



# Demystifying COBie


David Spehar, dRofus

*Presented by*



# Table of Contents

Why COBie?	<u>5</u>
How does COBie work?	<u>7</u>
Roles and Responsibilities	<u>9</u>
The Building Owner	<u>9</u>
The Design Team	<u>13</u>
The Builder	<u>14</u>
COBie Data Management with dRofus Conclusion	<u>15</u>
Flexible Data Structure	<u>18</u>
The COBie Deliverable	<u>22</u>
References	<u>25</u>



# DESCRIPTION



What really is COBie, how is it being used and what does it mean to deliver it? More importantly, how much will it cost me? These are just some of the questions teams are asking as more and more building owners are starting to adopt BIM standards and require data deliverables at handover. Increased pressure is put on design and construction teams to manage data deliverables, many times in a COBie format. In this white paper I will address the who's, what's, why's and how's around capturing, managing and ultimately delivering COBie.

I'll explain the importance of understanding the building owner's needs for capturing data to ensure teams efficiently collaborate to meet requirements with minimal waste and rework. Finally, I'll talk about using dRofus to manage the COBie data in a collaborative platform and how it can be used to provide a true COBie-compliant BIM deliverable – not just a COBie spreadsheet.

# ABOUT THE AUTHOR



David has over 30 years of experience in the AEC industry and has been a registered architect since 1994. He was an early adopter of BIM and has been an advocate of leveraging emerging technologies since 2003. David joined dRofus in 2017 as a Client Solutions Manager and is currently the Regional Sales Manager for the Americas responsible for assisting clients with the successful adoption of dRofus through transitioning to data-driven design workflows.

In addition to his experience with applying design technologies, David has a strong background in architectural project management. He was a project manager for over 12 years before shifting his focus to implementing technology in large design firms. David is a well-known subject matter expert on design technology and has regularly presented at various international industry events throughout his career.



# Why Cobie?

COBie is an international standard for building data exchange regularly used in the handover process from construction to operations and is widely associated with BIM. The COBie Project, from which the current standard originated, was started in 2005 by the United States Army Corps of Engineers, NASA and the National Institute of Building Sciences (NIBS).

According to Bill East, Prairie Sky Consulting, the COBie project started by asking 2 questions around how data could be used to replace the traditional process of handing over disconnected and unorganized physical media. Media such as Operations & Maintenance manuals and paper drawings, or even electronic media such as pdf's and CAD files that end up in a pile somewhere.

The first question was – “What content really needs to be exchanged?”  
Which raised several other questions like:



Which assets need to be captured? Such as mechanical equipment, plumbing fixtures, etc.



How much information about the asset is required? Such as manufacturer, model number, warranty information, etc.



And how should the information be categorized? By each asset type, by every instance of an asset type, which system they are part of, where the assets are physically located in a building, etc.

The second question was – “In what format should information be delivered?” XML was established as the underlying standard which is conveniently readable in Excel. While Excel is not a standard, nor the best way to capture COBie data, it is a ubiquitous software that just about everyone knows how to use. Because of this, it has become the default approach to capturing and delivering COBie

In simplest terms, COBie is a data structure intended to replace unorganized traditional handover documentation with organized handover information that can be input directly into facility operations, maintenance and asset management software systems at project completion.

As part of the COBie project, studies were done to estimate the value to building owners. If we're going to make such a huge process change, what is the ROI? One such study titled the Assessment of Life Cycle Information Exchanges (LCie) by the US Army Corps of Engineers estimated a staggering savings of 96% or over a half million dollars. Is 96% savings realistic on every project? Maybe not, but the bottom line is that, if done the right way and for the right reasons, the savings realized in a data handover will most definitely exceed the investment in software and people required to deliver it.

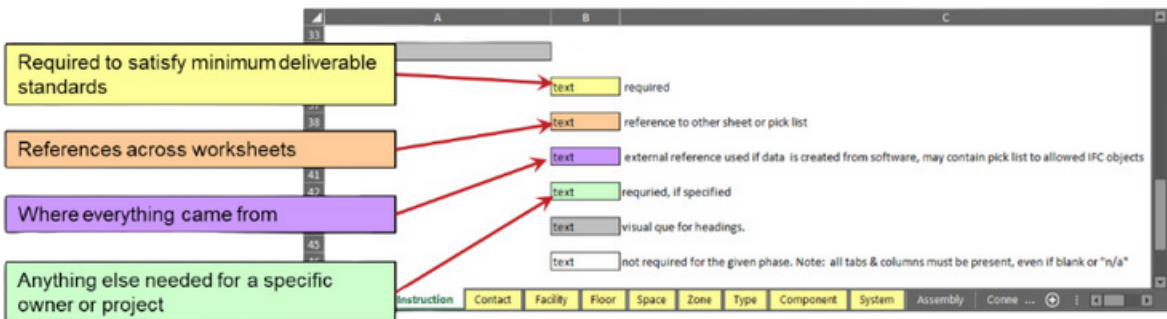
So after all the research, testing and validation showing that a data standard is good - why is COBie so misunderstood?

Too often COBie is viewed as a reactive data collection exercise, not collaborative information management.

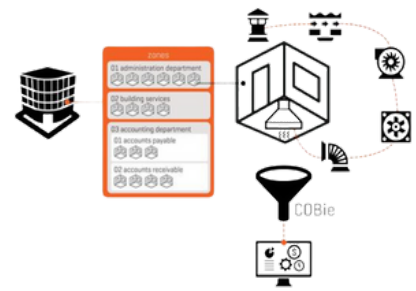
# HOW DOES COBIE WORK?

Since we've established that the COBie Excel file is the most common and human readable format, let's use it to help demystify the ambiguity around organizing the data and who's responsible for what.

If you're not familiar with the COBie Excel file, you can get information on the latest versions [here](#). It contains multiple worksheets that represent a data schema. Each is organized so that every column represents a data field and each row is for the unique elements that contain the data. The columns are color coded to define their purpose. Some are required and clearly defined for a minimum deliverable (yellow and salmon), while others are source software or owner specific (purple and green).



Each worksheet represents a table in the data schema and the salmon fields are a way to connect the data across worksheets. The idea is that anything someone would need to know about an element is contained in a particular worksheet or can be found through these connections.



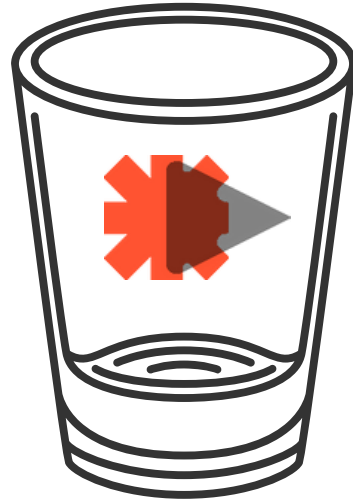
Here’s an example. Let’s say you are responsible for maintaining all the VAV boxes in a building. What would you need to know? How many different types there are, where are they located, which air handlers are they connected to, who manufactured them, what type of service is required, when is it required, and who can you contact for help? Basically, COBIE is formatted to help teams capture all this information (and more) IF needed.

Sounds simple, right? Unfortunately, it is usually complicated by lack of clarity or definition around what is REALLY required and why.

You’ve probably heard requests for a COBIE spreadsheet, a COBIE ifc model or a COBIE compliant BIM. Or in some cases, a request for all three. Because of this, there is much confusion around how to collect, organize and manage a COBIE deliverable. Many times this confusion stems from teams focusing all of their energy on answering the second question in the intro above (In what format should information be delivered?) when no one has asked or provided a clear answer to the more important first question (What content really needs to be exchanged?). Unfortunately, confusion leads to uncertainty around scope which leads to teams either overestimating the cost to collect the data or underestimating it hoping that someone else will figure it out.

Before we look at how COBIE data is collected and delivered, let’s talk about the importance of defined roles and responsibilities to make the process as efficient as possible. After all, COBIE is intended to be a collaborative effort.





## Roles and Responsibilities

The key to a successful and cost effective COBie deliverable is clarity. Clarity that is often overlooked because of uncertainty around what information is required or what will be done with the information. Clarity around who's responsibility it is to gather said information and when it needs to be gathered. Clarity defining the final deliverable format for the information.

Clarity that only the Building Owner can provide. So, when we talk about roles and responsibilities, clearly the Building Owner's is paramount.

## The Building Owner

The Building Owner holds the keys to a team's efficiency and accuracy. Before a team can even start planning their deliverable there are several steps an owner must take to define the contents that will greatly impact the cost of capturing COBie data. Would you hand an empty glass to a bartender and ask them to just fill it with something?

## Define Purpose

Data doesn't become information until it is organized, structured and has a purpose. COBie solves the organized and structured part but it doesn't solve the purpose. What is the purpose of the COBie data? Assuming it will be consumed by a Computer Aided Facility Management (CAFM) or Integrated Workplace Management System (IWMS), the big questions are how and for what aspects of facility/asset management? Defining the purpose first will help guide you through deciding which worksheets and which data fields are required.

## Define Worksheets

While the worksheets are connected, they may not all be required based on the purpose. If you don't need them – don't include them. For instance, if information isn't needed to define the specific tasks required to maintain an asset, then maybe the Job worksheet isn't required. Or if the Zones aren't required to categorize spaces, then don't include that worksheet either. Think of how much time could be spent capturing data that will never be used.

## Define Required Data

Be practical about what you need to know. Is knowing the shape, color, finish, constituents or features of an asset type important? There are roughly 50 columns across the worksheets that are green or "required, if specified". If they're not required, don't specify them. This, on its own, can greatly reduce the cost of capturing the data.

## Define Assets

This one is almost too obvious to mention but it needs to be said since it has huge ramifications across many worksheets. With thousands or tens of thousands of products installed in a building, this can get out of control quickly. All products may not be considered assets, so be precise about what you need to manage. If possible, avoid “throwing them in just in case”.

## Define Data Sources

In many cases there are suggestions for fields such as Category or Description, or there are fields that are intended to match values found in the drawings. Will you be using the OmniClass Tables for the Category? If so, how deep into the tables is required? If not, what will be used (and provide it)? What should the Description be? Something that ties a specific naming or coding system of yours to the values on the drawings? Or better yet, provide the design team with your specific codes to input into the drawings to start. Many times teams spend a great deal of effort retroactively applying codes to content that could have been done much faster if known up front.

## Document it!

And finally, make sure everything is clearly documented so teams can plan accordingly. Here’s a great example from Alberta Infrastructure’s Digital Project Delivery COBie Requirements that does just this.



### Responsibility Matrix

- x Indicates when the field is first populated by the Consultant
- x Indicates when the field is updated by the Consultant
- o Indicates when the field is first populated by the Contractor
- o Indicates when the field is updated by the Contractor

COBie column	Requirement Information	Example	Project Phase			
			SD	DD	CO	SP
Floor.Name	All levels that exist within the facility	P1	x	x	o	
Floor.CreatedBy	Reference to Contact.Email	James.Smith@email.ca	x	x	o	
Floor.CreatedOn	Date and Time created	2017-07-21T09:25:25	x	x	o	
Floor.Category	Category of floor	Floor	x	x	o	
Floor.ExtSystem	Software used to generate the element	Autodesk Revit Architecture 2011	x	x		x
Floor.ExtObject	IFC type of the object	IfcBuildingStorey	x	x		x
Floor.ExtIdentifier	GUID provided by automated export	3eM8WbY_59RR5TDWs3wwJN	x	x		x
Floor.Description	Corresponding level description	Parking Level One	x	x	o	
Floor.Elevation	Elevation of floor in mm	-8600	x	x	o	

## The Design Team



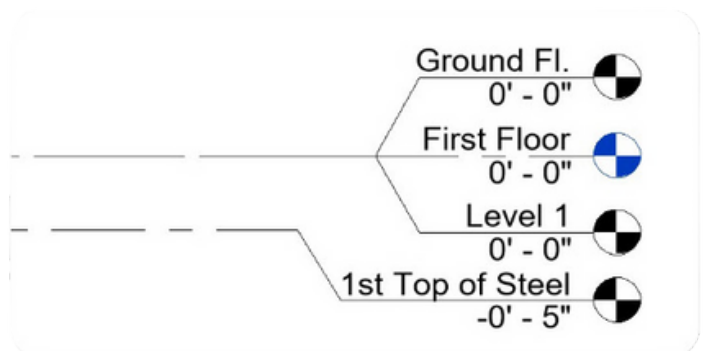
## Model Fidelity

For those of you living in the BIM management world – the importance of well-built models is nothing new. Unfortunately, it continues to be an issue that plagues almost every project. Before the days of data deliverables like COBie it was an annoyance, now it can be the source of huge amounts of lost revenue. Lost revenue fixing poorly built or incomplete models to create accurate data exports.

## Data Consistency

While not all data needs to (or should) live in models, models can (and should) be a source for a portion of it. Since many COBie fields reference drawing information or physical location of rooms and assets, it stands to reason that the models should be accurate since they are the underlying source of the drawings. Simple things like consistent naming of levels across models is now a necessity. Also, identifying the

correct source content for connecting COBie data is critical. Are the toilets in the plumbing model the correct ones or are they the toilets in the architecture model? (don't pretend you don't know what I'm talking about)

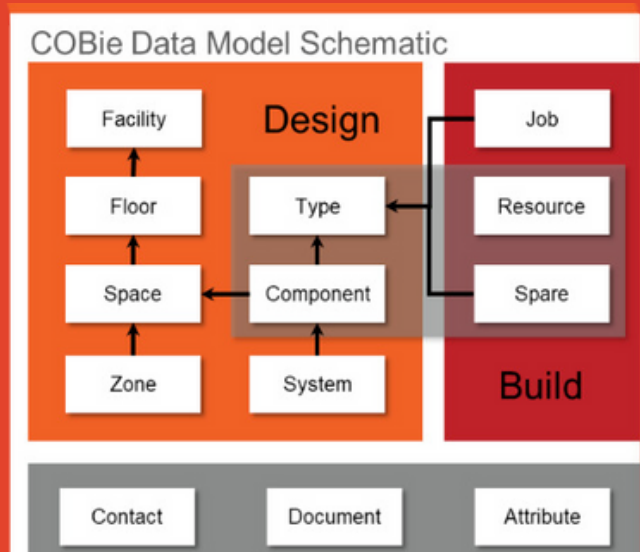


# The Builder

The final COBie deliverable rests in the Construction Team's hands. Typically the Trade Contractors will be responsible for adding asset information and in most cases the Construction Manager is ultimately responsible for collecting, organizing and validating everything. When you really break down a Design COBie deliverable vs. a Construction COBie deliverable you can see that most of the effort falls on the contractors.

## Data Consistency

While there is no denying that there's effort involved in preparing a data deliverable like COBie, it will require far less effort if it is not treated as a data collection exercise but instead, it's treated as a collaborative effort where the building owner defines their requirements and the design team have done their part to collaborate on defining the building's contents. This integrated data collection workflow is where the use of collaborative database software is a must. But which one?



# COBie Data Management with dRofus

There's a perception that all this COBie data should live in BIM but BIM authoring platforms only represent a portion of the software platforms that support the buildingSMART alliance information exchange standards. Many assume that COBie data collection can be completed in the Design Team's BIM and then "handed off" to the Construction Team for a final IFC export. There are tools like the Revit COBie toolkit which many are familiar with. If you've used it, you know that it works but has proven to be time consuming, cumbersome and prone to user error. In reality a portion of the data can be captured in models but you'll most likely benefit from a software better suited for managing large amounts of data in a tabular format with a single aggregated export. I'm sure there are integrated teams somewhere that will say they've done it in models but I'm fairly certain they are the exception.

---

## ENTER dRofus

You might ask "What is dRofus?" [dRofus](#) is a planning and data management software used to create and manage a project's design requirements. We're also a BIM collaboration tool with Archicad, Revit and IFC integrations allowing teams to validate design requirements against

model development. dRofus supports the evolution of projects by allowing teams to share their knowledge in a collaborative platform where owner requirements are captured and validated through all project stages.

This “one-stop-shop” approach to managing all of a project’s design requirements in a single database naturally lends itself to delivering COBie. dRofus is used for much more than just collecting COBie data, but we’re going to focus on the COBie stuff for now. And if you’re interested, we have worked directly with the buildingSMART alliance to test and demonstrate our ability to meet their information exchange standards which is documented [here](#).

## How dRofus works for COBie

dRofus is built on a bunch of different modules intended to capture a building’s design requirements that can be expanded upon as a project progresses. Coincidentally, this follows the COBie data structure pretty closely. The process is fairly straightforward:

**1**

Project requirements get planned in dRofus so teams can see exactly what needs to go into the building. From the rooms to where assets are used to the systems they’re part of.

**2**

Then modeled content gets validated against these requirements. So now COBie data isn’t extracted out of models after the fact – rather it is connected to and validated against defined requirements. This vastly improves the speed and quality of the deliverable.

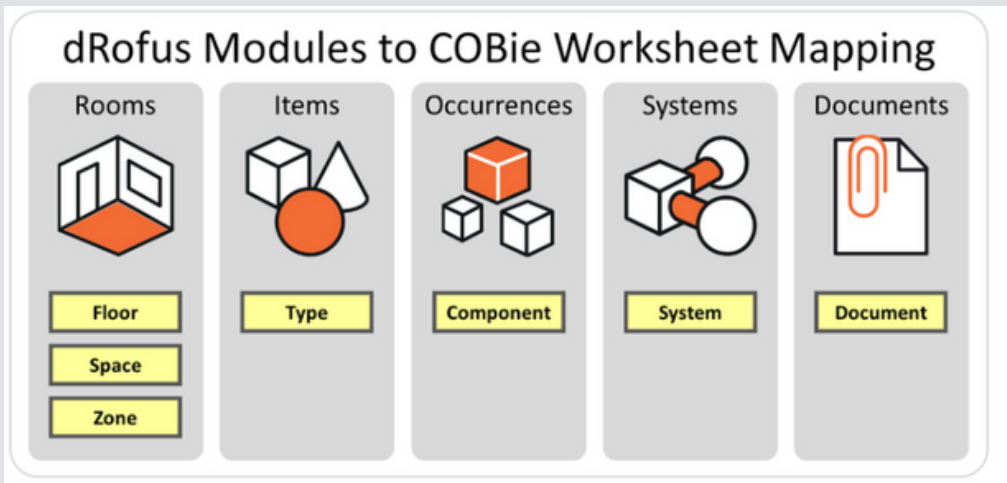


# 3

Each team member adds their information directly into a shared location, as it's known. So now contractors can come in and layer their "as installed" information directly onto the existing data without duplicating effort.

# 4

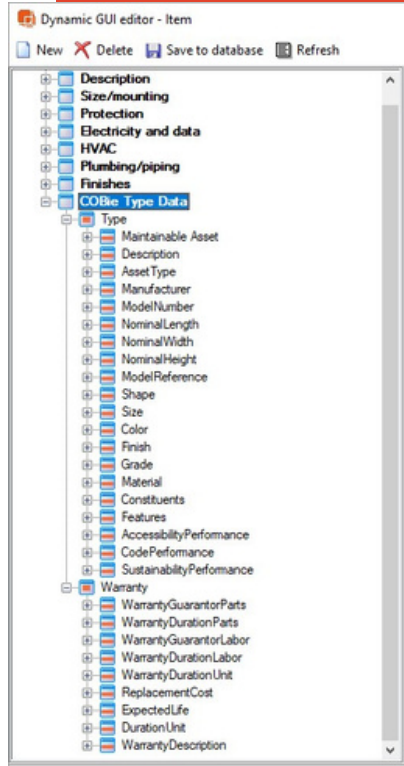
Ultimately the data is maintained in a single database, not federated models. It is connected to the models but only where we need to know information about the geometry in the models. And this is all in a structure ready for the COBie Excel export without any additional effort to create the export.



Most common connections between dRofus Modules and COBie Worksheets

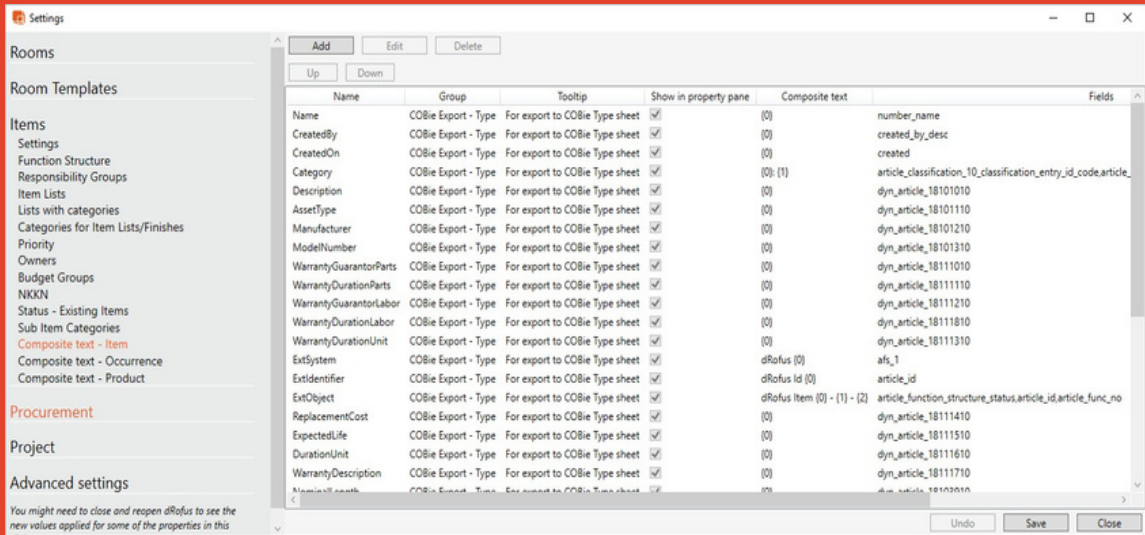
# Flexible Data Structure

At its core, dRofus contains attributes for all elements in a database. These attributes are organized and grouped for easy management as either Room Data, Item Data or Occurrence Data. Attributes can be defined and modified to suit project requirements. In some cases, out of the box attributes are used to populate COBie fields (such as Type.Name and Type.Category), in others they can be created as custom attributes.



For example, the image to the right shows how we define all the custom attributes that are required for Items to populate the COBie Type worksheet.

Once these attributes are defined in a database they are then mapped to Composite Text fields that are containers for reporting one or more attributes in a single export field. In the case of COBie, each Composite Text field represents a placeholder for data to go into a column in an Excel worksheet. We call this a “flexible” data structure because the connection between attributes and Composite Texts can be modified to suit a particular Building Owner’s needs.



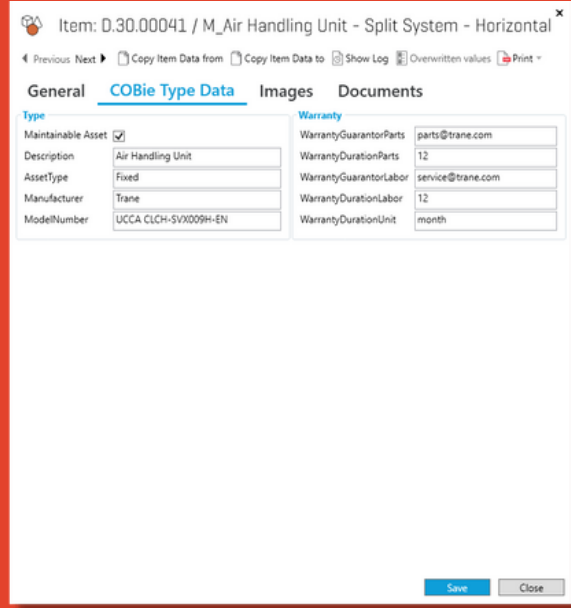
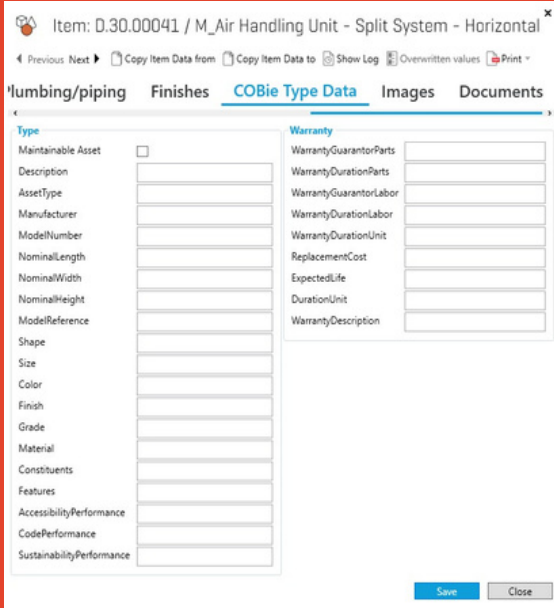
Example of Composite Text fields for Items that map to the COBie Type worksheet

At this point you're probably saying, "Sounds great but, man, that's a lot of stuff." It is. This is where the Building Owner's responsibility to define "how much stuff" becomes so important.

## Defining Building Owner Deliverable Requirements

Remember the Building Owner's responsibilities above? The need to clearly define which COBie Worksheets are required; the required data fields in each worksheet; the assets that need to be collected; and the sources for data? Once a dRofus database is configured to capture "all the COBie data" it is easy to customize.

Meet a specific Building Owner's needs. Through view filters and checkboxes it is easy to isolate only what a team needs to capture versus what a team could capture. Don't get me wrong, there is still a lot of work a team must do to actually input data either directly in dRofus, through our BIM connection or through Excel imports but that is just the nature of the process. What you will find is that there is a great deal of efficiency, automation and data integrity in collecting the information for a COBie deliverable in dRofus.



## BIM Connection

Once a dRofus database is configured to capture the required COBie data, the team can decide to what level the content in a model, such as Archicad or Revit, will be used to populate the data. Or for that matter, if a model is even needed. They can also decide which models and how much data is connected since a single dRofus database can be connected to multiple models. In each case, the team decides which dRofus attributes are connected to modeled content and where data is pushed to the models from dRofus, pushed to dRofus from the models or just managed in dRofus. The goal is to leverage as much modelled content as practical to maintain a connection to geometry without wholly relying on the model and to consolidate the COBie data in a single database.

# DEMYSTIFYING COBIE



The screenshot shows an Excel spreadsheet with columns for various COBie data points. Below the spreadsheet is the buildingSMART logo, which includes the text 'buildingSMART International Alliance for Interoperability' and 'COBie compliant deliverable'.

The screenshot shows the Revit software interface. On the left, the 'Properties' window displays details for 'M\_VAV Unit - Single Duct 150 mm'. The 'Mechanical Equipment (1)' section includes fields for 'Supply Air Pressure Drop' (54.70 Pa), 'Air Flow' (16.19 L/s), and 'Mark' (VAV-113). The 'Project Browser' shows a tree view of the project structure, including '1st Floor - Mech' and 'Basement - Mech'. On the right, the 'dRofus' window displays 'System Component D.30.00030/004 150 mm: VAV-113'. The 'COBie Export - Component' section lists fields such as 'Name', 'CreatedBy', 'CreatedOn', 'Space', 'Description', 'ExtSystem', 'ExtObject', 'ExtIdentifier', 'SerialNumber', 'InstallationDate', 'WarrantyStartDate', 'TagNumber', 'Barcode', and 'AssetIdentifier'. The 'COBie Export - System' section lists fields for 'Name', 'CreatedBy', 'CreatedOn', 'Category', 'ComponentName', and 'Description'. The background shows a 2D mechanical layout of a VAV unit.

Example of dRofus Occurrence containing COBie Component and System data connected to Revit Family

# The COBie Deliverable

Ideally there wouldn't be a COBie deliverable (sorry COBie) but rather a direct connection of the structured data to the CAFM/IWMS platforms (which can be done through our API). However, since we've done all the work to format the Excel export, let's do it.

## Excel Export

For the export, we basically map the Composite Text fields mentioned above to the associated columns in the various Excel worksheets in the COBie template. With all the connections in place, the team can run the export whenever they need to validate conformance or provide milestone deliverables such as COBie Design or COBie Construction deliverables.

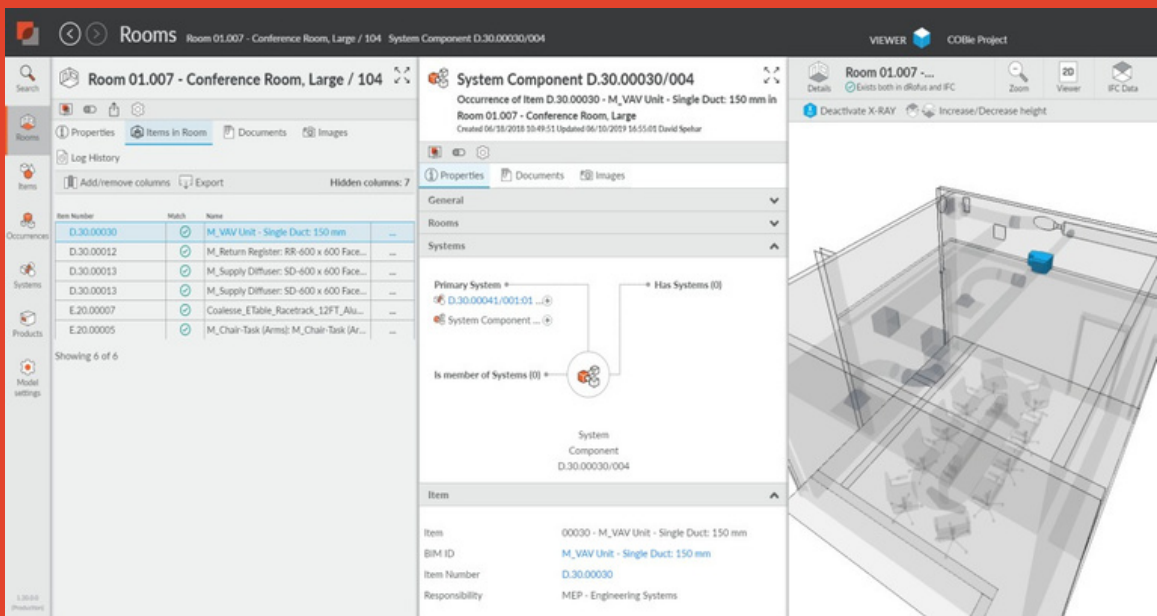
The diagram illustrates the workflow for creating COBie data structure and export settings. It consists of four main components:

- Item Type Data:** A screenshot of a software interface showing a table of item types with columns for Name, Description, and Manufacturer.
- Item Composite Text for Export:** A screenshot of a 'Settings' dialog box for 'Room Templates'. It lists various attributes (e.g., Name, Group, Facility, Manufacturer) and their corresponding export settings (e.g., 'For export to COBie Type sheet').
- Excel Export Map:** A screenshot of an 'Excel exports' dialog box. It shows a list of item types and their corresponding target sheets in the Excel template (e.g., 'COBie Export - Type' maps to 'Type').
- Excel Spreadsheet:** A screenshot of an Excel spreadsheet showing the resulting data structure. The columns include Name, Group, Facility, Floor, Space, Zone, Component, System, Assembly, Connection, Space, Resource, Job, and Date. The rows contain data for various item types, such as 'AC Unit Term 1', 'AC Unit Term 2', and 'AC Unit Term 3'.

Example of creating the data structure and export settings for Items into the Type worksheet

# IFC

And finally, all the COBie attributes managed in dRofus can be pushed into the Archicad or Revit models at project completion and exported as an ifc. Since the parameter data in the models will be the same as the source attributes in dRofus, the ifc can be linked back to dRofus with a complete picture of the project in an ifc viewer. Granted, most facility managers will take the tabular data view over the ifc, we think it looks way cooler this way.



# Conclusion

Make no mistake in understanding the effort a team must put forth to define, capture, manage and ultimately provide a COBie deliverable. While it can be a sizeable task the effort can be greatly reduced if treated as collaborative information management and not reactive data gathering. I hope this whitepaper has provided insight on some best practices for ensuring the entire team collaborates on an efficient and meaningful deliverable that serves the Building Owner for years to come. Since operating a building historically accounts for 80% of the total lifecycle cost, it's time and money well spent.



# References

If you're a glutton for punishment and this whitepaper wasn't enough, here are links to the referenced material listed above plus a couple more useful resources:

<http://www.infrastructure.alberta.ca/992.htm>

[https://www.nibs.org/page/bsa\\_cobie](https://www.nibs.org/page/bsa_cobie)

[https://www.nibs.org/page/bsa\\_infoexchange](https://www.nibs.org/page/bsa_infoexchange)

<https://www.prairieskyconsulting.com/>

<https://hub.a.a.ufl.edu/programs--courses/cobie-academy/>