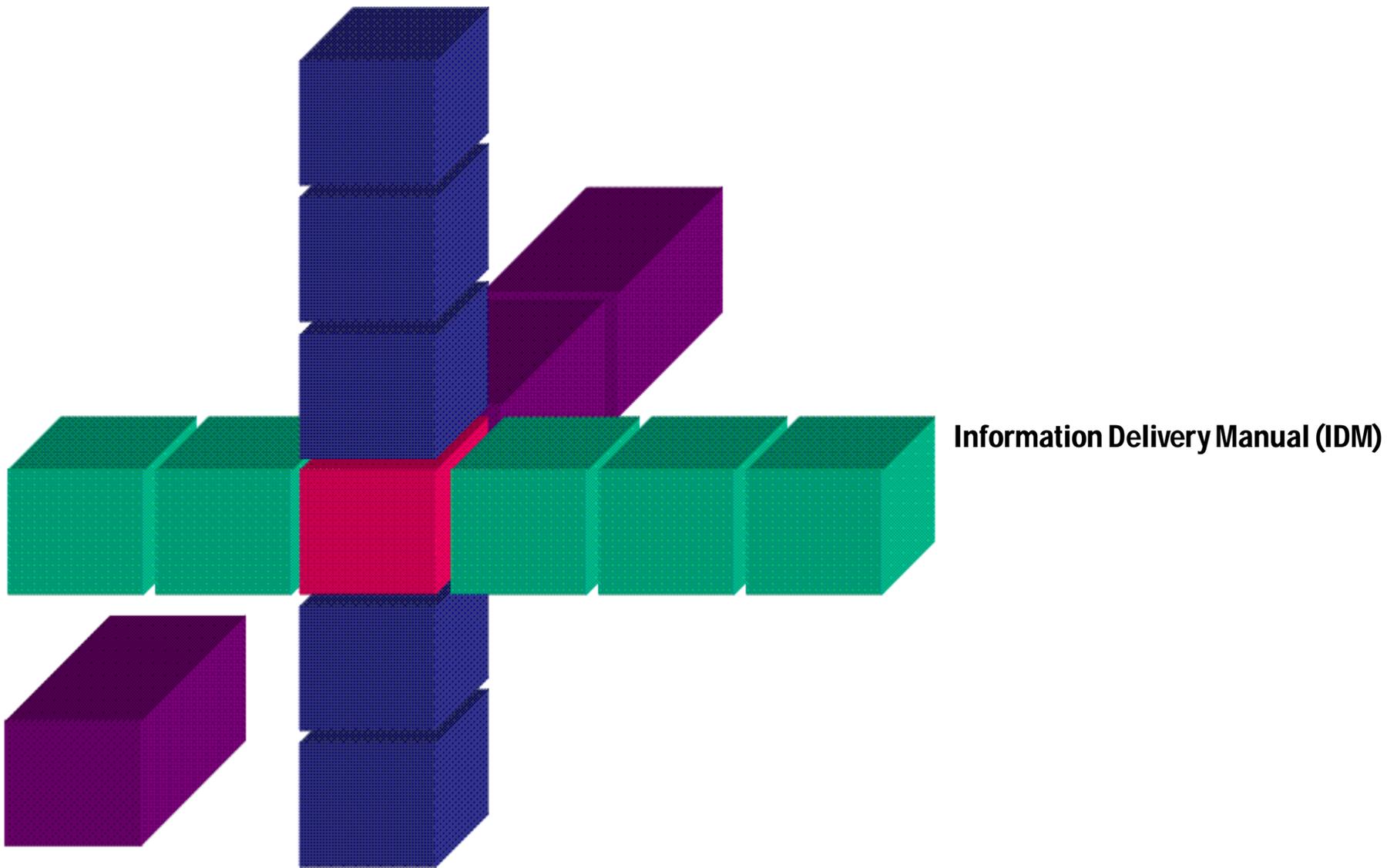
ENTITY fp_represent_product;
END_ENTITY;

ENTITY fp_select_actor;
END_ENTITY;

ENTITY fp_select_date_time;
END_ENTITY;
```

```
END_SCHEMA ;
```

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)



## Model Inventory (FP)

Added by [Janice Wix](#), last edited by [Janice Wix](#) on Feb 05, 2007 ([view change](#))

Labels: (None)

# Model Inventory

---

## Overview

---

This functional part enables an inventory of various types of element to be compiled as explicit objects. These are the typical inventories that would be expected within a facilities management requirement and enable lists of equipment to be maintained together with financial information concerning the items in the inventory.

The major types of inventory that can be captured include:

- Asset
- Furniture
- Space

In addition, user defined inventories can be defined for particular types of element. Such lists may be useful when developing operating and maintenance information.

Information that can be included within an inventory includes:

- Identifier of the type of inventory (asset, furniture or space)
- The list of elements (furniture), groups (assets) or spaces that comprise the inventory
- The organizational unit within whose jurisdiction the inventory falls
- The identity of the person whose responsibility it is to maintain the inventory
- The date on which the inventory was last updated
- The current value of the inventory (totalled from the value of the contents)
- The initial value of the inventory

### Basic Development of Inventory

Technically, an inventory is simply a group that is constrained by certain rules to contain only certain other entities. However, unlike a system that can contain many different types of entities that acting together in a 'systematic' way, an inventory generally contains only a single type of entity (even if the entity is devised at a high level of abstraction within the IFC schema).

Thus, the functional part development can be seen as:

- Identify the entity type
- Identify all relevant occurrences of that entity type
- Collect them together into a group
- Identify the group as an inventory
- Add further attributes to the inventory.

### Concepts of inventory

This functional part deals with the following inventory concepts:

1. Asset inventory model\_inventory (asset)
2. Furniture inventory model\_inventory (furnishing\_element)
3. Space inventory model\_inventory (space)

## Suggested modifications to IFC model and documentation

\* Current and original cost values for inventories are explicitly captured. Although this is a legal use of attributes within the model, it is now more normal to expect cost values to be associated with entities via `IfcRelAssociatesAppliedValue` relationship. Therefore, these attributes should be deprecated and the provision of cost values provided through the `fp_associate_cost` capability. In this case, `CurrentValue` and `OriginalValue` should both be asserted as cost types in `IfcCostValue`

## Results

---

Inventory of entity of interest is captured together with relevant date, cost and organizational information.

Description	Entity/Pset/Functional Part	MAN	REC	OPT
<b>Create the inventory in which entities will be recorded.</b>				
Set the occurrence of the inventory	<code>IfcInventory</code>	✓		
Set the global unique identifier	<code>IfcInventory.GlobalId : IfcGloballyUniqueId</code>	✓		
Apply the owner history <code>IfcInventory</code> .	<code>OwnerHistory : fp_apply_owner_history</code>	✓		
Specify the name of the inventory  <i>Several inventories of the same type may be maintained by an organization so as to be able to identify ownership of items by e.g. department, building part etc.). Each inventory should carry an immediately identifiable name to support this.</i>	<code>IfcInventory.Name : IfcLabel</code>		✓	
Specify a description for the inventory	<code>IfcInventory.Description : IfcText</code>			✓
<b>Assert the type of inventory</b>				
Select the inventory type	<code>IfcInventory.InventoryType : IfcInventoryTypeEnum</code>	✓		

<p>Selection should be made from the predefined list available that includes:</p> <ul style="list-style-type: none"> <li>● Asset Inventory</li> <li>● Space Inventory</li> <li>● Furniture Inventory</li> <li>● User Defined</li> <li>● Not Defined</li> </ul>				
<p><b>Concept of asset inventory</b>  From the selection of the type of inventory, the entities that can be collected are controlled by rules to ensure that only those entities that are appropriate are included.</p>				
<p>Collect the entities that are to be within the inventory</p> <p>If the value of the inventory type specified above is 'ASSETINVENTORY' then all of the entities in the collection shall be occurrences of IfcAsset.</p>	fp_model_asset (IfcAsset)	✔		
<p><b>Concept of furniture inventory</b></p>				
<p>Collect the entities that are to be within the inventory</p> <p>If the value of the inventory type specified above is 'FURNITUREINVENTORY' then all of the entities in the collection shall be occurrences of IfcFurnishingElement.</p>	fp_model_furniture (IfcFurnishingElement)	✔		
<p><b>Concept of space inventory</b></p>				
<p>Collect the entities that are to be within the inventory</p>	fp_model_space (IfcSpace)	✔		

<p><i>If the value of the inventory type specified above is 'SPACEINVENTORY' then all of the entities in the collection shall be occurrences of IfcSpace.</i></p>				
<b>Assign the inventoried collection to the inventory instance</b>				
<p>Assert the inventory</p>	IfcRelAssignsToGroup.RelatingGroup::IfcInventory	✓		
<p>Assert the collection into the inventory</p>	IfcRelAssignsToGroup.RelatedObjects::IfcAsset OR IfcRelAssignsToGroup.RelatedObjects::IfcFurnishingElement OR IfcRelAssignsToGroup.RelatedObjects::IfcSpace	✓		
<p>Assert the type of related objects from the predefined list if required</p> <p><i>This value can be used to identify the parent supertype of related objects as e.g. product, group, process, control etc. If the related objects have mixed superotypes, then the value .NOTDEFINED. should be used.</i></p> <p><i>Use .GROUP. for asset inventory</i></p> <p><i>Use .PRODUCT. for furniture or space inventory</i></p>	IfcRelAssignsToGroup.RelatedObjectType::IfcObjectTypeEnum			✓
<p>Assert the identity of the relationship</p>	IfcRelAssignsToGroup.GlobalId::IfcGloballyUniqueId	✓		
<p>Assert the owner history of the relationship</p>	IfcRelAssignsToGroup.OwnerHistory::fp_apply_owner_history	✓		
<p>Specify a name of the relationship</p>	IfcRelAssignsToGroup.Name		✓	

Specify a description of the relationship	IfcRelAssignsToGroup.Description			
<b>Assert additional cost and actor attributes for the inventory</b>				
<p>Define the organizational unit to which the inventory is applicable</p> <p><i>Jurisdiction may be either an organization or a person (or both in the case of a person and organization). Typically however, the organization unit is anticipated to be an occurrence of IfcOrganization</i></p>	IfcInventory.Jurisdiction::fp_select_actor			
<p>Designate the person or persons who are responsible for the inventory</p> <p><i>Responsibility is always considered to rest with one or more person. The target datatype of this attribute must therefore be IfcPerson.</i></p> <p><i>Several occurrences of IfcPerson may be designated as having responsibility. There is no upper limit to the number providing that there is at least one.</i></p>	IfcInventory.ResponsiblePersons::fp_select_actor			
Record the date on which the last update of the inventory was carried out	IfcInventory.LastUpdateDate::IfcCalendarDate			
Record the day component	IfcCalendarDate.DayComponent::IfcDayInMonthNumber			
Record the month component	IfcCalendarDate.MonthComponent::IfcMonthInYearNumberNumber			
Record the year component	IfcCalendarDate.YearComponent::IfcYearNumberNumber			

<p>The following cost information for inventories is explicitly captured. However, these attributes should be deprecated and the provision of cost values provided through the fp_associate_cost capability.</p> <p><i>CurrentValue and OrginalValue should both be asserted as cost types in IfcCostValue</i></p>				
<p>Make an estimate of the current cost value of the inventory</p> <p>_For details of attribute settings for IfcCostValue, refer to fp_associate_cost _</p>	IfcInventory.CurrentValue:: IfcCostValue			
<p>Make an estimate of the original cost value of the inventory</p> <p><i>For details of attribute settings for IfcCostValue, refer to fp_associate_cost _</i></p>	IfcInventory.OriginalValue:: IfcCostValue			

## IFC Entities Required

- IfcCalendarDate
- IfcCostValue
- IfcGroup
- IfcInventory
- IfcObject
- IfcRelationship
- IfcRelAssigns
- IfcRelAssignsToGroup
- IfcRoot

## IFC Datatypes Required

---

- IfcAppliedValueSelect
- IfcDayInMonthNumber
- IfcGloballyUniqueId
- IfcIdentifier
- IfcInventoryTypeEnum
- IfcLabel
- IfcMonetaryMeasure
- IfcMonthInYearNumber
- IfcObjectTypeEnum
- IfcRatioMeasure
- IfcText
- IfcYearNumber

## IFC Functions Required

---

- IfcValidCalendarDate

## IFC Property Sets Required

---

- -

## IDM Functional Parts Required

---

- fp\_apply\_owner\_history
- fp\_associate\_cost
- fp\_measure\_with\_unit
- fp\_model\_asset
- fp\_model\_furniture
- fp\_model\_space
- fp\_select\_actor
- fp\_select\_date\_time

## EXPRESS-G

---

## EXPRESS Schema

---

```
SCHEMA FP_MODEL_INVENTORY;  
  
TYPE IfcGloballyUniqueId = STRING (22) FIXED;  
END_TYPE;  
  
TYPE IfcLabel = STRING;  
END_TYPE;  
  
TYPE IfcMonetaryMeasure = REAL;  
END_TYPE;  
  
TYPE IfcRatioMeasure = REAL;  
END_TYPE;  
  
TYPE IfcText = STRING;  
END_TYPE;  
  
TYPE IfcInventoryTypeEnum = ENUMERATION OF  
  (ASSETINVENTORY,  
   SPACEINVENTORY,  
   FURNITUREINVENTORY,  
   USERDEFINED,  
   NOTDEFINED);  
END_TYPE;  
  
TYPE IfcAppliedValueSelect = SELECT  
  (IfcRatioMeasure,  
   IfcMonetaryMeasure,  
   fp_measure_with_unit);  
END_TYPE;  
  
TYPE IfcObjectTypeEnum = ENUMERATION OF  
  (PRODUCT,  
   PROCESS,  
   CONTROL,  
   RESOURCE,  
   ACTOR,  
   GROUP,  
   PROJECT,  
   NOTDEFINED);  
END_TYPE;  
  
TYPE IfcDayInMonthNumber = INTEGER;  
END_TYPE;  
  
TYPE IfcMonthInYearNumber = INTEGER;  
  WHERE  
    WR1 : { 1<= SELF <= 12 };  
END_TYPE;  
  
TYPE IfcYearNumber = INTEGER;
```

```
END_TYPE;

ENTITY IfcCostValue
  SUBTYPE OF(IfcAppliedValue);
  CostType : IfcLabel;
  Condition : OPTIONAL IfcText;
END_ENTITY;

ENTITY IfcAppliedValue
  ABSTRACT SUPERTYPE;
  Name : OPTIONAL IfcLabel;
  Description : OPTIONAL IfcText;
  AppliedValue : OPTIONAL IfcAppliedValueSelect;
  UnitBasis : fp_measure_with_unit;
  ApplicableDate : fp_select_date_time;
  FixedUntilDate : fp_select_date_time;
  WHERE
    WR1 : EXISTS (AppliedValue) OR
          EXISTS (ValueOfComponents);
END_ENTITY;

ENTITY IfcObject
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcRoot);
  ObjectType : OPTIONAL IfcLabel;
  INVERSE
  HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
  WHERE
    WR1 : SIZEOF(QUERY(temp <* IsDefinedBy | 'IFC2X2_FINAL.IFCRELDEFINESBYTYPE' IN TYPEOF(temp))) <= 1;
END_ENTITY;

ENTITY IfcRoot
  ABSTRACT SUPERTYPE OF (ONEOF(IfcObject, IfcRelationship));
  GlobalId : IfcGloballyUniqueId;
  Name : OPTIONAL IfcLabel;
  Description : OPTIONAL IfcText;
  OwnerHistory : fp_apply_owner_history;
  UNIQUE
  UR1 : GlobalId;
END_ENTITY;

ENTITY IfcRelationship
  SUBTYPE OF(IfcRoot);
END_ENTITY;

ENTITY IfcRelAssigns
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcRelationship);
  RelatedObjects : SET [1:?] OF IfcObject;
  RelatedObjectsType : OPTIONAL IfcObjectTypeEnum;
  WHERE
    WR1 : IfcCorrectObjectAssignment(RelatedObjectsType, RelatedObjects);
```

```
END_ENTITY;

ENTITY IfcRelAssignsToGroup
  SUBTYPE OF(IfcRelAssigns);
  RelatingGroup : IfcGroup;
  WHERE
    WR1 : SIZEOF(QUERY(Temp <* SELF\IfcRelAssigns.RelatedObjects | RelatingGroup :=: Temp)) = 0;
END_ENTITY;

ENTITY IfcGroup
  SUBTYPE OF(IfcObject);
  INVERSE
    IsGroupedBy : IfcRelAssignsToGroup FOR RelatingGroup;
END_ENTITY;

ENTITY IfcInventory
  SUBTYPE OF(IfcGroup);
  InventoryType      : IfcInventoryTypeEnum;
  CurrentValue       : OPTIONAL IfcCostValue;
  OriginalValue      : OPTIONAL IfcCostValue;
  Jurisdiction       : fp_select_actor;
  ResponsiblePersons : SET [1:?] OF fp_select_actor;
  LastUpdateDate     : IfcCalendarDate;
  WHERE
    WR41 : SIZEOF(QUERY(temp <* SELF\IfcGroup.IsGroupedBy.RelatedObjects |
      NOT( ('IFC2X2_FINAL.IFCSPACE' IN TYPEOF (temp)) OR
        ('IFC2X2_FINAL.IFCASSET' IN TYPEOF (temp)) OR
        ('IFC2X2_FINAL.IFCFURNISHINGELEMENT' IN TYPEOF (temp))
      ))) = 0;
END_ENTITY;

ENTITY IfcCalendarDate;
  DayComponent      : IfcDayInMonthNumber;
  MonthComponent    : IfcMonthInYearNumber;
  YearComponent     : IfcYearNumber;
  WHERE
    WR21 : IfcValidCalendarDate (SELF);
END_ENTITY;

ENTITY fp_apply_owner_history;
END_ENTITY;

ENTITY fp_associate_cost;
END_ENTITY;

ENTITY fp_measure_with_unit;
END_ENTITY;

ENTITY fp_model_asset;
END_ENTITY;

ENTITY fp_model_furniture;
```

```
END_ENTITY;

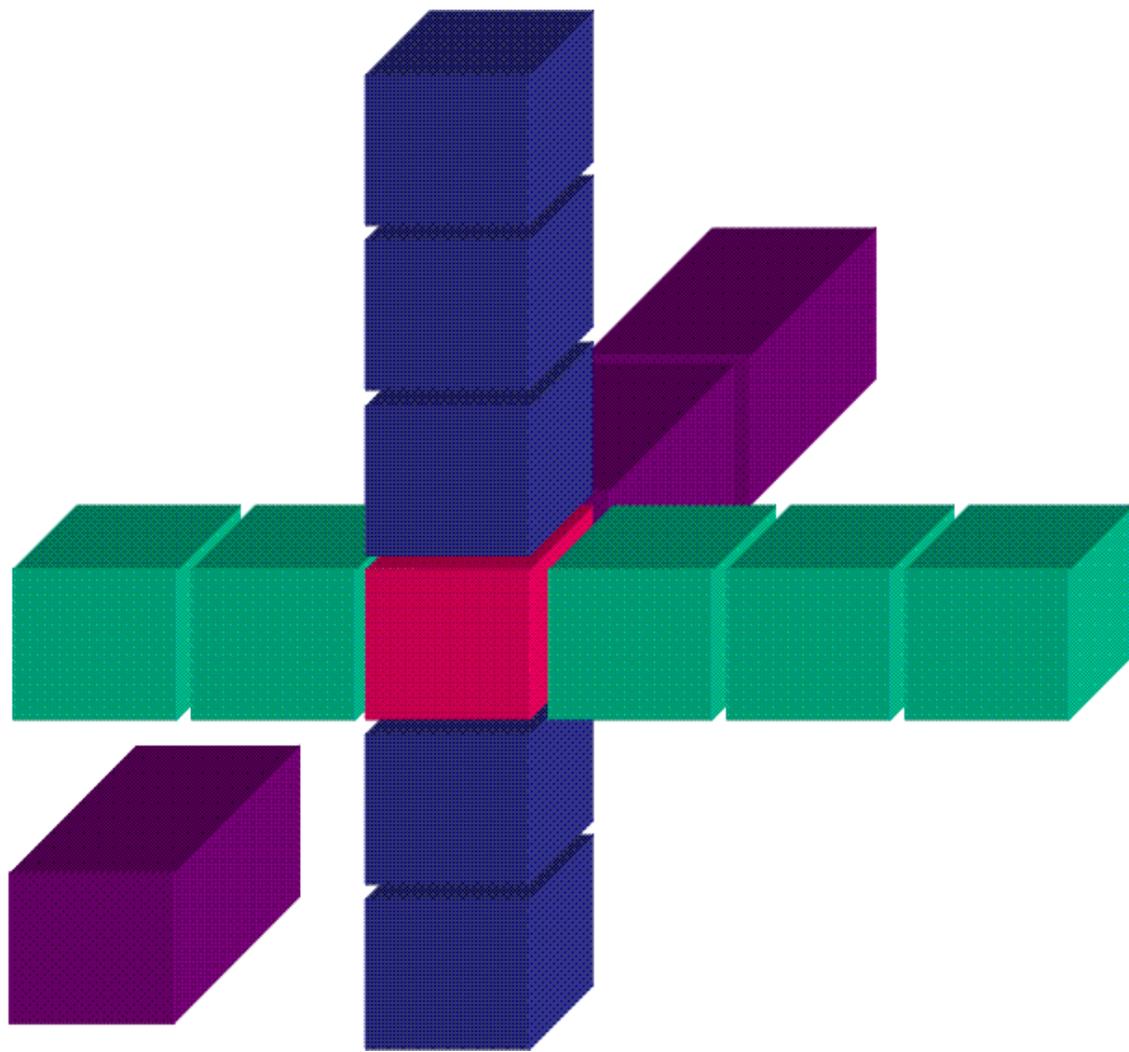
ENTITY fp_model_space;
END_ENTITY;

ENTITY fp_select_actor;
END_ENTITY;

ENTITY fp_select_date_time;
END_ENTITY;

FUNCTION IfcValidCalendarDate
(Date : IfcCalendarDate)
  : LOGICAL;
IF NOT ({1 <= Date.DayComponent <= 31}) THEN
  RETURN(FALSE);
END_IF;
CASE Date.MonthComponent OF
  4      : RETURN({ 1<= Date.DayComponent <= 30});
  6      : RETURN({ 1<= Date.DayComponent <= 30});
  9      : RETURN({ 1<= Date.DayComponent <= 30});
  11     : RETURN({ 1<= Date.DayComponent <= 30});
  2      :
BEGIN
  IF (IfcLeapYear(Date.YearComponent)) THEN
    RETURN({ 1<= Date.DayComponent <= 29});
  ELSE
    RETURN({ 1<= Date.DayComponent <= 28});
  END_IF;
END;
OTHERWISE : RETURN(TRUE);
END_CASE;
END_FUNCTION;

END_SCHEMA;
```



**Information Delivery Manual (IDM)**

## Nests (FP)

Added by [Janice Wix](#), last edited by [Janice Wix](#) on Feb 10, 2007

Labels: (None)

## Nests

## Overview

Describes a whole/part relationship in which the parent object (the whole) is aggregated from a set of child objects (the parts) by nesting. Nesting is a particular form of aggregation in which the parent object is formed by aggregating a set of child objects of the same type (class)

### Concept: Nest objects

The concept of nesting objects is used in a number of situations within the IFC model, allowing objects to be broken down into their constituent parts whilst retaining information about both the whole and the individual parts.

In a nest, the whole is referred to as the 'relating object' whilst the parts are referred to as the 'related objects'. There is always exactly one relating object and there may be one or many related objects.

The nesting relationship is described at a high level in the IFC model as between occurrences of IfcObject. Practically, the relationship will be between instances of a subtype of IfcObject and these inherit the relationship.

To ensure that there is clarity about entities that are participating in a nest, reference to this functional part within the IDM will indicate parameters for the aggregation as:

fp\_nests[object]

Results

Declares the existence of a specific nesting relationship

Description	Entity/Pset/Functional Part	MAN	REC	OPT
<b>Assert general attributes of the relationship</b>				
Set the global unique identifier	IfcRelNests.GlobalId :: IfcGloballyUniqueId	✓		
Apply the owner history	IfcRelNests.OwnerHistory :: fp_apply_owner_history	✓		
<b>Establish the relationship entity</b>				
Set the occurrence of the parent entity in the relationship	IfcObject <parent subtype>	✓		
Set the occurrences of the child entities in the relationship	IfcObject <child subtypes>	✓		
Assert the parent relationship				
<b>IFC Entities Required</b>				
<ul style="list-style-type: none"> <li>IfcObject</li> <li>IfcRelationship</li> </ul> <p>Refer to functional part or exchange requirement for specification of the parent or relating entity to be used.</p> <ul style="list-style-type: none"> <li>IfcRelDecomposes</li> <li>IfcRelNests</li> </ul>	IfcRelNests.RelatingObject :: IfcObject <parent>	✓		

- IfcRoot

## IFC Datatypes Required

---

- IfcGloballyUniqueId
- IfcLabel
- IfcText

## IFC Functions Required

---

- -

## IFC Property Sets Required

---

- -

## IDM Functional Parts Required

---

- fp\_apply\_owner\_history

## EXPRESS Schema

---

```
SCHEMA FP_NESTS;  
  
  TYPE IfcGloballyUniqueId = STRING (22) FIXED;  
  END_TYPE;  
  
  TYPE IfcLabel = STRING;  
  END_TYPE;  
  
  TYPE IfcText = STRING;  
  END_TYPE;  
  
  ENTITY IfcObject  
    ABSTRACT SUPERTYPE  
    SUBTYPE OF(IfcRoot);  
    ObjectType      : OPTIONAL IfcLabel;  
    INVERSE  
    Decomposes      : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;  
    IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;  
    WHERE  
    WR1 : SIZEOF(QUERY(temp <* IsDefinedBy | 'IFC2X2_FINAL.IFCRELDEFINESBYTYPE' IN TYPEOF(temp))) <= 1;  
  END_ENTITY;  
  
  ENTITY IfcRoot  
    ABSTRACT SUPERTYPE OF (ONEOF(IfcObject, IfcRelationship));
```

```

    GlobalId      : IfcGloballyUniqueId;
    Name         : OPTIONAL IfcLabel;
    Description  : OPTIONAL IfcText;
    OwnerHistory : fp_apply_owner_history;
  UNIQUE
    URL : GlobalId;
  END_ENTITY;

  ENTITY IfcRelationship
    ABSTRACT SUPERTYPE
    SUBTYPE OF(IfcRoot);
  END_ENTITY;

  ENTITY IfcRelDecomposes
    ABSTRACT SUPERTYPE
    SUBTYPE OF(IfcRelationship);
    RelatingObject : IfcObject;
    RelatedObjects : SET [1:?] OF IfcObject;
  WHERE
    WR1 : SIZEOF(QUERY(Temp <* RelatedObjects | RelatingObject ::= Temp)) = 0;
  END_ENTITY;

  ENTITY IfcRelNests
    SUBTYPE OF(IfcRelDecomposes);
  WHERE
    WR1 : SIZEOF(QUERY(Temp <* SELF\IfcRelDecomposes.RelatedObjects |
      NOT(TYPEOF(SELF\IfcRelDecomposes.RelatingObject) = TYPEOF(Temp)))) = 0;
  END_ENTITY;

  ENTITY fp_apply_owner_history;
  END_ENTITY;

  END_SCHEMA;

```

## Examples of Nesting Objects

### Example 1: Nesting Compartmentalized Tanks

Tanks may be provided so that fluid storage is in one compartment (the whole tank) or in multiple, separated compartments. The latter will be particularly the case with large tanks. A compartment is an IfcFlowStorageDevice specified as a partial element through the occurrence level property set Pset\_FlowStorageDeviceTank by setting the TankComposition value to PARTIAL

A tank is an IfcFlowStorageDevice specified as a complete element through the occurrence level property set Pset\_FlowStorageDeviceTank by setting the TankComposition value to ELEMENT

Multiple tanks acting together through interconnection are an instance of IfcTankType specified as a complex element through the occurrence level property set Pset\_FlowStorageDeviceTank by setting the TankComposition value to COMPLEX

IfcRelNests is used to connect compartments to tanks (since both are instances of IfcTank)

```

/* TYPE LEVEL PROPERTY SETS AT #100, #200 NOT SHOWN IN THIS EXAMPLE */
/* REPRESENTATION MAP AT #1011 NOT SHOWN IN THIS EXAMPLE */

```

```

/* definition of individual compartments by tank type -no geometry shown- */
#1=IFCTANKTYPE('abcdefghijklmnopqrst11', #2, 'Tank_Part', $, 'IfcFlowStorageDevice', (#100,#200), (#1011), $, $, .PREFORMED.);
#2=IFCOWNERHISTORY(...);

/* relating the occurrences to the type */
#10=IFCRELDEFINESBYTYPE('abcdefghijklmnopqrst10', #2, $, $, (#20,#21,#22,#23,#24,#25,#26,#27), #1);

/* definition of the occurrences - placement references given but no geometry shown- */
#20=IFCFLOWSTORAGEDEVICE('abcdefghijklmnopqrst20',#2,$,$,$,#10020,#10002,$);
#21=IFCFLOWSTORAGEDEVICE('abcdefghijklmnopqrst21',#2,$,$,$,#10021,#10002,$);
#22=IFCFLOWSTORAGEDEVICE('abcdefghijklmnopqrst22',#2,$,$,$,#10022,#10002,$);
#23=IFCFLOWSTORAGEDEVICE('abcdefghijklmnopqrst23',#2,$,$,$,#10023,#10002,$);
#24=IFCFLOWSTORAGEDEVICE('abcdefghijklmnopqrst24',#2,$,$,$,#10024,#10002,$);
#25=IFCFLOWSTORAGEDEVICE('abcdefghijklmnopqrst25',#2,$,$,$,#10025,#10002,$);
#26=IFCFLOWSTORAGEDEVICE('abcdefghijklmnopqrst26',#2,$,$,$,#10026,#10002,$);
#27=IFCFLOWSTORAGEDEVICE('abcdefghijklmnopqrst27',#2,$,$,$,#10027,#10002,$);
#28=IFCFLOWSTORAGEDEVICE('abcdefghijklmnopqrst28',#2,$,$,$,#10028,#10003,$);

/* placing the partial tanks into the elemental tank */
#30=IFCRELNESTS('abcdefghijklmnopqrst30', #2, $, 'Compartmentalized tank nesting', #28, (#20,#21,#22,#23,#24,#25,#26,#27));

/* relating a property set to the occurrences */
#50=IFCRELDEFINESBYPROPERTIES('abcdefghijklmnopqrst50', #2, $, $, (#20,#21,#22,#23,#24,#25,#26,#27), #300);
#51=IFCRELDEFINESBYPROPERTIES('abcdefghijklmnopqrst51', #2, $, $, (#28), #301);

/* definition of occurrence property sets */
#300=IFCPROPERTYSET('abcdefghijklmnopqrst300', #2, 'Pset_FlowStorageDeviceTank', $, (#310,#321,#322));
#301=IFCPROPERTYSET('abcdefghijklmnopqrst301', #2, 'Pset_FlowStorageDeviceTank', $, (#311,#321,#322));

/* definition of specific properties for a preformed tank */
#310=IFCPROPERTYENUMERATEDVALUE('TankComposition', $, ('PARTIAL'), #351);
#311=IFCPROPERTYENUMERATEDVALUE('TankComposition', $, ('ELEMENT'), #351);
#321=IFCPROPERTYSINGLEVALUE('HasLadder', $, IFCBOOLEAN(.F.), $);
#322=IFCPROPERTYSINGLEVALUE('HasVisualIndicator', $, IFCBOOLEAN(.F.), $);

/* definition of property enumerations used in specific type property set */
#351=IFCPROPERTYENUMERATION('PEnum_TankComposition', ('COMPLEX', 'ELEMENT', 'PARTIAL', 'NOTKNOWN', 'UNSET'), $);

```

## Example 2: Nested Cost Schedules

In this example, the estimate for hot water services (described in the examples for `fp_model_cost_schedule` including instances of organization, person and date) is one cost schedule in a set that is combined to produce a total estimate for all services on a proposal that is name 'New Munkerud'. To obtain the complete estimate, the domestic cold water services and the waste water estimates are also included.

```

/* ownership information*/
#2=IFCOWNERHISTORY(...);

/* define the cost schedule */
#2000=IFCCOSTSCHEDULE('abcdefghijklmnopqrst2000',#2,'New Mumkerud','All Services',$,#3001,#3002,#4001,'ISSUED',$,#4001,'S20050630-0',.ESTIMATE.);
#2001=IFCCOSTSCHEDULE('abcdefghijklmnopqrst2001',#2,'Domestic Hot Water','Pipework only',$,#3001,#3002,#4001,'ISSUED',$,#4001,'S20050630-1',.ESTIMATE.);
#2002=IFCCOSTSCHEDULE('abcdefghijklmnopqrst2002',#2,'Domestic Cold Water','Pipework only',$,#3001,#3002,#4001,'ISSUED',$,#4001,'S20050630-2',.ESTIMATE.);

```

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 2.1.3 Build: #408 Jan 23, 2006) - [Bug/feature request](#) - [Contact Administrators](#)