



Deployment guidelines for IFC 4.3.x

Date: February 1st, 2021

Status: Approved by SCTE, SCE and Technical Services team

Version: 0.6

Contents

1	Background & situation	3
1.1	Current situation	3
1.2	Scope	3
2	Topics	4
2.1	Single source of truth	4
2.1.1	Stages of development process for IFC Schema	4
2.1.2	Stages of development process for Documentation	6
2.1.3	Summary	7
2.2	Expected deliverables	7
2.3	Version numbering	8
2.4	Public versus Private	8
2.5	Change management and version management	9
2.6	Backward compatibility	9
2.7	Property Harmonisation	9
2.8	Long term maintenance and release management	10
2.9	Community inputs	10
2.10	Continues integration	11
2.10.1	Quality checks	11
2.10.2	Issue management	11
2.10.3	Publication in bSDD	11
2.10.4	Translations	11
2.11	ISO	12
2.12	Example files	12
2.13	Software certification	13
2.14	Copyright Statements	13
2.15	Tooling and miscellaneous	13
3	Conclusions & summary	14
4	Terminology	16

1 Background & situation

The IFC 4.3.x release in 2020 is created by multiple individual projects. The projects have a schedule, budget and preferences for deployment.

This had led to questions for buildingSMART about the deployment tooling and policy. Recently, the IfcTunnel project was added to the team that is releasing IFC 4.3.x. With an upcoming first formal release, it is also time to start defining requirements for long-term maintenance.

This document attempts to strike a balance between project resources and long-term stability and maintenance of IFC while focussing on the quality of the deliverables.

This document is created to describe the buildingSMART International policy regarding the release and maintenance of IFC 4.3.x.

1.1 Current situation

The situation at the end of 2020, is set out below:

- The backward compatibility of IFC 4.3.x to IFC 4.2 and IFC 4.1 is broken. It has a completely different IfcAlignment than IFC 4.2 and IFC 4.1. This means software tools that have implemented IFC 4.1 and/or IFC 4.2 need to change their implementations to support IFC 4.3, and IFC 4.1 and IFC 4.2 files cannot be imported in a software tool that only has support for IFC 4.3. This is the reason why IFC 4.1 and IFC 4.2 have formally been 'withdrawn'.
- Although 2 release candidates for IFC 4.3 have been published, the alignment model is still changing based on the input from Software Vendors that are implementing it.
- The 4.3 Release Candidate 2 has a lot of quality issues, found in several reviews and automated quality checks.
- The current IFC 4.3 has no single source of truth. Schema, documentation, example files and Psets are at several locations and not consistent with each other. This led to a situation where not everything was published in RC1 and RC2 and example files are not up to date with the latest development on alignment modelling.
- There is no formal defined level of quality requirements for the deliverables and no formal policy on use of deployment tooling. With multiple projects working on something that should result in a single IFC release, the lack of a formal policy has created several bottlenecks that are limiting progress and hindering efficiency of the projects.
- IFC 4.3 has a strong implementation forum with engagement from Software Vendors that are working on an implementation and testing the release candidates. Vendors have voiced concerns about the stability of the release, specifically on the alignment model and geometry.
- The toolchain that is used to generate content to publish the IFC 4.3 release candidates requires a lot of manual actions that can only be performed by a limited amount of people. This causes a bottleneck in the publication process.

1.2 Scope

This document is focussed just on the release of IFC 4.3.x and does not describe the MVD policy for IFC 4.3. The Standards Committee Technical Executive will publish a statement about MVDs for IFC 4.3.x in Q1 of 2021.

2 Topics

2.1 Single source of truth

Currently there is no single source of truth for the development of IFC. During the first two release candidates, several data had to be combined from several sources. This resulted in missing data in the latest RC2 because parts got lost in email, and inconsistencies in the schema because the integral overview of what all projects have been doing was only available after the release.

With more and more stakeholders involved in IFC 4.3.x, it is getting increasingly important that everyone has an integral overview of the status of everyone's work. Therefore, these guidelines mandate a single source of truth to be used during the development. A single source of truth also allows to perform continuous automated quality checks on the data. This will help find inconsistencies and missing data.

It is almost impossible to define one single source of truth for everything involved in IFC releases. Therefore, the following policy is defined for IFC 4.3.x:

- The IFC schema (meaning its structure, naming of entities, predefined types, packages (domains), properties, Psets, etc.) will be stored as a class diagram in Enterprise Architect (EA).
- The documentation (meaning the definitions of the entities, definitions of properties, elaboration on usage, figures/pictures, etc.) will be stored as Markdown (MD) files on GitHub.

The choice for Enterprise Architect was made because the projects involved in the development of IFC 4.3.x already use Enterprise Architect and a cloud server to collaborate on the class diagram. The creation of the class diagram, and any other data in EA, should adhere to the IFC Modelling guidelines that have been created by the Infra and Rail projects. Where necessary these guidelines will receive incremental updates.

The creation of the Markdown files, and any other documentation should adhere to the formatting guidelines. Where necessary these guidelines will receive incremental updates.

Currently, the projects are using an online cloud server to collaborate on the class diagram in Enterprise Architect. At any time, this data can be exported to XMI and used further down the flow of the deployment tools. Every export to XMI should be published at the appropriate folder on GitHub. Every XMI export should use the correct export profile.

2.1.1 Stages of development process for IFC Schema

There are different stages in the process while developing the IFC schema. This chapter provides a suggestion to setup the process according to ISO 19650.

There are several stages during the development of the IFC schema:

- Work in Progress
- Shared
- Published

The 'work in progress' can be done in any tool of choice. Some project participants might prefer tools like BimQ, IfcDoc or excel sheets. This work is done in a private environment by a 'task team' that is developing a solution as part of the overall work.

When the task team has a solution, it needs to be integrated in the 'Shared' stage of the development. This 'Shared' stage is facilitated by the Class diagram in Enterprise Architect. The focus here is not on the visual representation of the class diagram, but on the data model representing IFC. Integrating the work of task teams into the shared environment needs a check/review/approve before it is integrated. This needs to be done by a coordination group with representatives from the projects and the Release Manager. This 'Shared' stage has the latest information approved for sharing with all projects (at the time of writing Infra, Rail and Tunnel).

When the projects decide to trigger quality checks and generate extracts, an XMI can be exported from Enterprise Architect that will be 'published' on GitHub in <https://github.com/buildingSMART/IFC4.3.x-development/tree/master/schemas> as 'IFC.xml' overwriting the previous version. This XMI holds the data model and has no focus on the visual representation of the UML class diagram. The export can only be done by someone with pre-approved authorisation from the Release Manager. Immediately after publication of the XMI, scrips that run on GitHub will perform quality checks and generate extracts like an exp schema and HTML documentation.

The resulting extracts and quality checking results can be evaluated by the project teams and used for a coordinated follow up where task teams work in solutions again. The cycle than starts again. Task teams can only work on tasks when they are given after a joint coordination by the projects.

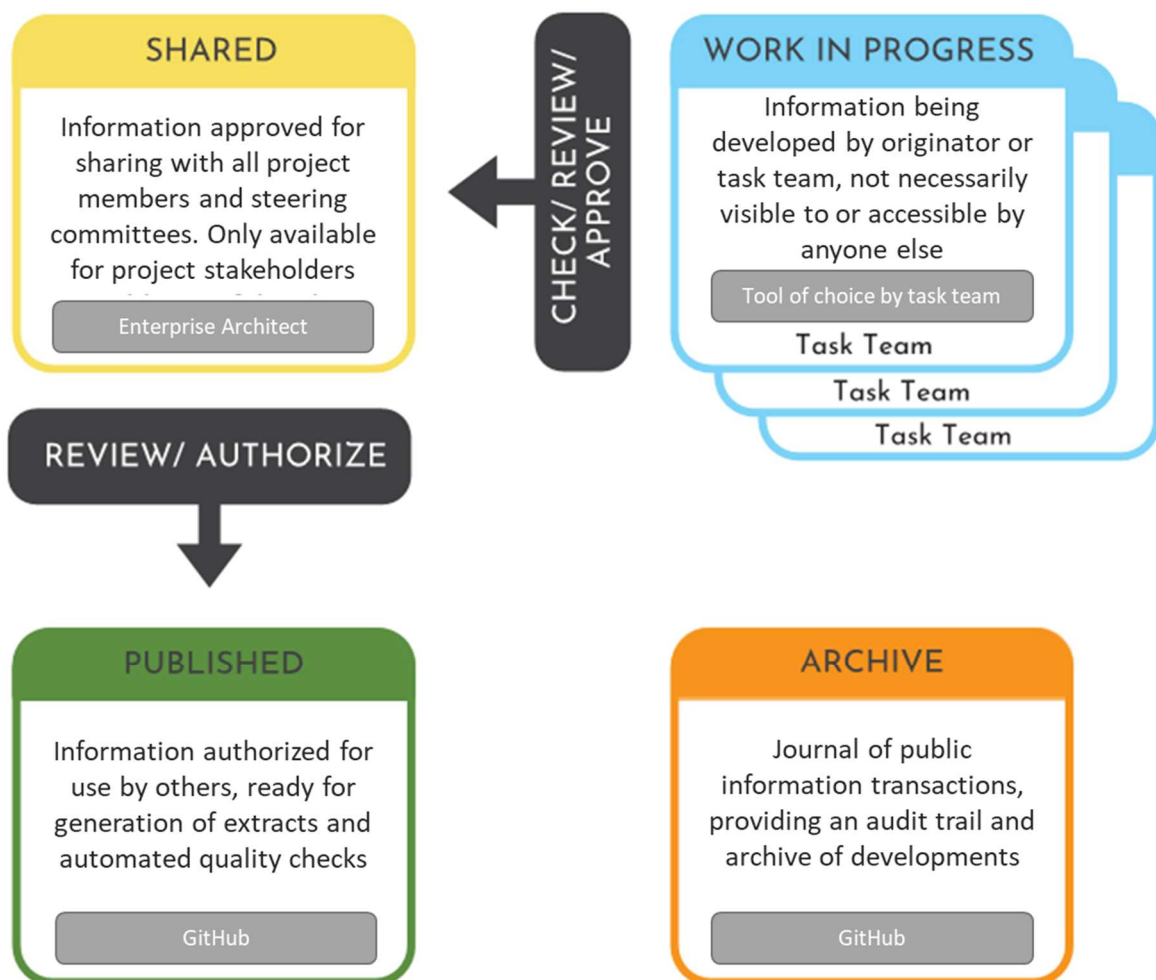


Figure 1: stages of development for IFC 4.3.x schema according to ISO 19650. Applicable tools in grey box.

Every XMI that will be published to GitHub, including the exp schema and the quality checks will be versioned and archived by GitHub.

This ISO 19650 process allows the use of ‘best of breed’ tools for task specific work, while focussing on having an up-to-date integral overview of the work in Enterprise Architect cloud that is accessible for all task teams and project stakeholders.

2.1.2 Stages of development process for Documentation

Besides the IFC Schema, the documentation is equally important for a high-quality release of IFC. The documentation is more dynamic than the schema, and more people are working simultaneously on developing and progressing it. The stages of ISO 19650 also apply here, but the number of iterations will probably be higher. Therefore, it is important that the throughput of the process is high, and everyone can keep up to date with the latest developments.

This is why the use of Markdown files on GitHub has been chosen to be used. The ‘work in progress’ can be done in other tools like Word or a text editor, but for the documentation it is highly recommended to use the built-in GitHub markdown editor. After editing or adding documentation, the work is shared with others. Depending on the authorisation of the documentation authors it will be integrated and published immediately, or after authorisation of someone who has been appointed to check the quality.

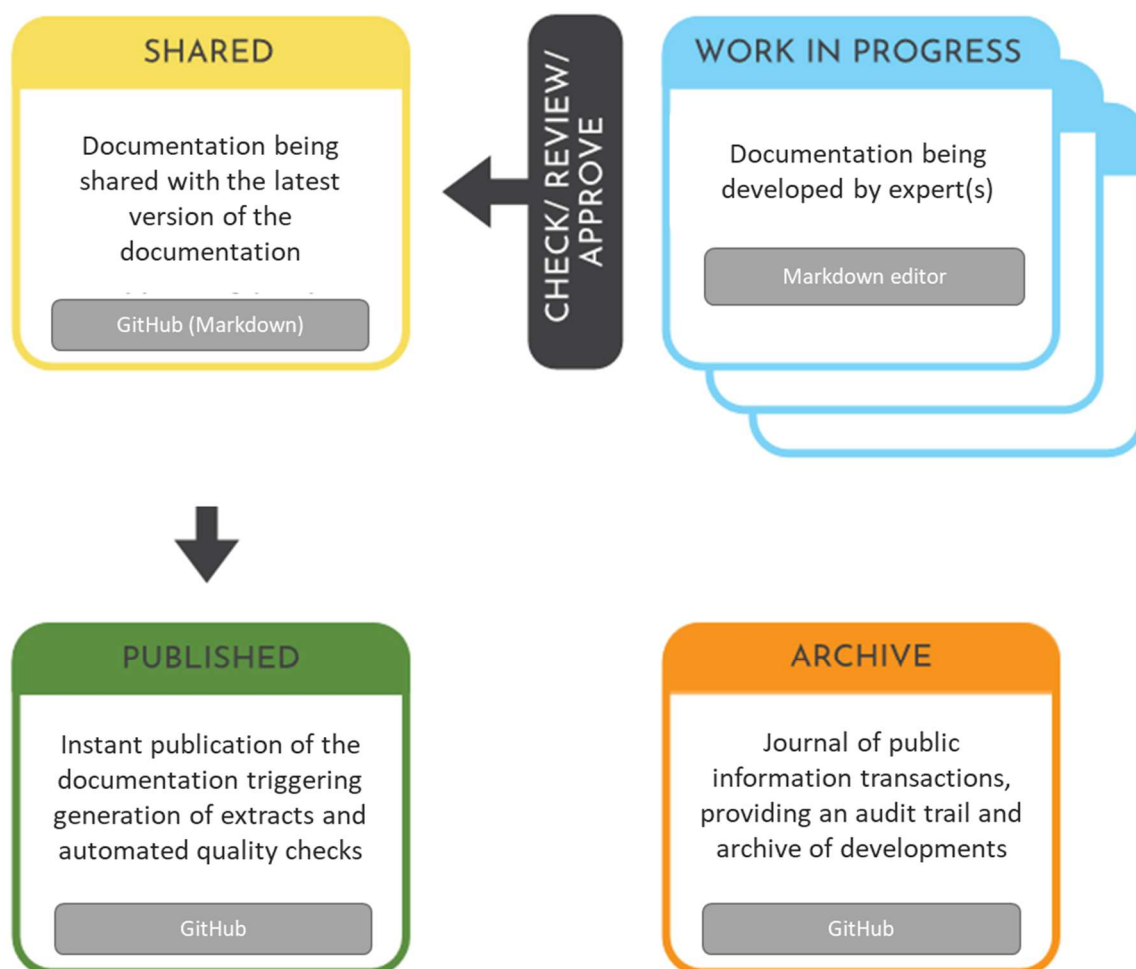


Figure 2: stage of development for IFC 4.3.x documentation according to ISO19650. Applicable tools in grey box.

After every commit of updated documentation, quality checks will run, the HTML extracts will be updated, and everything will be versioned and archived.

Markdown files are on <https://github.com/buildingSMART/IFC4.3.x-development/tree/master/docs>

2.1.3 Summary

In summary, the latest version of the IFC schema lives in an UML class diagram on Enterprise Architect. The latest and most up to date version of the documentation source is available as Markdown on GitHub. This content is available for the project stakeholders.

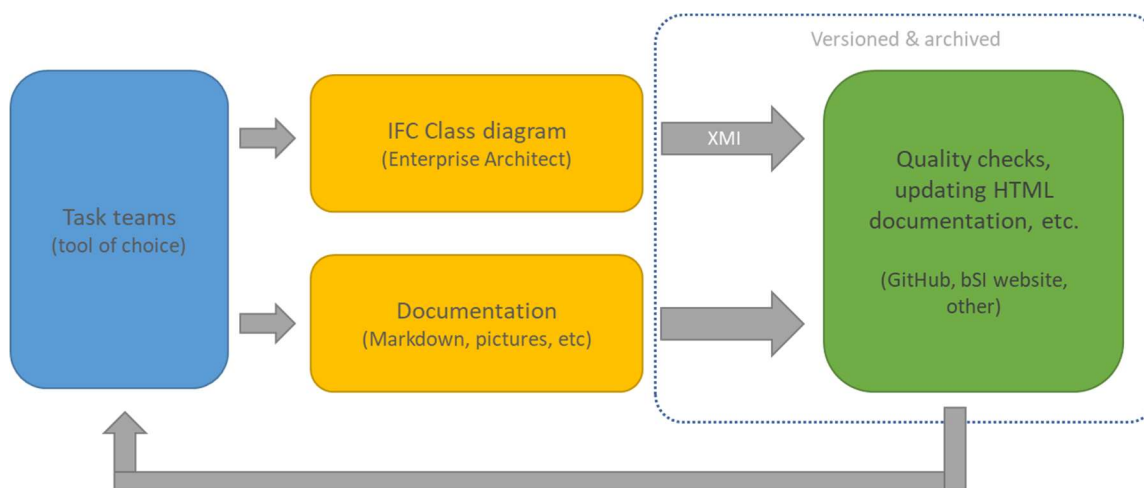


Figure 3: overview of the process of deployment, in line with ISO 19650

When reviewed and published, the automatic generation of extracts and quality checks will be triggered. Results of that are publicly available for everyone.

Everything that is published on GitHub will be versioned and Archived.

2.2 Expected deliverables

buildingSMART International expects the following deliverables at the end of each project, and with every release or release candidate:

- The source EA file(s) modelled according to the most recent UML modelling guidelines (already in possession of the projects).
- The XMI, generated from EA according to the correct profile.
- Documentation of every IFC entity, stored as Markdown on GitHub and according to the formatting guidelines for documentation.
- A detailed overview of changes compared to the previous release and the latest official release¹. This should be a table with change per entity, property, etc. For IFC 4.3.x, the IFC 4.2.0 can be regarded as the latest official release to compare to.
- Release notes with the most significant changes compared to IFC 4.2.0, IFC4, and to the previous release candidate.
- Multiple example files, providing references for implementors to implement the version in their software. The main and important geometric entities should have

¹ Since IFC 4.3.x is very different from, and not backward compatible with IFC 4.2 and IFC 4.1, those versions have been withdrawn. The latest official release to compare to would be IFC4 (4.0.2.1). It is unrealistic to ask a detailed change overview to such an old version.

extensive examples. For IFC 4.3.x the focus should be on alignment, but also encompass more regular examples.

- An overview of quality checks performed by the release team, including the results.

Optionally, the following deliverables may be produced:

- HTML package with documentation
- Express schema + Pset XML
- XSD Schema
- ifcOWL + Documentation
- ifcJSON Schema
- ifcHDF5 schema
- A script to transform IFC 4.0.2.1 and/or IFC 2x3 TC1 files to IFC 4.3.x

When the optional deliverables are produced, they need to be of sufficient quality, and they need to be consistent, correct and complete. When these deliverables are not produced by the project(s), some will be generated by buildingSMART International. A formal version of these deliverables will be published on standards.buildingsmart.org.

2.3 Version numbering

When publishing an intermediate version of the schema on GitHub, the release team can choose to add the version in the XMI.

The generation of extracts will use that version number during the generation of the extracts and quality checks. The version number provided in the XMI will be extended with the commit ID or date/time stamp. This ID is a unique identifier to track the source of issues and keep all information in sync.

Formal releases like a 'Candidate Standard' or 'Final Standard' can only be published after approval of the Standards Committee following the buildingSMART Process.

2.4 Public versus Private

During the development of IFC 4.3.x there is a balance between publicly available information and information in a private environment.

Project stakeholders have an interest to develop in a protected private environment for some time, before releasing the results to the public. This is respected in this policy document.

The following policy applies:

<i>Publicly available</i>	<i>Not public during development; made public after an (intermediate or incremental) release</i>	<i>Not public; in possession of buildingSMART International Management office after release</i>
Markdown Documentation	XMI	Source EA file(s)
Quality checks (bSI MO)	Changelog	EA Cloud access
Generated content (HTML package, Express schema, etc)	Release notes	Report on the results of quality checks
	Example files	
	Additional Quality checks	

2.5 Change management and version management

Version and change management for documentation is done automatically by using Markdown files on GitHub.

Every meaningful export of EA to XMI should also be published in the appropriate folder on GitHub. Since it is difficult to automatically detect changes in the XMI, the release team should take notes during changes in EA to be able to create proper release notes and a detailed overview of changes.

2.6 Backward compatibility

With the changes to `IfcAlignment` and alignment curve in IFC 4.3 RC2, backward compatibility to previous versions with alignment is broken. This means IFC 4.3 will not be backward compatible with IFC 4.2 and IFC 4.1. Given the fact that this backward compatibility cannot be in place, decisions about Property harmonisations and Software Certification can have different outcomes.

2.7 Property Harmonisation

Currently, there are 4 sources for the IFC 4.3 release:

- The IFC 4.2 base
- Infra project
- Rail project
- Tunnel project

In normal circumstances, the projects will be extending IFC 4.2 with new entities and properties. The task of the projects is to harmonise entities and properties among each other (hence the importance of a single source of truth).

In principle, properties of the base IFC 4.2 do not have to change, since 4.3 is in principle an extension. A situation could occur where property harmonisation with IFC 4.2 is also desired. For example, in cases where there are new insights have been gathered and a change in properties from the 4.2 base is of value. Since backward compatibility to IFC 4.2 is already broken, improvements to properties from the IFC 4.2 base are allowed.

The buildingSMART policy on properties is that every property with a certain name should have the same definition/description, same data type and same allowed values. This means that every property name should be unique but can be used multiple times in different Psets.

This policy is in line with the upcoming ISO 12006-3 standard for dictionaries. The buildingSMART Data Dictionary (bSDD) also adheres to this policy for properties.

It is advised to have one Markdown page on GitHub for every property(name) with the description and definition. This page can be referred to by the Psets and used for the generation of the HTML documentation.

Note: this document describes the policy for IFC 4.3.x deployment. buildingSMART International is aware that on several topics, and especially this property normalisation topic, it is difficult for the projects to fully adhere to this policy. It is recommended to the projects to follow this policy when working on the topics, but there is not a requirement to fix all the IFC 4.2 heritage that does not comply to this policy yet. Support from buildingSMART International is available for the projects to normalise properties in the full schema.

2.8 Long term maintenance and release management

After the delivery of the final version of IFC 4.3.0, there will be updates. Some of those updates will come from projects (like the Tunnel project), and some will come from community inputs and implementors feedback.

Separate domain experts will be asked to form maintenance teams for the individual domains. A release manager will be appointed for major and minor releases of IFC 4.3.x. and coordinate with the domain expert teams. The release manager will also coordinate with other release managers and stakeholders like the SCTE.

Maintenance will be done on GitHub, using the Markdown documentation and the EA generated XMI.

It is a strongly recommendation to have an XMI that can be processed by multiple tools (like Papyrus, Altova, etc). In the beginning, EA will be used to update the schema, continuing the work as it has been started by the projects.

A first step would be to modularise so multiple XMI files together represent the full IFC schema. Separate files help with maintenance by separate domain expert teams. This modularisation is also represented in the folder structure of the MD files for the documentation.

Every change to the documentation or to the XMI will trigger an automatic generation of HTML documentation and updated Express schema and quality checks. Manual triggers at the correct time (coordinated by the release manager) will generate more machine-readable schemas, linking to the bSDD and to the Translations framework.

Depending on the speed of development of IFC5, and the desired updates for IFC 4.3.x, a suitable effort will be made to transform from EA to tool independent representation of a class diagram for IFC.

2.9 Community inputs

After the projects deliver a final result, the IFC 4.3.x maintenance will come under the responsibility of buildingSMART International.

Publishing IFC 4.3.x on GitHub makes it transparent and available for broad community inputs. After the release of IFC 4.3.0, there can be two sources that provide updates:

- Updates from projects adhering to the buildingSMART Process and are coordinated by the Release manager.
- Community suggestions for changes to the schema or the documentation are being submitted as 'pull requests' on GitHub. When a pull request comes in, a quality assurance and checking procedure will follow. This procedure will provide guidelines for decision making and instructions for motivations on approval or rejection of the community suggestion. This procedure will be published in a separate document in 2021.

All changes are being tracked and documented. Traceability will go hand in hand with version control.

2.10 Continues integration

After every update of the XMI or Markdown, automated quality checks are being performed. This is done using Travis tools and Python scripts. This is done by the deployment team from buildingSMART International, in close coordination with the release manager. Users do not have to perform any special task for this.

The scripts perform quality checks and reports for the release managers and the domain experts.

Part of this 'continues integration' setup with Travis is the publication of the IFC entities, predefined types, properties, etc. into the buildingSMART Data Dictionary and the Translations framework.

2.10.1 Quality checks

Every new XMI commit, and some additional events, will trigger the quality assessment scripts to run. The scripts are transparent and openly available for everyone to check on GitHub.

The scripts check for the correct use of the term 'Pset_', duplicate names between properties and predefined type enumerations, and many other checks.

The kind of checks that are performed by the script(s) will change over time, depending on the needs of the project and the stakeholders. These scripts are the most important quality checking source for the release manager and the SCTE review.

Projects are advised to submit requests for specific quality checking functionalities to buildingSMART International.

2.10.2 Issue management

The results of the Quality checks that run with every new publication of XMI on GitHub provide results as GitHub issues on <https://github.com/buildingSMART/IFC4.3.x-development/issues>

The open issues can be reviewed by the project team, assigned, grouped, prioritized and discussed.

When a new version of the IFC Schema (in XMI) is published in the GitHub repo the Quality checks will be triggered again. When issues are fixed in the new version of IFC, these issues are automatically closed. New issues that are found will be added to the list. Issues that have manually been closed, but not resolved in the XMI will be opened again.

It is advised that the projects use the same location to also discuss manually found issues, but this suggestion is open for the projects to decide on themselves.

2.10.3 Publication in bSDD

Part of this 'continues integration' setup with Travis is the publication of the IFC entities, predefined types, properties, etc. into the buildingSMART Data Dictionary.

A script will be written that collects data from the XMI and MD to aggregate into a JSON file. That JSON file is read by the bSDD (when the release manager approves a formal new version).

2.10.4 Translations

Part of this 'continues integration' setup with Travis is the publication of the IFC entities, predefined types, properties, and all their definitions into the Translations Framework.

buildingSMART uses the Crowdin Translations framework which is available on translations.buildingsmart.org.

A script will be written that collects data from the XMI and MD to aggregate into POT files (an international standard for translations templates). The POT files are structured according to the IFC domains (packages in XMI) and are read by the translation framework. Every update triggers a new translations workflow.

When translated strings have been proofread and approved, they will be available through an API, and automatically pushed to the bSDD.

2.11 ISO

There is a strong desire from stakeholders that IFC 4.3.0 will be submitted to ISO for formal approval.

During the development of IFC 4.3.0, the projects should conform to the ISO guidelines for writing a standard. These guidelines were made available for the first time in January 2019 and have been updated subsequently. Project participants can request information about this from their Room coordinators.

There are no known ISO requirements for the layout of the HTML documentation of the standard.

After the buildingSMART Final release of IFC 4.3.0, a process will be started to submit IFC to ISO. This will be coordinated by a representative from buildingSMART International that will be part of the IFC 4.3.x release team during development.

2.12 Example files

Example files are of great importance for interoperable implementations of IFC. The projects need to deliver as many example files as possible. The 'Software Validation' phase of the project guarantees that the files are valid.

Example files will be stored in the 'IFC 4.3' folder on <https://github.com/buildingSMART/Sample-Test-Files>

Given the continues development of the IFC schema (specifically on the alignment model) the example files can become out of sync with the latest version of the IFC schema. This is why it is strongly advised to add the coordinator of the example files to the team that coordinates work from task teams that is integrated in the 'shared' stage (see Figure 1)Figure 1: stages of development for IFC 4.3.x schema according to ISO 19650. Applicable tools in grey box..

The projects are organizing interactions and engagements with Software Vendors while the development of the IFC schema is still undergoing significant changes. Software Vendors should be informed immediately when example files are out of sync with the development on the IFC Schema to avoid unnecessary commitment of recourses from implementors on outdated examples.

2.13 Software certification

Software Certification for IFC compliance is done against an MVD. Guidelines for IFC 4.3.x MVDs will be published in early 2021.

Since the current IFC 4.3 RC2 is not backward compatible with IF 4.2 and IFC 4.1, there will be no investments from buildingSMART International to create a Software Certification program for IFC 4.2 and IFC 4.1.

After the release of IFC 4.3.0 and the finalization of the MVDs, a process to define Software Certification requirements will be started by the Technical Services steering committee of buildingSMART International.

2.14 Copyright Statements

All deliverables should have appropriate copyright statements. All documents should have at least the notice “© *Copyright buildingSMART International Limited*” to ensure that buildingSMART International’s copyright is made clear. Depending on the content of the deliverable, additional notices should be added. This could be (but not limited to):

- Copyright Notice
- Copyright exclusion Notice
- No Warranty Notice
- Trademark Notice
- Correspondence Notice

2.15 Tooling and miscellaneous

There have been many discussions about deployment tooling. These deployment guidelines are focussing on having a process that creates an integral overview of the developments. This provides efficiency and quality to the distributed development of IFC 4.3.x.

The main priority of these guidelines is to make sure there is one location with the latest coordinated integral version of the IFC Class Diagram that is available for all stakeholders in the projects.

Because the involved projects already made use of Enterprise Architect and modelling of IFC as class diagrams in the cloud, these deployment guidelines are built on top of that already made decision.

The task groups that work on parts of the IFC development are still free to choose the tool they prefer (IfcDoc, BimQ, excel, etc) as long as the result of their coordinated activity is integrated in the shared space on the Enterprise Architect Cloud server (see Figure 1).

For future maintenance and update of IFC 4.3.x the IFC Class diagram is preferably not limited to Enterprise Architect, but available in many different UML modelling tools. This topic needs to be evaluated in a later phase after the final release of IFC 4.3.0.

3 Conclusions & summary

With multiple projects working on one deliverable, having an integrated overview of the status is vital for consistency and quality control.

With the release of IFC 4.3.0 coming closer, the characteristic of collaboration between projects is changing. Where it was more efficient to focus on individual domain extension first, the focus should be towards delivering one integrated and consistent standard.

With the strong desire of stakeholders to publish IFC 4.3.0 as an ISO standard, the integration and coordination between projects is vital.

Having an integrated overview of the ongoing work should therefore be available at any given time in the process, and not only when a release candidate is published.

This document states deployment procedures based on phases 'Task Team', 'Shared' and 'Published'. The shared space is where work from all tasks teams is being aggregated and coordinated.

This document suggests the use of Enterprise Architect cloud to be facilitate this 'shared space'. This environment is already in use by the projects and has proven to be capable of importing from multiple sources that are in use by the projects. It also has authorisation management to manage rights to the data and keep the data private for non-stakeholders.

This document suggests GitHub to be used as the 'published' space. This is where buildingSMART International is facilitating quality control algorithms.

When projects collaboratively suggest the use of other tools to be used to reach the state objectives, they are free to do so.

In summary, the most important points from this policy are:

- Objective of this document and defined procedure, is to have one shared area where the latest (agreed) version of IFC 4.3.x is always available for the stakeholders and project participants.
- This integral overview for all stakeholders in all projects allows for efficient coordination and continues quality checks to be performed. It avoids surprises when publishing release candidates.
- A release manager will be appointed for major releases of IFC 4.3.x to help coordinate the integral overview and quality control between projects.
- Enterprise Architect (cloud) will be used as 'shared space' for the IFC data model. This decision is based on the fact that all projects already use this tool and many stakeholders are able to work with it.
- Markdown files on GitHub will be the single source of truth for documentation and definitions. This decision is made on the fact that it does not require any additional tools, provides the possibility for many stakeholders to work on the

documentation in parallel, and integrates very good with the quality control systems of buildingSMART International.

- Quality checks need to be done by the projects; quality checks from buildingSMART International will run on every published change.
- All published deliverables will be versioned and traceable on GitHub. Formal deliverables will be published on standards.buildingsmart.org after approval by SCTE, SCE and SC.
- A separate document about the MVD policy and deliverable guidelines will follow later.

4 Terminology

bSDD	buildingSMART Data Dictionary	An online service from buildingSMART International to host and distribute data standards.
bSI	buildingSMART International	
class diagram		A class diagram in the Unified Modelling Language is a static structure diagram that describes the structure of a system by showing the system's classes, their attributes and the relationships among objects.
EA	Enterprise Architect	Software tool used to create and share definitions. In buildingSMART used to create Class diagrams of IFC.
GitHub		Online service from Microsoft to collaborate on projects. Any time the term 'on GitHub' is used in this document, it refers to this repo: https://github.com/buildingSMART/IFC4.3.x-development
ISO	International Organisation for Standardisation	ISO is an independent, non-governmental international organization with a membership of 165 national standards bodies.
MD	Markdown	A syntax for defining and sharing text. In buildingSMART it is used to create documentation for IFC.
MVD	Model View Definition	A subset from IFC, plus additional requirements and a definition of a conformance level for software.
Pset		A standardized set of Properties, defined by buildingSMART and part of the IFC standard.
RC2	Release Candidate 2	the second release candidate of an IFC publication.
RCx	Release Candidate x	The x release candidate of an IFC publication.
Release Candidate		An informal publication of a possible formal release of a buildingSMART Standard. A Release Candidate can be published leading up to a Final Standard, or to a Production Standard, or to a Candidate standard. To avoid confusion: a published 'Release Candidate' is not a 'Candidate Standards'
SCTE	Standards Committee Technical Executive	the highest technical policy and advisory board of buildingSMART International
Travis		Online service for continuous integration during project development and deployment. In buildingSMART used to run quality checking scrips every time a certain trigger occurs on GitHub (for example an update of documentation or a new version of IFC).
UML	Unified Modelling Language	Standard language to define agreements. In buildingSMART it is used to define the IFC data standard.
XMI	XML Metadata Interchange	A standard for exchanging metadata information via Extensible Markup Language (XML). In

		buildingSMART used to exchange the IFC schema from Enterprise Architect (EA) to others. Typically, IFC is published as XMI on GitHub.
--	--	---