The Open BIM Object Standard

(The OBOS)

Introduction

Market feedback in Australia and New Zealand that the lack of standards limited the sharing of BIM content led NATSPEC and Masterspec to initiate the development of an open BIM object standard. Discussions with sister organisations around the world identified similar needs in other countries.

Most jurisdictions have, or are moving to, protocols or standards for BIM processes. There are also a variety of BIM standards that address some aspects of BIM content or tools but many other aspects have no guidance at all. Therefore, it is possible to create widely varying content that comply with these standards but are very different in implementation.

As a result, many practices develop their own internal standards and are often reluctant to use any content produced by third-parties.

A number of object libraries and **BIM authoring system**-based groups have developed their own standards to address this, but the proprietary nature of these standards mean that it is difficult for a jurisdiction to adopt these as a requirement.

The intent of this standard is to cover the primary aspects of BIM object development to provide sufficient commonality, that practitioners will accept and use content developed to the requirements of this standard.

This standard is to be an open standard, jointly and severally owned by all participants. Local requirements, variations or additions are to be dealt with in annexures to the standard.

The initial version of this standard was developed by NATSPEC, Australia, and Masterspec, New Zealand, and is made freely available to any organisation wishing to use and adopt it in their jurisdiction or join in its future development.

Document Conventions

Requirements of this standard are expressed in sentences which include the word ‘shall’. To say that an object complies with or conforms to this standard, all the ‘shall’ requirements of this standard need to have been applied to the BIM object.

Any recommendations of this standard are expressed in sentences which include the word ‘should’. Whilst not a requirement of this standard, it is recommended that an object include these items.

The standard uses the word ‘can’ to nominate that a concept is a possibility, for example, an addition to the main concept. The object author can include these items, if required.

Terms are identified within the standard using **bold** formatted text for the first time they appear in each clause, with their definition provided in Section 8 - *Definitions and Glossary*.

1.0 BIM Objects

This section of the standard describes what an object is, and the basic object types.

1.1 BIM components and assemblies

This sub-section of the standard discusses the concept of individual BIM objects and BIM object assemblies. Defining the broad requirements for each.

1.1.1 BIM objects

BIM objects shall be modelled as a component object or a system object.

- Component objects are discrete, or loadable, objects where the geometry is created by the object modeller and represents a specific physical element or component.
- System objects are objects where the geometry is provided by the **BIM authoring system** for the user to configure. Typically, these are **layered objects** such as walls, floors, roofs and ceilings, but can also include, in some BIM authoring systems, windows, doors, piping, columns, beams and similar elements.

1.1.2 BIM assemblies

BIM objects can be grouped into **assemblies** to allow for reuse of common groupings of physical elements.

1.2 Proprietary and generic BIM objects

This sub-section of the standard discusses the differences between proprietary (manufacturer) BIM objects and generic BIM objects.

1.2.1 Object type

The object shall be modelled as a **generic object** or a **proprietary object**.

1.2.2 Generic objects

Generic objects shall be, as a minimum, recognisable as a representation of the physical element that it is representing.

1.2.3 Proprietary objects

Proprietary objects shall be an appropriate geometric representation of the physical element with accurate critical dimensions and **metadata**.

1.2.4 Project objects

Project objects are unique project-specific objects created as either a generic or proprietary object for use in a project and not intended for inclusion in a shared **library**.

2.0 Naming Conventions

This section of the standard sets out the requirements and rules for naming object files, properties, property sets, materials, views and material image files.

2.1 General naming requirements

This sub-section of the standard defines the general naming requirements applicable to all situations covered by this standard.

2.1.1 Naming fields

Object files, materials and material image files shall be named using the naming **field**s from Table 2A, arranged in accordance with the relevant naming clauses below.

Field	Description	Example
<Type>	Shall be used to identify the type of object or material.	Door
<Subtype>	Shall be used to identify the subtype of the object or material.	Interior
<Source>	The product manufacturer identification. Shall be used for proprietary objects.	SupaDoors
<Product/Range identifier>	The manufacturer's product or product range identifier. Shall be used for proprietary objects.	D130ST
<Differentiator>	Can be used to provide additional information required to identify the object or material.	760W
<Originator>	A 3 to 6 character code, can be used to identify the object provider. Should be used for objects provided by an object **library**.	ABC

Table 2A – Naming fields

2.1.2 Characters

Names and naming **field**s shall include only the following characters:

- Uppercase letters (A to Z) from the ISO basic Latin alphabet.
- Lowercase letters (a to z) from the ISO basic Latin alphabet.
- Numbers (0 to 9).
- Underscore (_), for separating fields within a name.
- A single period character (.), used only to separate the file name from the file extension.

Names and naming fields shall not include any of the following characters:

- Symbols or mathematical operators, including, but not limited to, (! " , £ $ % ^ & * { }[ ] + - = < > ? | \ / @ ' ~ # ¬ ` ').
- Spaces.

The use of the hyphen character should be avoided as it can cause errors when the name (or **property**) is used within formulae, due to the hyphen also representing the mathematical subtraction operator.

2.1.3 Naming rules

PascalCase shall be used to join separate words within naming **field**s and for the naming of properties.

Naming fields shall be separated by an underscore (_).

Where required, abbreviations can be used. Recognised industry abbreviations shall be used where they exist, for example, PVC.

2.2 Object file naming requirements

This sub-section of the standard defines the requirements for object file naming. Including Proprietary/Generic/Project object naming, Parametric object naming and Dimensional information in file names.

2.2.1 File naming rules

The object file name shall be a unique human readable description of the object.

The object file name shall be in accordance with clause 2.1.

Relevant dimensional information can be included in the <Differentiator> **field** and should include the unit of measure. For example, 300x700mm, 400mmx30m, 1200mm.

2.2.2 Parametric object file naming

The subtype **field** of a **parametric** object file name shall be a pascal case concatenation of the variants represented by the object. For example, RectangularSquare.

2.2.3 File name structure

The object file name shall be structured as follows, unless specified otherwise by local requirements:

<Type>_<Subtype>_< Source>_< Product/Range identifier>_< Differentiator>_< Originator>

2.3 Property naming requirements

This sub-section of the standard defines the requirements for the naming of properties (also known as attributes or parameters) associated with BIM objects. BIM properties are variables that store data relating to the object. The naming of properties needs to provide a human readable description of the property requirement and also adopt the conventions and methodologies used within modern programming language techniques to allow automated use of the data contained within.

2.3.1 Unique properties

Each unique concept describing information about an object shall have a unique **property** name.

2.3.2 Property naming rules

Properties shall be named in accordance with clause 2.1.

Properties shall be named in a consistent and logical manner to aid clarity and usability.

Each **property** name shall be a unique human readable description of the property value requirement.

2.3.3 Boolean properties

Properties that require the value to be completed with a Yes/No or True/False enumerator, shall be named to indicate such a requirement, for example, 'IsExternal'.

2.3.4 Suffix

If user edited/created **property set**s are not supported in the **BIM authoring system** being used (see clause 4.10), a 3 to 6 character suffix can be added to the end of each **property**

name, separated by an underscore(_), to identify the source of the property. For example, DoorPanelHeight_ANZRS.

Note: A suffix cannot be added to the end of property names from defined data schemas such as **IFC** (Industry Foundation Classes) or **COBie**. The property names from any defined data schema must not be changed or amended in any way.

2.4 Property set naming requirements

This sub-section of the standard defines requirements for the naming of property sets associated with BIM objects. This is to facilitate the metadata being in a format consistent with IFC. Not all BIM authoring systems fully support the concept of property sets for BIM objects, for example Revit only allows the user to assign object properties to hard-coded groups, however the shared parameter file does assign each property to a group, which is an equivalent concept to property sets.

2.4.1 Property sets

Object properties shall be grouped into logical sets, see clause 4.10.

2.4.2 Property set naming rules

Property sets shall be named in accordance with clause 2.1.

Property set names shall provide a descriptive context for the grouping.

Property set names shall have a prefix followed by an underscore to identify the origin of the property set, for example, *OBOS_*, *COBie_*, *ANZRS_*.

IFC property sets shall be named with the corresponding IFC property set name (*Pset_*).

For properties identified in this standard, the property set name shall be as shown in italics in the table title. For example, for the properties in **Table 4A – *OBOS_Admin* properties** the property set name is shall be *OBOS_Admin*.

2.5 Material naming requirements

This sub-section of the standard defines the requirements for the naming of materials associated with BIM objects.

2.5.1 Material naming rules

Materials shall be named in accordance with clause 2.1.

Material names shall be unique human readable descriptions of the material.

2.5.2 Material name structure

The material name shall be structured as follows:

<Type>_<Subtype>_<Differentiator>

2.6 View control naming requirements

This sub-section of the standard defines the requirements for the naming of view controls associated with BIM objects. BIM authoring systems control object viewing in different ways,

such as layers and subcategories. Where the views can be named by the user, and are shared with objects, then it is important to name them consistently to avoid a proliferation.

2.6.1 View control naming rules

Where the **BIM authoring system** allows for user defined **view control**, views shall be named as follows:

- In accordance with clause 2.1.
- In plural form.
- Be consistent and logical to aid clarity and usability.

2.6.2 View control name structure

The **view control** name shall be structured as follows, using the **field**s from Table 2B:

<UsageGroup>_<UsageSubGroup>_<Differentiator>

Field	Description
<UsageGroup>	Shall be used to name the group's primary purpose or use, for example, *Plumbing*.
<UsageSubGroup>	Should be used to name the sub group's purpose or use if the usage group covers multiple concepts, for example, *Plumbing_Sanitary*.
<Differentiator>	Can be used to further differentiate the sub group, for example, *Plumbing_Sanitary_Fittings*.

Table 2B – View control naming fields

2.7 Material image file naming requirements

This sub-section of the standard defines the requirements for the naming of image files associated with materials.

2.7.1 Material image file naming rules

Image files shall be named in accordance with clause 2.1.

2.7.2 Material image file name structure

Image files shall be named identically to their corresponding material (see clause 2.5), with the addition of the file format extension of either .bmp or .jpg.

3.0 Object Classification

This section of the standard sets out the requirements for classifying the object and assigning it the correct IFC designation.

3.1 Object IFC designation

This sub-section of the standard defines the requirements for the designation of the object in IFC to aid interoperability of the BIM object between different BIM applications.

3.1.1 IFC type

The BIM object shall be designated the appropriate *IfcElementType* and *PredefinedType*, see clause 4.3.2.

3.2 Object classification

This sub-section of the standard defines the requirements for the primary classification of the BIM object. So that all objects are classified to one common system, Uniclass 2015 has been chosen as the primary classification system. Due to the proliferation of systems in use, it is likely that an object may need to support multiple classifications (addressed in Section 4 of this standard).

3.2.1 Uniclass 2015

BIM objects shall have a Uniclass 2015 classification assigned, see clause 4.6.1. The classification code should be as specific as possible.

3.2.2 Hard-coded classification

BIM objects can be assigned an assembly or costing code and keynote code.

4.0 Object Properties & Property Grouping

This section of the standard defines the properties that should be assigned to a BIM object. It also discusses the grouping of properties and the assignment of properties to object types or object instances.

4.1 General object property requirements

This sub-section of the standard defines the general rules for properties associated with BIM objects including the requirements for using non-conforming hard-coded properties within BIM authoring applications.

4.1.1 Provision of properties

Properties can either be embedded within the object or linked via a unique link to an external database.

4.1.2 Extent of properties

Only properties that are deemed necessary to adequately define the object for its intended BIM use(s), and to provide for object interoperability between BIM applications, shall be provided.

4.1.3 Duplication of properties

An object shall not include duplicate occurrences of a **property**.

Where an identical property, either in name or in description/value requirement, exists in multiple sources (for example, **IFC** and **COBie**), only include a single occurrence of that property.

IFC properties take precedence over properties from other sources.

4.1.4 Hard-coded properties

Hard-coded properties within the **BIM authoring system** shall be used.

Where necessary, hard-coded properties shall be mapped to their corresponding **IFC** properties at export, see clause 7.2.

4.1.5 Proprietary and generic object properties

Proprietary objects shall, as a minimum, include the same properties as those provided for a generic object of the same object type and subtype.

4.1.6 Property values

Property values shall be completed where known.

Proprietary objects shall have all Type property values completed, see clause 4.2.1.

Property values can be completed with:

- A fixed value, where only one value is available or a decision has been made on which value is required, for example, 6000.
- A value range, where a bounded range of values are available and a decision has not been made on which value is required. The lower and upper bound of the available range shall be separated by a hyphen, for example, 100 – 600.
- An enumerated value, where a number of values are available from a defined list of enumerators and a decision has not been made on which value is required. Each available enumerator shall be separated by a comma, for example, X, Y, Z.
- A formula, if the value is reliant on the value of another property.

Property values shall not:

- Finish with a full stop(.).
- Include units, unless specifically required to do so in the property description.

4.1.7 Property units

Units shall be metric.

4.1.8 Property data type

Properties shall have their data type defined using **ANSI SQL** data types, selected from the following list:

- Character (Text, String) – for fields requiring alpha numeric or text values.
- Integer – for fields requiring whole numbers.
- Decimal (Number, Currency) – for fields requiring decimal numbers.
- Date – for fields requiring a date in the format year-month-day (YYYY-MM-DD).
- Boolean (Yes/No, True/False) – for fields requiring a choice from two options.
- Hyperlink (URL) – a character field where values are links represented as www.website.domain/path.

If a data type listed above is not available, or a more specific **hard-coded** data type is available, within the **BIM authoring system** being used, use the most appropriate hard-coded data type. For example, if there is no Date data type, use a Character type and enter the date in the correct format.

4.1.9 Assembly properties

Assemblies shall not have properties assigned to the assembly itself.

All properties shall be assigned to the objects contained within the assembly.

4.2 Object type or instance properties

This sub-section of the standard discusses the concept of properties being type properties or instance properties.

Type objects represent defined specifications of physical elements. The type information contained on the object is common for all instances of the object.

An Instance of an object is a unique placement of a type object in a model. An Instance object can contain unique information such as serial numbers and installation or commissioning information.

4.2.1 Type properties

A **property** that has a consistent value for all instances of that object type shall be treated as a Type property. Values for Type properties of **library** objects can be pre-completed.

4.2.2 Instance properties

A **property** that requires a value that is specific to the instance of that object type shall be treated as an Instance property. Values for Instance properties of **library** objects cannot be pre-completed.

4.3 IFC properties

This sub-section of the standard defines the IFC properties to be assigned to a BIM object.

4.3.1 IFC for interoperability

To provide for greater interoperability of objects and models between different BIM applications, all objects shall include properties from the buildingSMART IFC4 (Addendum 2) schema that directly correspond to that object type.

4.3.2 IFC object designation properties

The object shall include a **property** named '*IfcExportAs'*, with the value completed with the relevant *IfcElementType* from IFC4 (Add 2). For example, 'IfcPile', for a pile foundation.

The object shall include a property named '*IfcExportType*', with the value completed with the relevant *PredefinedType*, selected from the enumerated list available for the nominated *IfcElementType*. For example, 'DRIVEN', for a driven pile foundation.

Note: BIM authoring applications may automatically assign the '*IfcExportAs'* and '*IfcExportType'* properties, based on the in-built tools or templates used for modelling the object, see clause 5.2.1.

4.3.3 IFC proxy object designation

If a relevant *IfcElementType* does not exist in IFC4 (Add2), for the object being modelled, then the '*IfcExportAs*' **property** shall be completed with 'IfcBuildingElementProxy' and the '*IfcExportType*' property shall be completed with 'USERDEFINED'.

When the *IfcBuildingElementProxy* has been nominated, the object shall include an additional property of '*ElementType*', with the value completed with a descriptive name to define the object type.

4.3.4 IFC common properties

If an **IFC** common **property set** (*Pset_XXXCommon*) exists for the object type, the properties from that property set shall be included.

If no IFC common property set exists for the object type, the properties from the IFC *Pset_BuildingElementProxyCommon* property set shall be included.

4.3.5 Additional IFC properties

Additional properties and **property set**s from the **IFC** schema can be included, such as Quantity sets (*Qto_*) and environmental impact properties.

If a **property** in the IFC4 (Add2) schema addresses the same concept to that of a property proposed for inclusion, the IFC property shall be used.

4.4 Administration properties

This sub-section of the standard defines properties to be assigned to a BIM object that relate to the administration and management of the object.

4.4.1 Object admin properties

The set of properties in Table 4A shall be included for all objects.

Property	Description	Data type	Example	Type / Instance
CreatedBy	Name of the person, organisation or **library** provider that created the object.	Character	SupaObjects	Type
CreatedByURL	URL hyperlink to the object creator's website.	Hyperlink	www.SupaObjects.com	Type
ModifiedIssue	To record the last date of issue (version or revision) of the object within an object **library**. Value to be completed in the following format <yyyy-mm-dd.no>. The 'no' suffix can be used if the object is issued multiple times on that date.	Date	2018-03-16.01	Type

Table 4A – *OBOS_Admin* properties

4.5 Proprietary object properties

This sub-section of the standard defines properties to be assigned to a proprietary object.

4.5.1 Manufacturer/Product identification properties

The sets of properties in Tables 4B and 4C shall be included for all objects that represent a proprietary manufacturer product (or may be used to do so in the future).

Property	Description	Data type	Example	Type / Instance
Manufacturer	Organization that manufactured and/or assembled the product.	Character	SupaDoors	Type
ModelLabel	Descriptive model name of the product model (or	Character	Solid Timber Door	Type

	product line) as assigned by the manufacturer.			
ModelReference	Model number or designator of the product model (or product line) as assigned by the manufacturer.	Character	D130ST	Type
ArticleNumber	Article number or reference that is be applied to a configured product according to a standard scheme for article number definition as defined by the manufacturer. It is often used as the purchasing number.	Character	D130ST2100X800	Type
GlobalTradeItemNumber	Global Trade Item Number (GTIN) is an identifier for trade items developed by GS1 (www.gs1.org).	Integer	00012345600012	Type
ProductionYear	Year of production of the manufactured product.	Integer	2018	Type or Instance
AssemblyPlace	Enumeration defining where the assembly is intended to take place, either in a factory or on the building site.	Character	FACTORY, OFFSITE, SITE, OTHER, NOT KNOWN, UNSET*	Type or Instance

* Permissible enumerators

Table 4B – *Pset_ManufacturerTypeInformation* properties

Property	Description	Data type	Example	Type / Instance
ManufacturerURL	URL hyperlink to the manufacturer's website.	Hyperlink	www.SupaDoors.com	Type
ProductURL	URL hyperlink to further product information, such as technical documentation or installation guides.	Hyperlink	http://www.SupaDoors.com/D130ST	Type

Table 4C – *OBOS_Manufacturer* properties

4.6 Classification properties

This sub-section of the standard defines properties to be assigned to an object to classify the object type.

4.6.1 Uniclass classification

To provide objects classified to at least one system (Uniclass2015), the set of properties in Table 4D shall be included.

NOTE: Substitute <Table> in the **property** name for the required Uniclass table name. For example, Products, Systems or Elements. Examples in Table 4D have used the Products table.

If a suitable classification does not exist in the Uniclass 2015 Products table, a suitable classification from the Elements or Systems table can be used.

Property	Description	Data type	Example	Type / Instance
Uniclass2015<Table>Code	Required classification code.	Character	Pr_30_59_23	Type
Uniclass2015<Table>Title	Required classification title.	Character	Doors frames and leaves	Type
Uniclass2015<Table>Version	Published version of the Uniclass 2015 table.	Decimal	1.10	Type

Table 4D – *OBOS_Classification* properties

4.6.2 Additional classifications

Multiple classification systems can be assigned to an object by including the set of properties in Table 4D for each classification required.

NOTE: Substitute <ClassificationSystemName> in the property name for the required classification system and table designation, if applicable. For example, OmniclassTable23. Examples in Table 4E have used Omniclass Table 23.

Property	Description	Data type	Example	Type / Instance
<ClassificationSystemName>Code	Required classification code.	Character	23-17-11-00	Type
<ClassificationSystemName>Title	Required classification title.	Character	Doors	Type
<ClassificationSystemName>Version	Published version of the classification system.	Character	2012-05-16	Type

Table 4E – *OBOS_Classification* additional properties

4.7 Construction specification properties

This sub-section of the standard defines properties to be assigned to an object to link an object to relevant construction specification information.

4.7.1 Specification linking properties

To provide for linking the object to construction specification information, the set of properties in Table 4F can be included.

NOTE: Substitute <SpecificationSystem> in the **property** name for the required specification system or project specification.

Property	Description	Data type	Example	Type / Instance

<SpecificationSystem>Code	Specification clause or section code/reference.	Character	0453	Type
<SpecificationSystem>Title	Specification clause or section title.	Character	Doors and access panels	Type
<SpecificationSystem>Version	Published version of the specification system.	Character	Oct-18	Type

Table 4F – *OBOS_Specification* properties

4.8 Asset/Facilities management properties

This sub-section of the standard defines properties to be assigned to an object for ongoing asset or facilities management purposes.

4.8.1 Asset properties

Where the recording of asset data for asset/facilities management purposes is a requirement, the sets of properties in tables 4G, 4H and 4J should be included.

Property	Description	Data type	Example	Type / Instance
AcquisitionDate	Date that the manufactured product was purchased.	Date	2018-03-16	Instance
BarCode	Bar code given to an occurrence of the product.	Integer	9781119060055	Instance
SerialNumber	Serial number assigned to an occurrence of a product.	Character	P345	Instance
BatchReference	Identity of the batch reference from which an occurrence of a product is taken.	Character	P345-20180214	Instance

Table 4G – *Pset_ManufacturerOccurrence* properties

Property	Description	Data type	Example	Type / Instance
WarrantyIdentifier	Identifier assigned to a warranty.	Character	D130ST5135	Type/ Instance
WarrantyStartDate	Date on which the warranty commences.	Date	2018-03-18	Instance
WarrantyEndDate	Date on which the warranty expires.	Date	2020-03-18	Instance
IsExtendedWarranty	Indication of whether this is an extended warranty whose duration is greater than that normally assigned to a product.	Boolean	TRUE or FALSE*	Instance
WarrantyPeriod	Time duration during which a manufacturer or supplier guarantees or warrants the performance of a product. Suggested units = Years.	Integer	2	Type
WarrantyContent	Content of the warranty.	Character	Product and installation	Type/ Instance
PointOfContact	Organization that should be contacted for action under the terms of the warranty.	Character	SupaDoors	Type

Exclusions	Items, conditions or actions that may be excluded from the warranty or that may cause the warranty to become void.	Character	Vandalism	Type/ Instance

* Permissible enumerators

Table 4H – *Pset_Warranty* properties

Property	Description	Data type	Example	Type / Instance
ServiceLifeDuration	Length or duration of a service life. A lower bound indicates pessimistic service life, an upper bound indicates optimistic service life, and a single value indicates the typical service life. Suggested units = Years.	Integer	2	Type
MeanTimeBetweenFailure	Average time duration between instances of failure of a product. Suggested units = Years.	Integer	3	Type

Table 4J – *Pset_ServiceLife* properties

4.8.2 COBie properties

The **COBie** properties in Tables 4K and 4L can be included, if required. If also using properties from Tables 4G, 4H and 4J, do not include any duplicate properties from Tables 4K and 4L.

Property	Description	Data type	Example	Type / Instance
Name	Unique human readable name for object type.	Character	Door	Type
Category	Unique identifier of the object within an external source (classification, document or **library**). It may be human readable (such as a classification code) or not (such as a GUID) depending on the context of its usage (which has to be determined by local agreement).	Character	23-17-11-00	Type
Description	Concise description of the object.	Character	Solid pine timber door	Type
AssetType	Identifies the predefined types of asset from which the type required may be set.	Character	FIXED or MOVABLE*	Type
Manufacturer	Email address of the organization that	Character	info@SupaDoors.com	Type

	manufactured and/or assembled the product.			
ModelNumber	Model number or designator of the product model (or product line) as assigned by the manufacturer.	Character	D130ST	Type
WarrantyGuarantorParts	Email address for organisation responsible for parts warranty.	Character	info@SupaDoors.com	Type
WarrantyDurationParts	Duration of parts warranty.	Integer	3	Type
WarrantyGuarantorLabor	Email address for organisation responsible for labour warranty.	Character	info@SupaDoors.com	Type
WarrantyDurationLabor	Duration of labour warranty.	Integer	3	Type
WarrantyDurationUnit	Units used to record warranty durations. Suggested units = Years.	Character	Years	Type
ReplacementCost	Cost to replace product.	Decimal	1000.00	Type
ExpectedLife	Expected serviceable life of product.	Integer	15	Type
DurationUnit	Units used to record expected life duration. Suggested units = Years.	Character	Years	Type
WarrantyDescription	Concise description of warranty content and any items, conditions or actions that are excluded from warranty or that will cause the warranty to become void.	Character	Product and installation	Type
NominalLength	Typically the primary or larger of the two perpendicular horizontal dimensions of the product.	Integer	800	Type
NominalWidth	Typically the secondary or smaller of the two perpendicular horizontal dimensions of the product.	Integer	40	Type
NominalHeight	Typically the vertical characteristic dimension of the product.	Integer	2100	Type
ModelReference	Descriptive model name of the product model (or product line) as assigned by the manufacturer.	Character	Solid pine timber door	Type
Shape	Characteristic shape of the product.	Character	Rectangular	Type

Size	Characteristic size of the product.	Character	Large	Type
Color	Characteristic colour of the product.	Character	White	Type
Finish	Specification of surface finish.	Character	Gloss paint	Type
Grade	Standard grading(s) to which the product corresponds.	Character	Internal	Type
Material	Characteristic material of which the product is manufactured.	Character	Pine	Type
Constituents	Details of various parts of the product.	Character	Solid panel door	Type
Features	Primary features or important characteristics of the product.	Character	Single leaf	Type
AccessibilityPerformance	Accessibility issue(s) that product satisfies.	Character	Automated	Type
CodePerformance	Code compliance requirement(s) that product satisfies.	Character	Fire rated	Type
SustainabilityPerformance	Sustainability issue(s) that product satisfies.	Character	Recycled timber	Type

* Permissible enumerators

Table 4K – *COBie_Type* properties

Property	Description	Data type	Example	Type / Instance
SerialNumber	Serial number assigned to an occurrence of a product.	Character	D130ST2100X800	Instance
InstallationDate	Date that the manufactured product was installed.	Date	2018-03-18	Instance
WarrantyStartDate	Date on which the warranty commences.	Date	2018-03-18	Instance
TagNumber	Tag identifier (number or label) for the instance of a product or equipment at installation.	Character	D456	Instance
BarCode	Bar code given to the instance of a product.	Integer	9781119060055	Instance
AssetIdentifier	Alternative identifier for specific component, used as defined by contract.	Character	D456-C	Instance

Table 4L – *COBie_Component* properties

4.9 Product performance properties

This sub-section of the standard provides for properties relating to product performance or compliance to assigned to the object.

4.9.1 Product declaration/certification properties

Where a statutory product declaration template (PDT), product technical statement (PTS), appraisal, certification or similar exists for the physical element, the object shall include the set of properties included in Table 4M.

NOTE: Substitute <TechnicalDocument> in the **property** name for the name of the technical document, for example, CodeMark.

Property	Description	Data type	Example	Type / Instance
<TechnicalDocument1>URL	URL hyperlink to technical document.	Hyperlink	www.codemark.com/1234	Type
<TechnicalDocument2>URL	URL hyperlink to technical document.	Hyperlink	www.productdata.com/1234	Type

Table 4M – *OBOS_Technical* properties

4.9.2 Technical performance properties

The object can include properties for items scheduled within the technical documents nominated in Table 4M, if required for the intended BIM use of the object.

4.10 Property grouping (Property sets)

This sub-section of the standard defines the requirements for grouping properties into property sets and the use of property sets. This is to facilitate the metadata being in a format consistent with IFC. Not all BIM authoring systems fully support the concept of property sets in their BIM objects, for example Revit only allows the user to assign object properties to hard-coded groups. However, the shared parameter file in Revit does assign a group to each property, which is an equivalent concept to property sets.

4.10.1 Property grouping requirements

Properties shall be grouped together into **property set**s (groups) for ease of **property** management, navigation and display.

Property sets shall be named in accordance with clause 2.4.

Properties shall be grouped logically and consistently, across all objects, based on their source and intended use/function.

If user edited/created property sets are not supported in the **BIM authoring system** being used, properties shall be assigned to the most relevant **hard-coded** property group within the BIM authoring system.

4.10.2 IFC property sets

IFC properties shall be grouped in their relevant IFC **property set**s (*Pset_*'s and *Qto_*'s).

4.10.3 Source of properties

Properties from any source other than **IFC**, shall be grouped in a **property set**, named to clearly identify the source of the properties. For example, COBie_, ANZRS_.

If user edited/created property sets are not supported in the **BIM authoring system** being used, a suffix can be added to each property name to identify its source, see clause 2.3.4.

5.0 Graphical/Geometric Object Modelling

This section of the standard defines the generic modelling requirements for the graphical element of the BIM object.

5.1 Object graphical detail

This sub-section of the standard discusses the graphical requirements for generic and proprietary objects.

5.1.1 Generic objects

Generic objects shall be visually recognisable as the object type they represent and dimensions shall represent the extent of the object and its connectivity.

Generic objects can include sufficient geometric detail for coordination purposes or to create a visualisation of the object and envisage its operation and use.

5.1.2 Proprietary objects

Proprietary objects shall be visually recognisable as the physical element they represent and critical dimensions shall be accurate for coordination purposes.

Proprietary objects can include sufficient geometric detail to create a visualisation of the object and envisage its operation and use.

5.2 Geometric modelling requirements

This sub-section of the standard discusses the requirements for creating the geometric BIM object, including requirements for the insertion point, units, scale, tools to be used, dimensioning and labelling.

5.2.1 Modelling tools

The object shall be modelled using the corresponding or most appropriate tool/template from within the **BIM authoring system**.

Consideration shall be given to the tool/template selected in regards to the **IFC** designation that will be assigned to the object, see clause 3.1.

5.2.2 Object origin point

The object **origin point** (or base point) shall be established for the BIM object, from which all object geometry will be set out.

Object origin points shall be consistent for identical or similar object types to allow for simple substitution of alternative BIM objects.

The object origin point can correspond to the **insertion point** of the object, see clause 5.2.3.

5.2.3 Insertion points

An **insertion point** shall be established for the BIM object that logically represents the placement of the object in the project model.

Insertion points shall be consistent for identical or similar object types to allow for simple substitution of alternative BIM objects.

The insertion point should be located within the extent of the object.

Particular consideration shall be given to objects with **parametric** geometry to allow for the geometry of the object to change in the model environment and the object's position to remain correct. For example, the insertion point of a column should be on its centreline to allow the column size to be changed, if necessary, without changing the position of the centreline of the column.

NOTE: **BIM authoring systems** usually require structural items to have their insertion point in the centroid of that element to aid any analytical calculations that the model may be used for.

5.2.4 Scale

The object shall be modelled at a scale of 1:1.

5.2.5 Units

Metric geometry in millimetres shall be used.

5.2.6 Fixed geometry

Fixed geometry shall be used where it is not intended for the object to be modifiable. For example, a **proprietary object** that is only available in one size.

5.2.7 Dimensioning

All dimensions shall be constrained to reference items, such as planes, lines or points, rather than directly to geometry.

Dimensions shall be derived automatically using the **associative dimensioning** functions within the **BIM authoring system**.

Dimensions should be located beyond the extents of the object geometry.

Dimensions shall not overlap and shall be clear to read.

5.2.8 Labels

All labels shall be constrained to reference items, such as planes, lines or points, rather than directly to geometry.

Labels that reflect information contained within the object **metadata** shall match such data.

5.2.9 Object colour

The default colour of grey can be used for objects that represent physical elements available in more than one colour.

Alternatively, a representative colour for the physical element can be used.

5.2.10 Drafting conventions

Visual drafting conventions such as line types, hatching and fill shall be utilised, using the relevant tools available within the **BIM authoring system**, to distinguish between different parts of the object and to show variances in depth of surface in different views.

Hatching should be assigned to the relevant materials, or part description, for the object, rather than added as 2D information.

Symbolic lines should be used in plan views, rather than solid geometry.

The use of arrays, formulas and voids when modelling the object shall be kept to a minimum.

2D lines and symbols can be used to indicate items that may not be modelled or to complement the 3D geometric information, such as to indicate flow direction or tap operating direction.

5.2.11 Material assignment

Proprietary objects should have appropriate materials assigned to them to represent the physical element's constituent materials.

Generic objects can have materials assigned to them.

5.3 What to model

This sub-section of the standard defines what should and shouldn't be included in the geometric model of the object, such as maintenance zones, hidden content and other spatial requirements.

5.3.1 Pre-planning

The geometric modelling of the object shall be planned, including any **parametric** relationships, so as to consider what graphical detail is required/necessary and what information will be visible in different views.

Note: Small elements or non-critical elements of the object may not require modelling as these may impact model performance.

5.3.2 Detail

The object shall represent the form of the physical element's external boundary without providing excessive or unnecessary detail.

Detail that would not be visible (or hidden), shall not be modelled, unless required for the intended BIM use. For example, the face of a chest of drawers only need show the face of each drawer, the drawer itself shall not be modelled.

5.3.3 Connection points

Objects that are required to connect to other objects, such as a fan unit to a duct, shall have the position of their connection points accurately modelled.

5.3.4 Spatial requirements

Objects representing physical elements that have spatial requirements, such as clearance zones, operation zones or maintenance zones, shall also have such spatial requirements modelled.

The modelled spatial requirements shall have their visibility controlled, with the default being that the spatial zones are not visible.

6.0 Object Functionality

This section of the standard discusses the requirements for improving the functionality and performance of BIM objects so as not to detrimentally affect the performance of a model into which they may be placed.

6.1 Object performance

This sub-section of the standard details measures to be taken to reduce the file size of BIM objects to improve model performance and reliability.

6.1.1 Purging

All unused or temporary modelling content shall be purged or deleted from the object, such as unused line types, reference items, images, construction lines or CAD content.

6.1.2 Saving and compressing

When the object modelling is complete, the file should be saved, using the 'Save As' function, with a different name to that used during its creation, to remove any temporary history data attached to the original file. Where supported by the **BIM authoring system**, the resulting file should also be compressed to minimise its size.

6.1.3 Testing

Objects shall be tested to ensure that they perform as intended.

6.2 Object relationships

This sub-section of the standard discusses the requirements for assemblies, nested objects and objects that are reliant on other objects to perform as expected.

6.2.1 Reliance

Objects shall be modelled so as not to be reliant on other objects, unless it is a specific requirement of that object type that it requires a host object, such as a wall mounted light fitting that would be reliant on a wall object.

Objects with **parametric** behaviour can be reliant on other object types, see clause 6.4.

6.2.2 Assemblies

Assemblies shall only be created where it may be necessary or beneficial to represent the group of component objects collectively within the assembly, rather than as individual component objects.

Assemblies can include multiple objects.

6.2.3 Nested objects

Nested objects (objects embedded within objects) shall only be created where it may be necessary or beneficial to represent the component objects within the nested object individually from the nested object itself.

Nested objects shall be restricted to two levels (i.e. an object within an object within an object).

6.3 Visibility/Views

This sub-section of the standard discusses the requirements for the visibility of objects and their display views. BIM authoring systems control object viewing in different ways, such as layers and subcategories, these need to managed to avoid a proliferation of views within models.

6.3.1 Thumbnails and previews

Thumbnail previews shall be set to a clear, appropriate and consistent view orientation and an appropriate size and resolution.

All reference items and dimensions shall not be visible in these images.

6.3.2 Object resolution

Provision for varying object resolution when viewing the object at different scales shall be provided and the graphical information displayed shall be controlled.

Three scales of resolution should be sufficient and shall be provided for each object as follows:

- Coarse: Typically used for low detail, low fidelity views, generally at a scale above 1:100. The geometry visible shall only be indicative of the physical element and can include symbolic 2D linework.
- Medium: Typically used for views at scales of 1:20 to 1:100. The geometry visible shall be sufficient to serve the purpose of representing the physical element.
- Fine: Typically used for highly detailed views, usually at scales of 1:1 to 1:20. The geometry visible can reflect the detailed geometry of the physical element.

6.3.3 View control

Views shall be named in accordance with clause 2.6.

Objects shall be assigned, in a consistent and logical manner, to the appropriate view or views.

6.4 Parametric behaviour

This sub-section of the standard discusses the requirements for creating objects with parametric behaviour.

6.4.1 Functional behaviour

Parametric objects that are reliant upon host objects in/on which they are placed, shall be created with the specific functionality to manipulate the host object, or vice versa, as necessary to suit their purpose. Such as a parametric window object that shall be modelled so as to create a corresponding void in the wall object into which it would be placed or alternatively if the wall thickness changed the window dimension would adjust to suit.

6.4.2 Geometric behaviour

Parametric objects shall be created when an object can represent geometrical variants of an object type.

The parametric capability of the object shall be limited to the available variants of the physical element it represents. For example, if a physical element is available in 4 different lengths, limit the parametric length function of the object to those 4 available lengths.

6.4.3 Multiple IFC types

Objects shall not be created with **parametric** capability to allow the object to represent multiple *IfcElementType*'s. For example, do not create one object that can represent a bath and also represent a sink, as these objects are represented by different *IfcElementType*'s.

In this situation duplicate parametric objects shall be created and assigned the corresponding *IfcElementType*, one to represent each *IfcElementType* required. For example, one to represent a parametric bath and one to represent a parametric sink.

6.4.4 Reference items

When modelling objects with **parametric** geometry, the parametric behaviour shall be controlled by reference items such as planes, lines or points.

6.4.5 Use limitation

The use of **parametric** behaviour shall be limited to that deemed necessary, or required, to avoid over-constraining the functionality of the object.

7.0 Import, Export and Linking

This section of the standard discusses methods of importing properties and content into a BIM object and also any requirements for the preparation of a BIM object for export.

7.1 Importing information

This sub-section of the standard discusses the requirements and recommendations for the import of information (graphical or non-graphical) into the BIM object. Imported geometry can limit functionality and parametric capability, increase possibility of file corruption and cause increased object file sizes.

7.1.1 Geometric information

Where possible, native geometry within the **BIM authoring system** should be used.

Geometry can be imported from other authoring systems or sources, but should only be used for temporary modelling purposes and removed when no longer required, see clause 6.1.1.

7.1.2 External data sets

Data sets (properties and property sets) can be imported into objects.

Imported data sets shall be in accordance with the data requirements of this standard.

7.2 Exporting BIM objects

This sub-section of the standard discusses the requirements and recommendations for the preparation of BIM objects for export to other BIM applications, such as the requirements for mapping non-conforming hard-coded properties from the BIM authoring system to the correct IFC property.

7.2.1 IFC assignment

The properties of '*IfcExportAs'* and '*IfcExportType'* shall be checked and values completed before object export, see clause 4.3.2.

7.2.2 Mapping of hard-coded properties

Where a **hard-coded** property is identical in description/value requirement to a defined **IFC property**, but does not identically reflect the name of the corresponding IFC property, then that hard-coded property shall be mapped to the corresponding IFC property at export.

Note: Some **BIM authoring systems** will automatically map hard-coded properties to the relevant IFC properties.

7.2.3 External data sets

An object can be linked to external data sets. Any properties required by downstream BIM uses/applications shall be natively stored on the object at or prior to export, unless such downstream BIM applications can also access the corresponding data sets.

7.2.4 Test export

Objects shall be tested to ensure export results are as intended.

8.0 Definitions & Glossary

This section of the standard provides definitions and terms used in the standard.

DEFINITIONS WILL BE COMPLETED FOR PUBLICATION

ANSI SQL

assemblies

associative dimensioning

BIM authoring system

COBie

field

generic object

hard-coded

IFC

insertion point

layered objects

library

metadata

origin point

parametric

property

property set

proprietary object

view control

9.0 Bibliography

This section of the standard lists documents that were reviewed and considered when compiling this standard. It is not implied that this standard conforms to the requirements of the documents listed below.

- AEC (UK) BIM Protocol for Autodesk Revit - v2.0
- AEC (UK) BIM Protocol for Bentley AECOsim Building Designer - v2.0
- AEC (UK) BIM Protocol for Nemetschek Vectorworks - v1.2
- AEC (UK) BIM Technology Protocol - v2.1.1
- AEC (UK) BIM Technology Protocol for GRAPHISOFT ARCHICAD - v2.0
- ISO/IEC 9075-2 Information technology — Database languages — SQL (2016)
- ANZRS - Australian and New Zealand Revit Standards - Version 3
- Autodesk Revit Model Content Style Guide - v2.1
- Autodesk Seek Metadata Style Guide - v1.0
- BIMinNZ-Handbook 2016
- BIMobject Content Style Guide - 2018
- BIMobject Revit Style Guide - 2018
- bimstore bible – Revit family creation standards - v15
- BRE BIM Attribute Naming Standard - Rev1.2
- BRE BIM Object Naming Standards - Rev1.2
- BRE BIM Shape Definitions Standard - Rev1.2
- BS 1192:2007 - Collaborative production of architectural, engineering and construction information - Code of practice
- BS 1192-4:2014 - Collaborative production of information - Part 4: Fulfilling employer's information exchange requirements using COBie - Code of practice
- BS 8536-1:2015 - Briefing for design and construction - Part 1: Code of practice for facilities management (Buildings infrastructure)
- BS 8536-2:2016 - Briefing for design and construction - Part 2: Code of practice for asset management (Linear and geographical infrastructure)
- BS 8541-1:2012 - Library objects for architecture, engineering and construction - Part 1: Identification and classification - Code of practice
- BS 8541-2:2011 - Library objects for architecture, engineering and construction - Part 2: Recommended 2D symbols of building elements for use in building information modelling
- BS 8541-3:2012 - Library objects for architecture, engineering and construction - Part 3: Shape and measurement – Code of practice
- BS 8541-4:2012 - Library objects for architecture, engineering and construction - Part 4: Attributes for specification and assessment – Code of practice
- BS 8541-5:2015 - Library objects for architecture, engineering and construction - Part 5: Assemblies - Code of practice
- BS 8541-6:2015 - Library objects for architecture, engineering and construction - Part 6: Product and facility declarations - Code of practice

- BS ISO 15686-4:2014 - Building Construction - Service Life Planning - Part 4: Service Life Planning using Building Information Modelling
- ISO DIS 19650-1:2017 (DRAFT) - Organization of information about construction - Information management using building information modelling - Part 1: Concepts and principles
- ISO DIS 19650-2:2017 (DRAFT) - Organization of information about construction - Information management using building information modelling - Part 2: Delivery phase of the assets
- CIC (Hong Kong) Building Information Modelling Standards (Phase One) - First Version
- CIC (UK) BIM Protocol v1.0
- DIN SPEC 91400:2017 - Building Information Modeling (BIM) - Classification according to STLB-Bau
- ISO 82045-5:2005 – Document management – Part 5: Application of metadata for the construction and facility management sector
- NATSPEC National BIM Guide - v1.0
- NBS (UK) BIM Object Standard - v1.3
- New York City Department of Design + Construction - BIM Guidelines - 2012
- New York City School Construction Authority – Building Information Modeling Guidelines and Standards for Architects and Engineers - v1.0
- Norwegian Home Builders' Association - BIM User Manual - v2.0
- ÖNORM A 6241-1:2015 - Digital-built documentation - Part 1: CAD-data structure and Building Information Modeling (BIM) - Level 2
- ÖNORM A 6241-2:2015 - Digital-built documentation - Part 2: Building Information Modeling (BIM) - Level 3 Ibim
- PAS 1192-2:2013 - Specification for information management for the capital/delivery phase of construction projects using building information modelling
- PAS 1192-3:2014 - Specification for information management for the operational phase of assets using building information modelling
- PAS 1192-5:2015 - Specification for security-minded building information modelling, digital built environments and smart asset management
- PAS 1192-6:2018 - Specification for collaborative sharing and use of structured Health and Safety information using BIM
- VA US BIM-Manual v2.2

New Zealand Requirements Annexure

<MASTERSPEC LOGO>

Foreword

Market feedback and the results of New Zealand National BIM surveys identified that the availability of good quality BIM objects was a constraint on BIM adoption in New Zealand and that the lack standards was part of this. As a result, Construction Information Ltd (Masterspec NZ) have collaborated with NATSPEC in Australia to develop this Open BIM Object standard.

BIM is a rapidly developing concept in New Zealand and around the world having the potential to be industry changing by facilitating digital communication between parties on construction projects and through the life of the built asset. For this to work effectively, the information needs to be created in a consistent and structured manner and this standard provides a basis for object authors and product manufacturers to do this with confidence.

Introduction

Users of BIM content need to have confidence that the objects they are getting are of a consistent quality. The intent of this standard is to provide a base structure that object developers, either in house or external, can work to.

The standard was developed after reviewing BIM related documents, guides and standard from around the world to identify what were the challenges object developers faced in creating consistent content. Several proprietary guides have been created to address aspects of this, but it was felt that there was a need for non-partisan, potentially global, standard that all could use.

Starting by developing a standard for use in Australia and New Zealand, the intention is that it is available for other jurisdictions to adopt and be able to join in contributing to ongoing development of it. As other countries come on board, if they have local requirements not met by the core document, they can develop their own annexure documents.

Scope

The approach taken is to provide the core requirements that are common to all locations in the main standard, then add a jurisdiction specific annexure that applies additional requirements or constraints over this. Therefore, this annexure is to be read in conjunction with the requirements of the Open BIM Object standard, for BIM objects being created for use in New Zealand.

NZ10.0 New Zealand Annexure

NZ10.1 Information requirements

NZ10.1.1 Classification properties

The BIM object shall include the classification properties detailed in Table NZ10A (see also the Open BIM Object standard, Clause 4.6.1). The properties shall be completed and grouped in the *OBOS_Classification* **property set**.

Property	Description	Data type	Example	Type / Instance
CBICode	The value shall be completed with a numeric value of the appropriate Coordinated Building Interchange (CBI) classification code.	Character	3821	Type
CBITitle	The value shall be completed with CBI classification description for the chosen CBI code.	Character	Timber Framing	Type
CBIVersion	The published version of CBI used.	Character	2016	Type

Table NZ10A – *OBOS_Classification* additional properties

Note 1: The **BIM authoring system** used may include **hard-coded** properties that can be used in place of these properties.

Note 2: For layered or composite objects, use the CBI code of the primary structural component.

NZ10.1.2 Product Technical Statement

The BIM object can include in the *OBOS_Technical* **property set** a property detailed in Table NZ10B containing a valid URL link to a PDF version of the Product Technical Statement (PTS) for the object.

Property	Description	Data type	Example	Type / Instance
ProductTechnical StatementURL	URL hyperlink to technical document.	Hyperlink	www.MyPTS.com/1234	Type

Table NZ10B – *OBOS_Technical* properties

NZ10.2 Naming

NZ10.2.1 File and material naming

File names and material names shall be structured as follows and composed using the fields defined in the Open BIM object standard, section 2.1.1 with the addition of a Classification field:

<Classification[1]>_<Type>_<Subtype>_<Source[2]>_<Product/Range identifier[2]>_<Differentiator[2]><Originator>

Note 1: The Classification field is the Coordinated Building Information (CBI) 4-digit code from the 'CBICode' property defined in clause NZ10.1.1.