MINIMALIST MODELLING AND CODING OF REVIT FOR QUANTITIES

Sr. TANG Ki-cheung

FHKIS RPS(QS) FSZCEA MHKIVM
Director of K C Tang Consultants Ltd.
Quantity Surveyors . Construction Cost and Contract Consultants Ltd.
Hong Kong, www.kctang.com.hk, kctang@kctang.com.hk
presented at
BIM Seminar Kuala Lumpur 2015 – 5D BIM for the Construction Industry
on 22 January 2015 at Pullman Hotel Kuala Lumpur, Malaysia
organized by
Royal Institution of Surveyors Malaysia

Biography

Mr. Tang is a qualified professional quantity surveyor. He has over 38 years' professional quantity surveying experience, handled over 740 No. projects in Hong Kong and Mainland China, facilitated or co-facilitated over 25 No. partnering / value management workshops, delivered speeches at over 35 No. professional and technical seminars, and pre-qualification structured training events, drafted the Standard Form of Contract for Maintenance and Renovation Works First Edition 2013, chaired a task force to draft the Standard Form of Domestic Sub-Contract since 2010, served as a council member of the Hong Kong Institute of Surveyors QS Division since 1998 and of the Hong Kong Institute of Value Management since 2010, chaired the Building Information Modelling Sub-Committee under HKIS's QS Divisional Council for years 2012 - 2014, and chaired the Organizing Committee for the International QS BIM Conference 2013 Hong Kong.

Abstract

One can model BIM to the last bolt and nut and can use third party software to take-off quantities from the BIM model, but this would mean extra work and more license fees. This presentation would show how a Revit model and its families of building blocks can be built, coded and scheduled in a minimalist and self-explanatory manner appropriate and sufficient enough for producing auditable dimensions and quantities for costing.

Keywords: coding, excel, qto, revit, schedules.

CAPABILITIES OF REVIT

Can Revit produce Bills of Quantities from Revit model with the press of a button? No, it is a modelling software, not a BQ production software.

Can Revit model provide quantities? Yes. Revit being a building information modelling ("*BIM*") software defines its building blocks ("*elements*") with *parameters* which contain a

lot of information including quantities of work and materials. That's why BIM is called "*parametric modelling*".

Are the quantities in compliance with the standard method of measurement? No, the SMMs of different countries vary.

Are the quantities provided by a Revit model sufficient and ready enough to be converted to quantities according to the standard method of measurement? No. Some of the parameters are not schedulable. The lengths, areas and volumes of elements provided by a Revit model may not be the desired lengths, areas and volumes for SMM purposes. Concrete shoulders at junctions of different mixes are not available. Formwork areas are not available. Wall and ceiling finish areas are not readily available.

Then, how can we use Revit model to produce quantities for estimates or BQ? There are third party quantity take-off software which can help extract quantities from a Revit model and provide them for estimates, bills of quantities and other uses. However, one would still need to write the descriptions and do some linking between the descriptions and the modelling elements. The linking resembles on-screen taking-off. Depending on the suitability of the model, the linking process may be quick or tedious. The beauty of such QTO software is that once the descriptions and linking are done, any changes to the models can be monitored. The downside is one has to invest in the license and training costs.

When descriptions are required to be written and quantities required to be classified according to the SMM, Quantity Surveyors are still required.

Can we still get something useful from Revit models without third party QTO software? Yes, it is the purpose of this paper to explain.

SHOULD QS BUILD MODELS?

Are Revit models readily available from Architects and Engineers? Probably not yet for most of the cases. While QTO software can handle 2D CAD drawings, the on-screen taking-off process would be more tedious for 2D CAD drawings.

Without a BIM model, should QS build up models?

QS has evolved from dimension sheet, cut-and-shuffle, schedule, scale rule, curvimeter, planimeter, Lotus 1-2-3 spreadsheet, Excel worksheet, digitizer, on-screen taking off, etc. There is no reason why the QTO must be textural and not graphical.

Having learned the basic tools, modelling the basic features with Revit is easier than with 2D CAD software. It should be within the capability of the average QS to learn, only if he or she has time.

For the bulk items like structure, fabric and finishes, the time to model and get quantities would be shorter than the time to do manual taking off or on-screen taking off from 2D CAD drawings.

Even if models are provided by Architects and Engineers, they may not have been modelled in such a way suitable for QTO. QS should be able to inspect and understand the models in

order to use the models. Furthermore, in order not to disturb the integrity of the models provided by the Architects and Engineers, QS would probably need his own set of "*QS parameters*" and adjust the parameters for his own purposes.

Therefore, the answer should be yes, QS should build up models in the absence of models or should be able to modify models to suit.

Should QS model every detail? No. Items like windows, doors, fittings and furniture vary in details between different projects and take some longer time to model. All the QS needs is the number for writing up full descriptions or measuring the component quantities per number with reference to the detailed drawings provided by the Architects and Engineers.

It should however be noted that the level of details of the models if provided by the Architects and Engineers should be no less than those traditionally provided for 2D CAD drawings for estimating, tendering and construction. This should be a simpler benchmark than whatever Level of Development or Level of Details (LOD) definitions.

HOW MUCH DO WE NEED TO MEASURE?

We may measure every bit of concrete, formwork, finishes, etc. exactly net when we measure the concrete members or room finishes. However, see the following table for the inter-relationship between dimensions and quantities:

Description	Primary Qty	Multiplier	Unit
Wall - grade C40 - T thick (centre line area)	A		Super
• Reinforced concrete Grade C40	A	T	Cube
• Wall formwork	A	2	Super
• To deduct junction with slab if wall measured through slab			
• To adjust for openings			
• To measure end of wall			
Column - grade C40 - W wide x D deep (height)	H		Run
• Reinforced concrete Grade C40	H	W x D	Cube
• Column formwork (junction with slab to deduct if measured through slab)	H	(W + D) x 2	Super
• To deduct junction with slab if column measured through slab			
Suspended beam - grade C30 - W wide x D deep - S slab (length)	L		Run
• Reinforced concrete Grade C30	L	W x (D – S)	Cube
• Beam formwork below slab	L	W + (D – S) x 2	Super
• Deduct slab formwork	L	W x -1	Super
• To adjust for shoulders to higher grade walls or columns			
Suspended slab - grade C30 - S thick (area)	A		Super
• Reinforced concrete Grade C30	A	S	Cube

Description	Primary Qty	Multiplier	Unit
• Slab formwork	A	2	Super
• To deduct junctions with walls, columns and beams			
• To adjust for openings			
Room - finishes group A - net plan area	A		Super
• Floor finishes	A		Super
• Ceiling finishes	A		Super
• To adjust for beam sides			
• To adjust for door opening			
• To adjust for work behind fixtures			
Room - finishes group A - perimeter including columns - H room height - S skirting height	L		Run
• Skirting	L		Run
• Wall finishes	L	H – S	Super
• To adjust for beam sides and ends			
• To adjust for openings			
• To adjust for work behind fixtures			
Window W1 - W wide x H high opening - T thick concrete wall - Room A	N		Nr
• Window W1, fully described	N		Nr
• Glazing	N	Detailed dimensions of W1	Super
• Deduct T wall concrete	N	W x H x T x -1	Cube
• Deduct wall formwork	N	W x H x 2	Super
• Add jambs and soffit formwork, T thick (or boxing number as appropriate)	N	W + H x 2	Run
• Deduct Room A wall finishes (assuming no dado)	N	W x H x -1	Super
• Add Room A wall finishes to window reveal	N	(W + H x 2) x reveal width	Run
• Add Room A window cill	N	W	Run
Door D1 - W wide x H high opening - T thick brickwall - Room finishes group A	N		Nr
• Door D1, fully described	N		Nr
• Deduct T thick brickwall	N	W x H x -1	Super
• Add lintol	N	W + end laps x 2	Run
• Deduct Room A skirting	N	W	Run
• Deduct Room A wall finishes	N	W x H x -1	Super
• Add Room A floor finishes	N	W x part of T as appropriate	Run
• Door frames, architraves, painting, dowels, holdfasts, etc.	N	Detailed dimensions of D1	As appropriate

The above shows that:

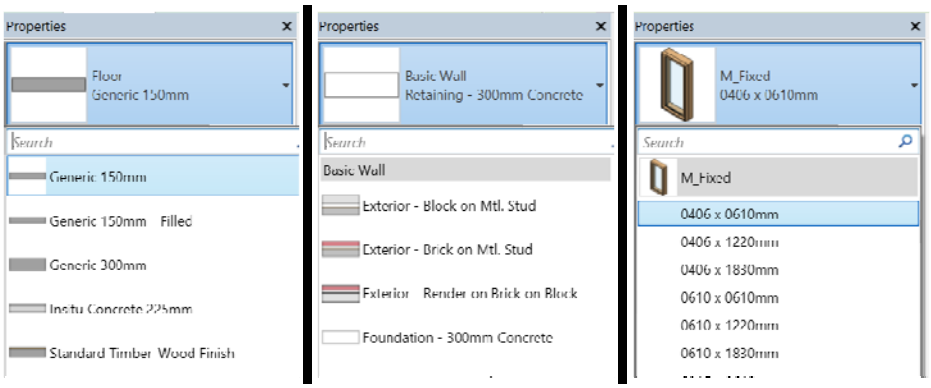
- We may measure the aggregate areas of different wall and slab thicknesses, aggregate lengths of different column and beam sections, aggregate plan areas and girths of rooms of the same finishes first (we may call these "primary quantities", those without bullets), before converting them into concrete, formwork and finishes quantities (we may call these "secondary quantities", those with bullets)
- We may measure the overall gross quantities first before making detailed adjustments
- Some adjustments can be made when we measure other elements, e.g. deduct slab formwork using beam quantities, adjust for wall openings when using window and door quantities, etc.
- The descriptions can be very short only to the extent of containing sufficient information for the purposes of the multiplier.

It follows that instead of expecting a Revit model to provide the quantity of every BQ item, we can just extract the primary quantities adequately described from the Revit model and use Excel to handle the secondary processing. The need for modelling to the last bolt and nut is reduced.

CODING AND WHY?

To convey the information from Revit to Excel, it would be good if the information is described in a concise, precise and consistent manner.

All Revit elements are classified and described by Family (Floor, Basic Wall, M_Fixed in the following images) and Types (those under the Search prompts in the images), but these descriptions may not contain the information the way we want and we would need to change them anyway.



Furthermore, if the model is provided by the Architect or Engineer and we do not want to change their descriptions of Families and Types, we may need to create a parameter to contain our the precise and consistent information we need.

The following coding ("*QS Desc*") should be sufficient and self-explanatory enough yet short and simple to represent the primary quantities in the above table:

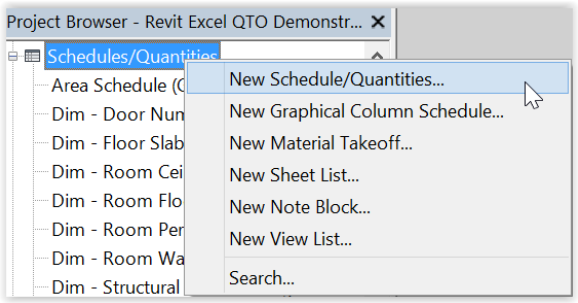
- WA-C40-100
- CL-C40-500x600
- BM-C30-300500 : 120SL
- SL-C30-120
- RM-F-A

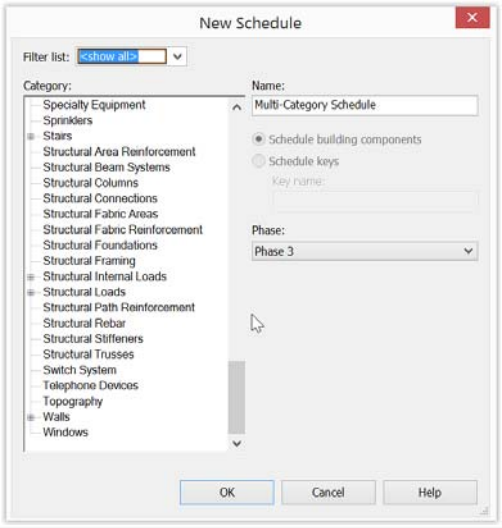
- RM-P-A
- WD-W1-1200x1500 : IWConc100-RoomA
- DR-D1-920x2200 : IWBrick125-RoomA

This way of coding should be more intuitive and understandable than those cryptic codes using (alpha) numeric codes.

SCHEDULES / QUANTITIES

Revit has the capabilities to produce many schedules, under the project browser:





<Dim - Wall Area>

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Level	Mark	Type	QS Tag	QS Qty	QS Unit	Family	Count	Length	Width	Area	Volume	QS Mean Area	Mean Area / Length
Level 1		IW-BLK-100		12.13	m2	Basic Wall	1	4550	100	12.13	1.213	12.13	2665
Level 1		EW-C20-125		9.69	m2	Basic Wall	1	4250	125	9.69	1.212	9.69	2281
Level 1		EW-C20-125		13.60	m2	Basic Wall	1	4400	125	13.60	1.700	13.60	3091
Level 1		EW-C20-125		13.60	m2	Basic Wall	1	4250	125	13.60	1.700	13.60	3200
Level 1		EW-C20-125		13.12	m2	Basic Wall	1	4100	125	13.12	1.640	13.12	3200
Level 1		EW-C20-125		12.41	m2	Basic Wall	1	4400	125	12.41	1.551	12.41	2819
Level 1		EW-C20-125		13.07	m2	Basic Wall	1	4550	125	13.07	1.634	13.07	2873
Level 1		EW-C20-125		12.83	m2	Basic Wall	1	4625	125	12.83	1.604	12.83	2774
Level 1		EW-C20-125		14.51	m2	Basic Wall	1	5150	125	14.51	1.814	14.51	2818
Level 1		EW-C20-125		14.30	m2	Basic Wall	1	5000	125	14.30	1.787	14.30	2859
Level 1	WA1	WA-C35-400	SL150-Ed	16.27	m2	Basic Wall	1	4850	400	16.27	6.278	15.70	3236
Level 1	WA2	WA-C35-300	SL150-Ed	19.25	m2	Basic Wall	1	4650	300	19.25	5.371	17.90	3850
Level 1	WA2A	WA-C35-300	SL150	2.77	m2	Basic Wall	1	658	300	2.77	0.658	2.19	3338
Level 1	WA3	WA-C35-300	SL150	17.71	m2	Basic Wall	1	4800	300	17.71	5.313	17.71	3690
Level 1	WA4	WA-C35-300	SL150-Ed	18.13	m2	Basic Wall	1	4700	300	18.13	5.094	16.98	3612
Level 1	WA5	WA-C35-300	SL150	17.30	m2	Basic Wall	1	4800	300	17.30	4.843	16.14	3363
Level 1	WA6	WA-C35-300	SL150	19.25	m2	Basic Wall	1	4788	300	19.25	5.429	18.10	3780
Level 1	WA7	WA-C35-300	SL150	19.25	m2	Basic Wall	1	4788	300	19.25	5.428	18.10	3780
Level 1	WA8	WA-C35-300	SL150-Ed	38.50	m2	Basic Wall	1	9875	300	38.50	11.550	38.50	3899
Grand total: 19				297.70			19	89183	4025	297.70	65.818	290.57	61128

With so many schedules to represent a modelled project, the choices of the columns in the schedules must be well co-ordinated. We cannot practically transfer the individual total quantity from the schedules manually one by one for billing purposes. This would be prone to errors.

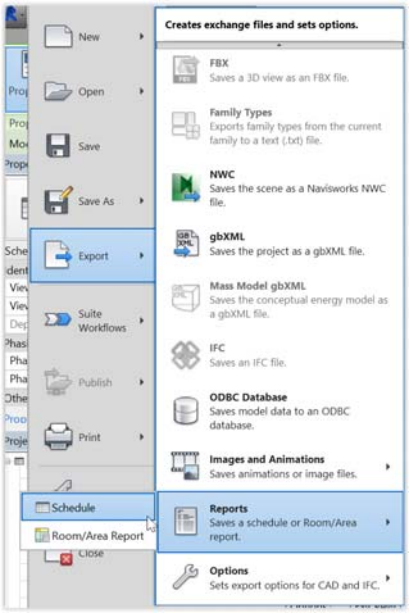
The above schedule has been designed such that:

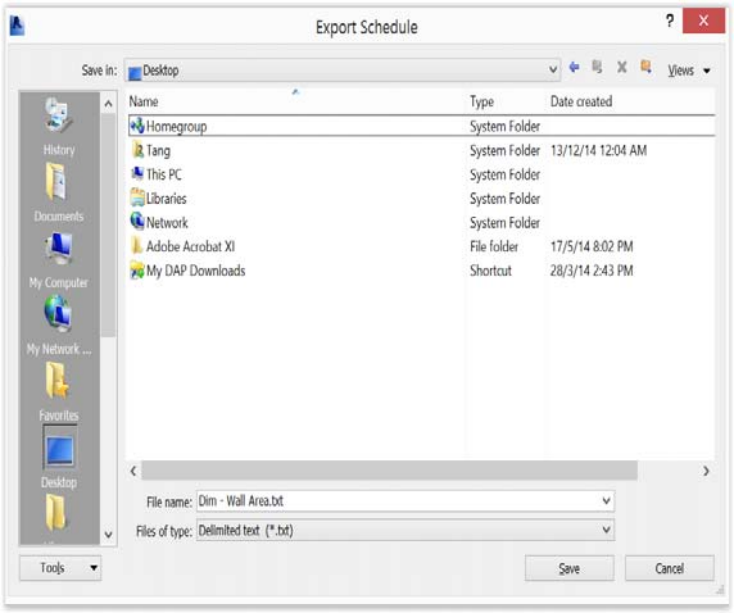
- The left 6 columns are consistently designed for all schedules
- The Level and Mark are for locational identification of the quantities, like our dimension sheets, for traceability
- Only the Type, QS Tag, QS Qty and QS Unit are really essential for billing
- The other columns are there for calculating the QS Qty or cross-checking, and can vary from schedules to schedules.

Revit can also export the schedules to a txt file which can then be imported by Excel to become a worksheet for use:

	A	B
1	Type : QS Tag : QS Unit	QS Qty
2	BM-C30-250x350 : SL150 : m	18.10
3	BM-C30-300x600 : SL150 : m	3.76
4	BM-C30-400x800 : I : m	17.00
5	BM-C30-400x800 : SL130 : m	12.75
6	BM-C30-400x800 : SL130-Edge : m	8.50
7	BM-C30-400x800 : SL150 : m	40.28
8	BM-C30-400x800 : SL150-Edge : m	18.50
9	CE-Plaster-Emulsion : : m2	228.22
10	CL-C40-450x600 : : m	4.00
11	CL-C40-450x600 : : Nr	1.00
12	CL-C40-450x600 : SL150 : m	8.00
13	CL-C40-450x600 : SL150 : Nr	2.00
14	CL-C40-450x600 : SL150-EdgeS : m	4.00
15	CL-C40-450x600 : SL150-EdgeS : Nr	1.00
16	CL-C40-600x750 : : m	8.00
17	CL-C40-600x750 : : Nr	2.00
18	CL-C40-600x750 : SL130-Corner : m	8.00
19	CL-C40-600x750 : SL130-Corner : Nr	2.00
20	CL-C40-600x750 : SL130-EdgeL : m	4.00
21	CL-C40-600x750 : SL130-EdgeL : Nr	1.00
22	CL-C40-600x750 : SL130-EdgeS : m	4.00
23	CL-C40-600x750 : SL130-EdgeS : Nr	1.00

QS Desc | All Dim | Area Schedule (Gross Bu



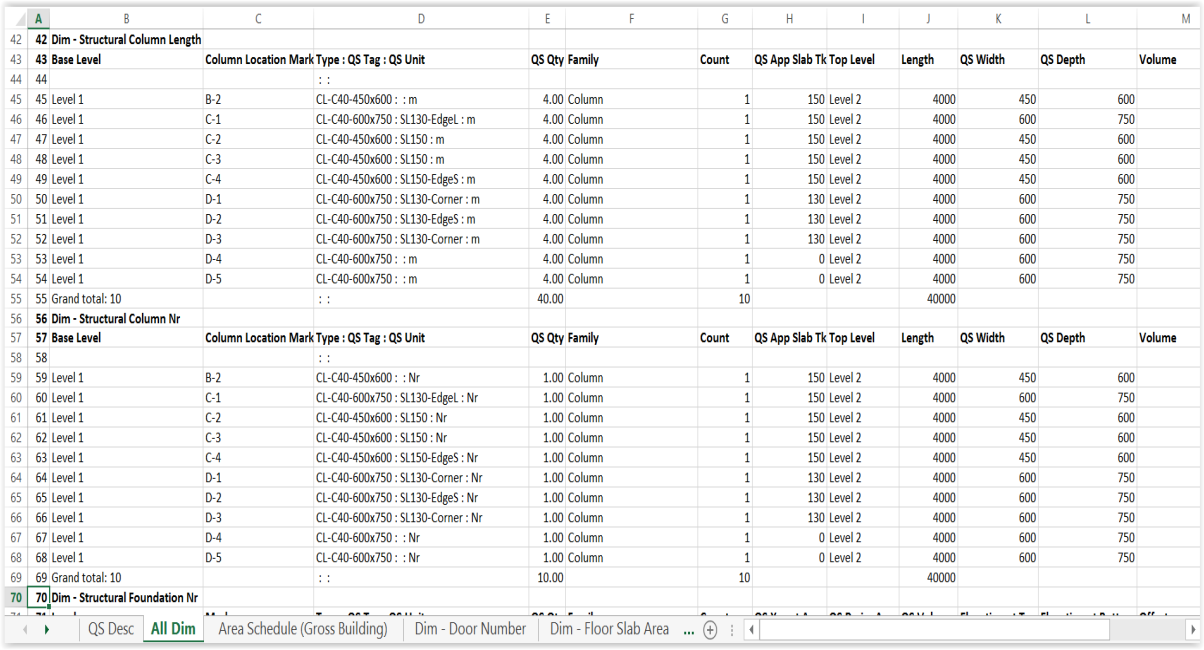


However, the export can only handle one schedule at a time. This would be a tedious process to export many schedules one at a time and merge all schedules together.

A macro (appended at the end of this paper) has been written whereby:

- All the schedules will be exported by a single command to one Excel file with one worksheet per schedule
- Schedules intended to provide the uniform left 6 columns for billing should have their names prefixed by “Dim – ” and the macro will combine the 3 columns of Type, QS Tag and QS Unit into a single “QS Desc” column in the style of “Type : QS Tag : QS Unit”
- An “All Dim” worksheet will be created to repeat all information contained in the “Dim – ” schedules
- A “QS Desc” worksheet will be created to contain a list of the unique QS Desc and their total QS Qty
- The total QS Qty will be useful for quick reference

- The Excel file name will contain the time down to the second it is created so that a re-export will not overwrite an existing Excel file.



	A	B	C	D	E	F	G	H	I	J	K	L	M
42	42	Dim - Structural Column Length											
43	43	Base Level	Column Location Mark	Type : QS Tag : QS Unit	QS Qty	Family	Count	QS App Slab Tk	Top Level	Length	QS Width	QS Depth	Volume
44	44			: :									
45	45	Level 1	B-2	CL-C40-450x600 : : m	4.00	Column	1	150	Level 2	4000	450	600	
46	46	Level 1	C-1	CL-C40-600x750 : SL130-EdgeL : m	4.00	Column	1	150	Level 2	4000	600	750	
47	47	Level 1	C-2	CL-C40-450x600 : SL150 : m	4.00	Column	1	150	Level 2	4000	450	600	
48	48	Level 1	C-3	CL-C40-450x600 : SL150 : m	4.00	Column	1	150	Level 2	4000	450	600	
49	49	Level 1	C-4	CL-C40-450x600 : SL150-EdgeS : m	4.00	Column	1	150	Level 2	4000	450	600	
50	50	Level 1	D-1	CL-C40-600x750 : SL130-Corner : m	4.00	Column	1	130	Level 2	4000	600	750	
51	51	Level 1	D-2	CL-C40-600x750 : SL130-EdgeS : m	4.00	Column	1	130	Level 2	4000	600	750	
52	52	Level 1	D-3	CL-C40-600x750 : SL130-Corner : m	4.00	Column	1	130	Level 2	4000	600	750	
53	53	Level 1	D-4	CL-C40-600x750 : : m	4.00	Column	1	0	Level 2	4000	600	750	
54	54	Level 1	D-5	CL-C40-600x750 : : m	4.00	Column	1	0	Level 2	4000	600	750	
55	55	Grand total: 10		: :	40.00		10			40000			
56	56	Dim - Structural Column Nr											
57	57	Base Level	Column Location Mark	Type : QS Tag : QS Unit	QS Qty	Family	Count	QS App Slab Tk	Top Level	Length	QS Width	QS Depth	Volume
58	58			: :									
59	59	Level 1	B-2	CL-C40-450x600 : : Nr	1.00	Column	1	150	Level 2	4000	450	600	
60	60	Level 1	C-1	CL-C40-600x750 : SL130-EdgeL : Nr	1.00	Column	1	150	Level 2	4000	600	750	
61	61	Level 1	C-2	CL-C40-450x600 : SL150 : Nr	1.00	Column	1	150	Level 2	4000	450	600	
62	62	Level 1	C-3	CL-C40-450x600 : SL150 : Nr	1.00	Column	1	150	Level 2	4000	450	600	
63	63	Level 1	C-4	CL-C40-450x600 : SL150-EdgeS : Nr	1.00	Column	1	150	Level 2	4000	450	600	
64	64	Level 1	D-1	CL-C40-600x750 : SL130-Corner : Nr	1.00	Column	1	130	Level 2	4000	600	750	
65	65	Level 1	D-2	CL-C40-600x750 : SL130-EdgeS : Nr	1.00	Column	1	130	Level 2	4000	600	750	
66	66	Level 1	D-3	CL-C40-600x750 : SL130-Corner : Nr	1.00	Column	1	130	Level 2	4000	600	750	
67	67	Level 1	D-4	CL-C40-600x750 : : Nr	1.00	Column	1	0	Level 2	4000	600	750	
68	68	Level 1	D-5	CL-C40-600x750 : : Nr	1.00	Column	1	0	Level 2	4000	600	750	
69	69	Grand total: 10		: :	10.00		10			40000			
70	70	Dim - Structural Foundation Nr											

QS Desc | All Dim | Area Schedule (Gross Building) | Dim - Door Number | Dim - Floor Slab Area ...

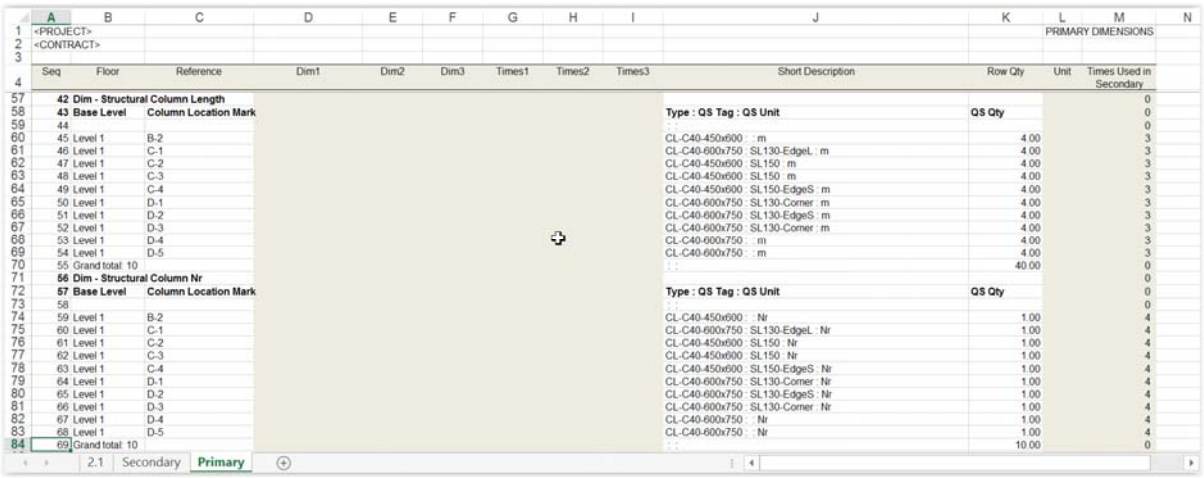
BILLING WORKFLOW

The primary quantities in the "All Dim" worksheet can be further processed for billing. The workflow are as follows:

- Data → (extraction from Revit schedules) → Primary Qty → (processing) → Secondary Qty → (processing) → Estimate or BQ
- Data → (direct measurement without using Revit) → Primary Qty → (processing) → Secondary Qty → (processing) → Estimate or BQ

PRIMARY WORKSHEET

The first 5 columns only of the "All Dim" worksheet shown above are copied to the lower region of a "Primary worksheet":

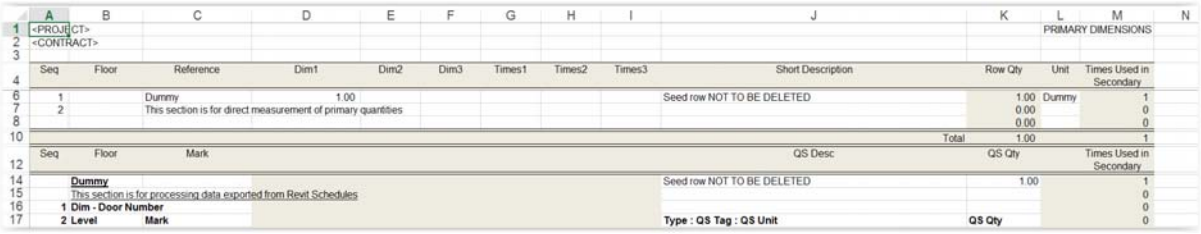


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	<PROJECT>												PRIMARY DIMENSIONS	
2	<CONTRACT>													
3														
4	Seq	Floor	Reference	Dim1	Dim2	Dim3	Times1	Times2	Times3	Short Description	Row Qty	Unit	Times Used in Secondary	
57	42	Dim - Structural Column Length											0	
58	43	Base Level	Column Location Mark							Type : QS Tag : QS Unit	QS Qty		0	
59	44									: :			0	
60	45	Level 1	B-2							CL-C40-450x600 : : m	4.00		3	
61	46	Level 1	C-1							CL-C40-600x750 : SL130-EdgeL : m	4.00		3	
62	47	Level 1	C-2							CL-C40-450x600 : SL150 : m	4.00		3	
63	48	Level 1	C-3							CL-C40-450x600 : SL150 : m	4.00		3	
64	49	Level 1	C-4							CL-C40-450x600 : SL150-EdgeS : m	4.00		3	
65	50	Level 1	D-1							CL-C40-600x750 : SL130-Corner : m	4.00		3	
66	51	Level 1	D-2							CL-C40-600x750 : SL130-EdgeS : m	4.00		3	
67	52	Level 1	D-3							CL-C40-600x750 : SL130-Corner : m	4.00		3	
68	53	Level 1	D-4							CL-C40-600x750 : : m	4.00		3	
69	54	Level 1	D-5							CL-C40-600x750 : : m	4.00		3	
70	55	Grand total: 10								: :	40.00		0	
71	56	Dim - Structural Column Nr											0	
72	57	Base Level	Column Location Mark							Type : QS Tag : QS Unit	QS Qty		0	
73	58									: :			0	
74	59	Level 1	B-2							CL-C40-450x600 : : Nr	1.00		4	
75	60	Level 1	C-1							CL-C40-600x750 : SL130-EdgeL : Nr	1.00		4	
76	61	Level 1	C-2							CL-C40-450x600 : SL150 : Nr	1.00		4	
77	62	Level 1	C-3							CL-C40-450x600 : SL150 : Nr	1.00		4	
78	63	Level 1	C-4							CL-C40-450x600 : SL150-EdgeS : Nr	1.00		4	
79	64	Level 1	D-1							CL-C40-600x750 : SL130-Corner : Nr	1.00		4	
80	65	Level 1	D-2							CL-C40-600x750 : SL130-EdgeS : Nr	1.00		4	
81	66	Level 1	D-3							CL-C40-600x750 : SL130-Corner : Nr	1.00		4	
82	67	Level 1	D-4							CL-C40-600x750 : : Nr	1.00		4	
83	68	Level 1	D-5							CL-C40-600x750 : : Nr	1.00		4	
84	69	Grand total: 10								: :	10.00		0	

2.1 | Secondary | Primary

The item sequence can be kept unchanged for easy referencing, or may be sorted temporarily. The Seq column helps re-sort them back to the original sequence.

Highlighted columns are not used for data exported from Revit Schedules because they are reserved for direct measurement at the upper region of the worksheet, where the formula used for Row Qty, say at Row 6, is =PRODUCT(E6:J6), meaning product of Dim1 …. Times3.



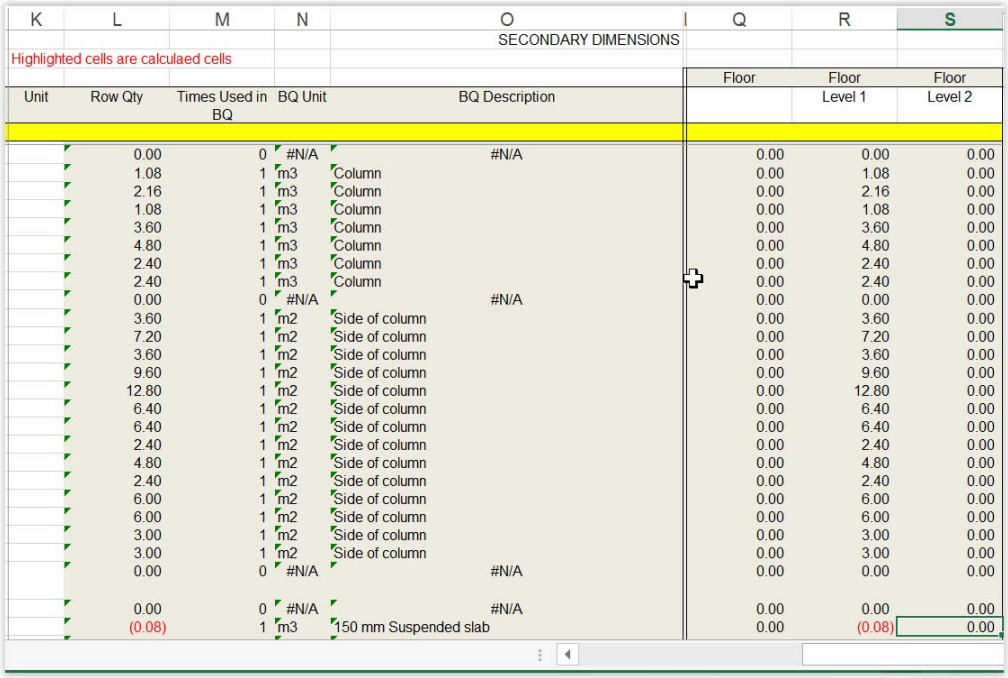
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	<PROJECT>												PRIMARY DIMENSIONS	
2	<CONTRACT>													
3														
4	Seq	Floor	Reference	Dim1	Dim2	Dim3	Times1	Times2	Times3	Short Description	Row Qty	Unit	Times Used in Secondary	
6	1		Dummy	1.00						Seed row NOT TO BE DELETED	1.00	Dummy	1	
7	2		This section is for direct measurement of primary quantities								0.00		0	
8											0.00		0	
10											Total 1.00		1	
12	Seq	Floor	Mark							QS Desc	QS Qty		Times Used in Secondary	
14		Dummy								Seed row NOT TO BE DELETED	1.00		1	
15		This section is for processing data exported from Revit Schedules											0	
16	1	Dim - Door Number											0	
17	2	Level	Mark							Type : QS Tag : QS Unit	QS Qty		0	

SECONDARY WORKSHEET

Column A only of the "QS Desc" worksheet shown above is copied to Column B in the lower region of a "Secondary worksheet":

	A	B	C	D	E	F	G	H	I	J
1	<PROJECT>									
2	<CONTRACT>									
3										
4	Seq	Floor	Reference	Dim1	Dim2	Dim3	Times1	Times2	Times3	Code
5			Short Description in Primary Sheet	Primary Qty			<---- headings for the lower region, if different ---->			
50	37		**Column concrete**							
51	38		CL-C40-450x600 : : m	4.00	0.45	0.60				CL-C40
52	39		CL-C40-450x600 : SL150 : m	8.00	0.45	0.60				CL-C40
53	40		CL-C40-450x600 : SL150-EdgeS : m	4.00	0.45	0.60				CL-C40
54	41		CL-C40-600x750 : : m	8.00	0.60	0.75				CL-C40
55	42		CL-C40-600x750 : SL130-Corner : m	8.00	0.80	0.75				CL-C40
56	43		CL-C40-600x750 : SL130-EdgeL : m	4.00	0.80	0.75				CL-C40
57	44		CL-C40-600x750 : SL130-EdgeS : m	4.00	0.80	0.75				CL-C40
58	45		**Column formwork**	0.00						
59	46		CL-C40-450x600 : : m	4.00	0.45		2.00			CL-FWK
60	47		CL-C40-450x600 : SL150 : m	8.00	0.45		2.00			CL-FWK
61	48		CL-C40-450x600 : SL150-EdgeS : m	4.00	0.45		2.00			CL-FWK
62	49		CL-C40-600x750 : : m	8.00	0.60		2.00			CL-FWK
63	50		CL-C40-600x750 : SL130-Corner : m	8.00	0.80		2.00			CL-FWK
64	51		CL-C40-600x750 : SL130-EdgeL : m	4.00	0.80		2.00			CL-FWK
65	52		CL-C40-600x750 : SL130-EdgeS : m	4.00	0.80		2.00			CL-FWK
66	53		CL-C40-450x600 : : m	4.00		0.60				CL-FWK
67	54		CL-C40-450x600 : SL150 : m	8.00		0.60				CL-FWK
68	55		CL-C40-450x600 : SL150-EdgeS : m	4.00		0.60				CL-FWK
69	56		CL-C40-600x750 : : m	8.00		0.75				CL-FWK
70	57		CL-C40-600x750 : SL130-Corner : m	8.00		0.75				CL-FWK
71	58		CL-C40-600x750 : SL130-EdgeL : m	4.00		0.75				CL-FWK
72	59		CL-C40-600x750 : SL130-EdgeS : m	4.00		0.75				CL-FWK
73	60		**Column / slab junction : deduct slab concrete :**	0.00						
74	61		CL-C40-450x600 : : Nr	1.00	0.45	0.60	0.00		0.00	
75	62		CL-C40-450x600 : SL150 : Nr	2.00	0.45	0.60	0.15		(1.00)	SL-C30-150

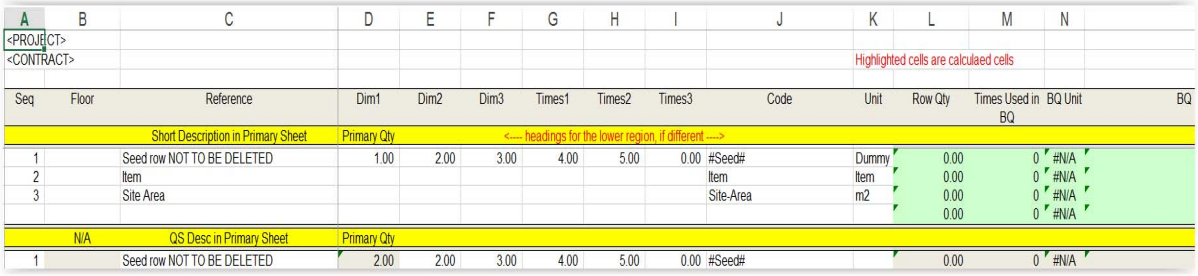
2.1 | Secondary | Primary



K	L	M	N	O	Q	R	S
				SECONDARY DIMENSIONS			
Highlighted cells are calculaed cells							
					Floor	Floor	Floor
Unit	Row Qty	Times Used in BQ	BQ Unit	BQ Description		Level 1	Level 2
	0.00	0	#N/A	#N/A	0.00	0.00	0.00
	1.08	1	m3	Column	0.00	1.08	0.00
	2.16	1	m3	Column	0.00	2.16	0.00
	1.08	1	m3	Column	0.00	1.08	0.00
	3.60	1	m3	Column	0.00	3.60	0.00
	4.80	1	m3	Column	0.00	4.80	0.00
	2.40	1	m3	Column	0.00	2.40	0.00
	2.40	1	m3	Column	0.00	2.40	0.00
	0.00	0	#N/A	#N/A	0.00	0.00	0.00
	3.60	1	m2	Side of column	0.00	3.60	0.00
	7.20	1	m2	Side of column	0.00	7.20	0.00
	3.60	1	m2	Side of column	0.00	3.60	0.00
	9.60	1	m2	Side of column	0.00	9.60	0.00
	12.80	1	m2	Side of column	0.00	12.80	0.00
	6.40	1	m2	Side of column	0.00	6.40	0.00
	6.40	1	m2	Side of column	0.00	6.40	0.00
	2.40	1	m2	Side of column	0.00	2.40	0.00
	4.80	1	m2	Side of column	0.00	4.80	0.00
	2.40	1	m2	Side of column	0.00	2.40	0.00
	6.00	1	m2	Side of column	0.00	6.00	0.00
	6.00	1	m2	Side of column	0.00	6.00	0.00
	3.00	1	m2	Side of column	0.00	3.00	0.00
	3.00	1	m2	Side of column	0.00	3.00	0.00
	0.00	0	#N/A	#N/A	0.00	0.00	0.00
	0.00	0	#N/A	#N/A	0.00	0.00	0.00
	(0.08)	1	m3	150 mm Suspended slab	0.00	(0.08)	0.00

right portion

The upper region of the Secondary worksheet is similar to the upper region of the Primary worksheet for direct measurement which can straightly go to the Estimate or BQ without further processing like the lower region.



A	B	C	D	E	F	G	H	I	J	K	L	M	N
<PROJECT>													
<CONTRACT>										Highlighted cells are calculaed cells			
Seq	Floor	Reference	Dim1	Dim2	Dim3	Times1	Times2	Times3	Code	Unit	Row Qty	Times Used in BQ	BQ Unit
		Short Description in Primary Sheet	Primary Qty			<---- headings for the lower region, if different ---->							BQ
1		Seed row NOT TO BE DELETED	1.00	2.00	3.00	4.00	5.00	0.00	#Seed#	Dummy	0.00	0	#N/A
2		Item							Item	Item	0.00	0	#N/A
3		Site Area							Site-Area	m2	0.00	0	#N/A
											0.00	0	#N/A
	N/A	QS Desc in Primary Sheet	Primary Qty										
1		Seed row NOT TO BE DELETED	2.00	2.00	3.00	4.00	5.00	0.00	#Seed#		0.00	0	#N/A

Further processing of the Primary Qty is done in the lower region.

By using a special formula in Column D in the lower region, each of the Primary Qty here is the total of the Primary Qty of the same QS Desc in the Primary worksheet. There is no need to sort and group the lines in the Primary worksheet in order to give group total per each QS Desc.

The special formula is actully very simple. For example, the Primary Qty at Row 14 is

=SUMIF(Primary!$J$5:$J$181;$C14;Primary!$K$5:$K$181)

where $C14 is the QS Desc on the same row, Primary!$K$5:$K$181 is the range of QS Desc in the Primary worksheet, and Primary!$J$5:$J$181 is the range of Row Qty in the Primary worksheet. If the QS Desc in the Primary worksheet matches the QS Desc in the Secondary worksheet, then add in the corresponding Row Qty in the Primary worksheet.

The Code will be the code of the Estimate or BQ items.

Again, the Row Qty is the product of all Dims and Times of the same row, using the "=Product()" function. They are also called Secondary Qty.

The columns like Times Used in BQ, BQ Unit, BQ Descriptions are making referece to the Estimate or BQ worksheet for error checking.

The Floor columns are for analysis of quantities by floors.

ESTIMATE OR BQ WORKSHEET

The Estimate or BQ worksheet is like this:

A	B	C	D	E	F	G	H	I	J	K	L
<PROJECT>						BILLS OF QUANTITIES					
<CONTRACT>			Short descriptions for illustration only			BILL NO. 2 - BUILDING			<Internal reference>		
						2.1 - ALL TRADES			Floor	Floor	Floor
	Item	Code	Description	Qty	Unit	Rate	HK$			Level 1	Level 2
2.1/	3		BEAMS								
2.1/	3.1		Reinforced concrete 30MPa in								
2.1/	3.1.1	BM-C30	Suspended beam	28	m3						28
2.1/	3.2		Formwork to								
2.1/	3.2.1	BM-FWK	Sides and soffit of suspended beam	179	m2						179
2.1/	3.2.2	BM-FWK-3.5-5	Sides and soffit of suspended beam; strutting 3.5 - 5.0 high	12	m2						12
2.1/	4		SUSPENDED SLABS								
2.1/	4.1		Reinforced concrete 30MPa in								
2.1/	4.1.1	SL-C30-130	130 mm Suspended slab	6	m3						7
2.1/	4.1.2	SL-C30-150	150 mm Suspended slab	30	m3						30

Similar to the Primary Qty or the Secondary worksheet, by using a special formula, each of the Qty here is the total of the Secondary Qty of the same Code in the Secondary worksheet.

The special formula is in the form of

=ROUND(SUMIF(Secondary!$J$6:$J$176;$C41;Secondary!$L$6:$L$176);0)

Basically, it means that if the Code in the Secondary worksheet matches the Code in the Estimate or BQ worksheet, then add in the corresponding Row Qty in the Secondary worksheet.

The Floor columns are for internal references only.

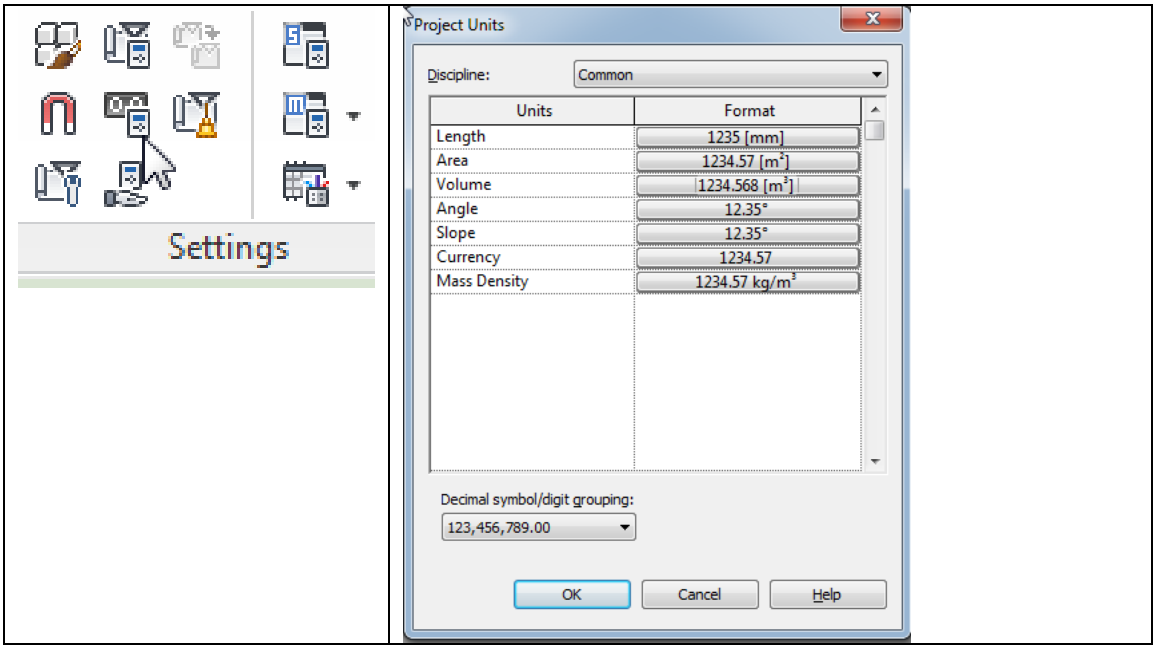
When issuing the Estimate or BQ in Excel softcopy, the formulae should be changed to values and other internal reference data should be removed with the empty columns hidden.

PROJECT UNITS

Revit schedules show the units against numerical values by default. It would not be convenient if the numerical values are exported to Excel worksheet for further calculations because they would not be recognised as numerical values unless the units are removed

A solution would be to define two decimal places for Area and three decimal places for Volume with the units hidden, so as to make them self-explanatory without the need of units.

To define the project units (in metric): click Manage > Project Unit icon



PARAMETERS

Parameters provided by Revit are called system parameters, which cannot be changed though some permit entry of values.

There are two types of parameters which one can define at the Family, Type or Instance level:

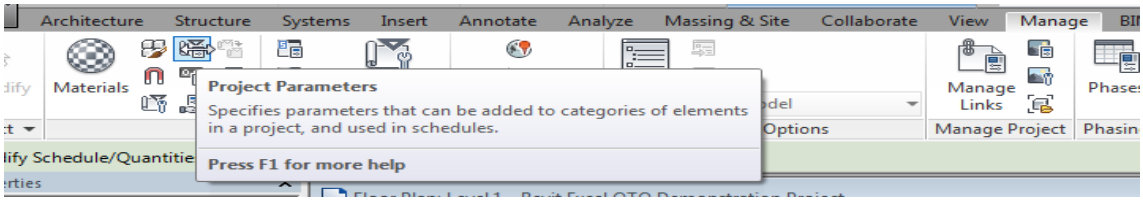
- Project Parameters: can appear in schedules but not in tags, but cannot be shared by other projects and families
- Shared Parameters: can appear in schedules and tags, shared by multiple projects and families, and exported to ODBC.

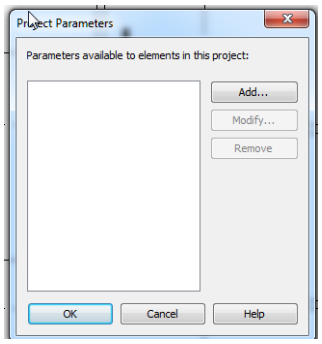
Shared Parameters are more versatile and useful.

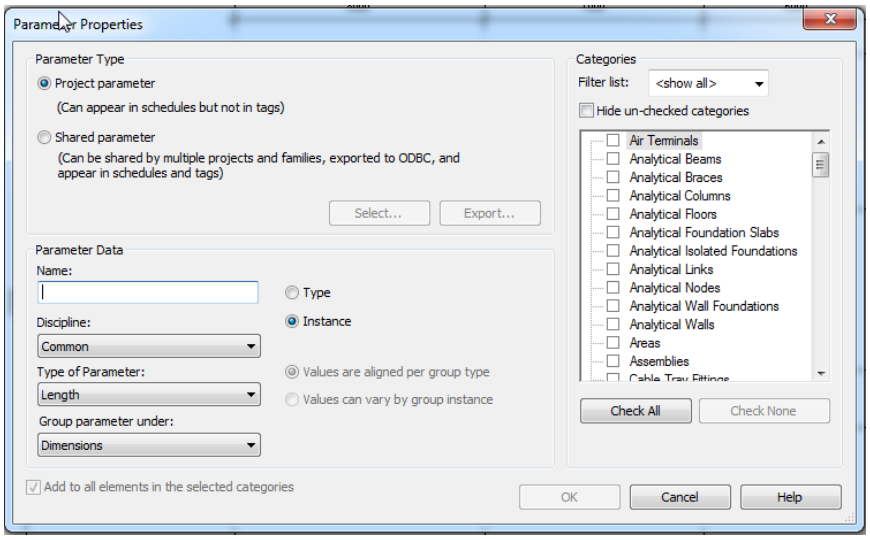
For some Families and Types, a shared parameter can be added to the properties directly. However, for others, only project parameters can be added to the properties, but a project parameter can borrow a shared parameter, so the shared parameter can still be used but indirectly.

PROJECT PARAMETERS

To add project parameters: select Manage > Settings > Project Parameters.



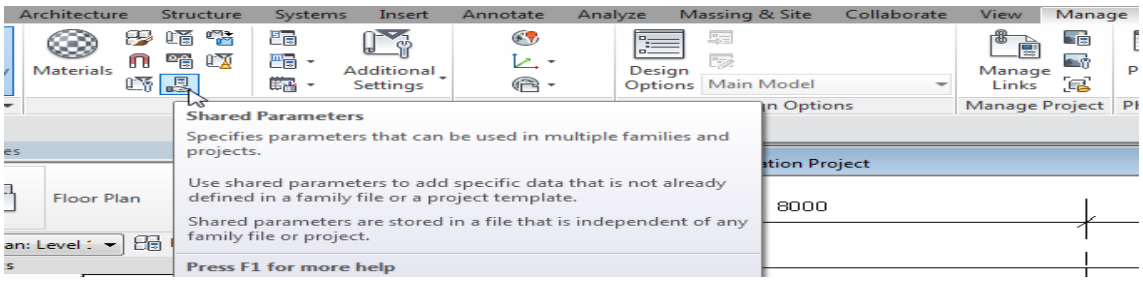




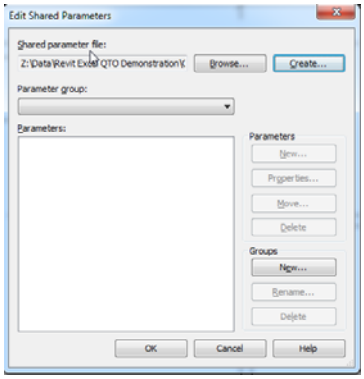
Note from the above that either a project parameter or a shared parameter can be added.

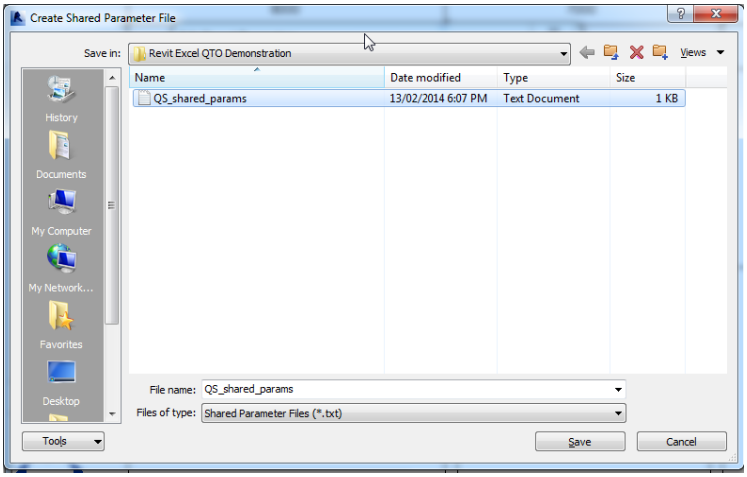
SHARED PARAMETERS

To add shared parameters: select Manage > Settings > Shared Parameters.



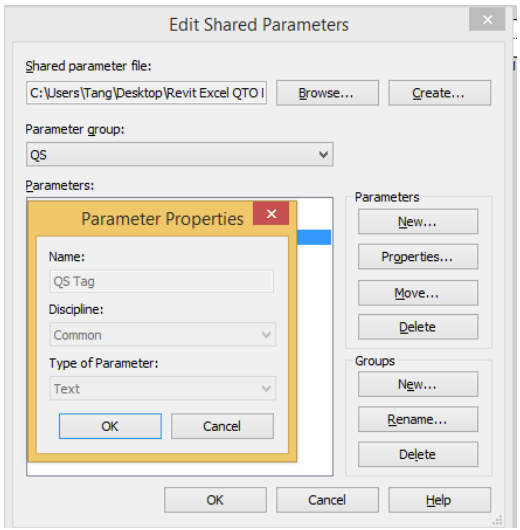
Select Create, go to a convenient folder, name a Shared Parameter file, e.g. "QS_shared_params", which is a txt file, save and return back to the screen on the left.





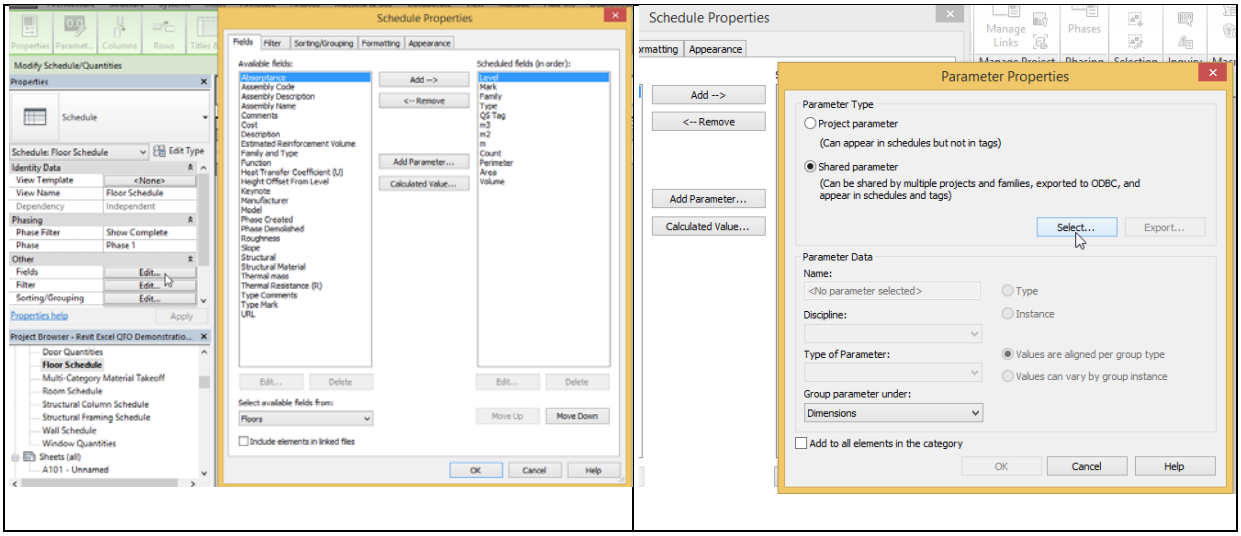
Select New under Group, name a New Parameter Group, e.g. "QS", and select OK.

Select New under Parameters, name a new parameter, e.g. “QS Tag”, select Common under Discipline, select Text under Type of Parameter, and select OK.



DEFINING SCHEDULE COLUMNS

Revit schedule columns (fields) can be selected from “Available fields” (parameters). If there are no suitable fields available, new columns can be defined either by “Add Parameter” or “Calculated Value”.



LEVEL AND MARK

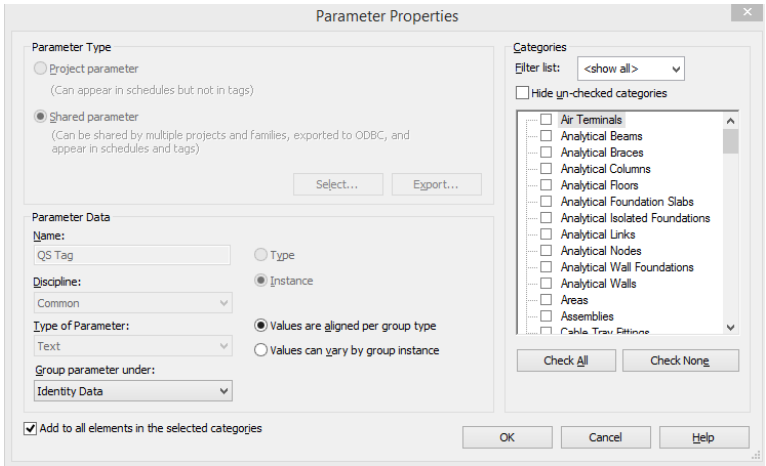
Level and Mark parameters are generally available with elements.

TYPE

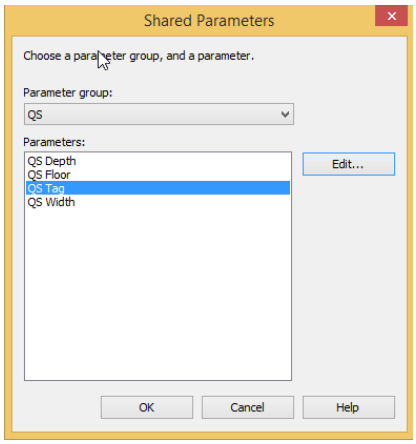
Elements are classified by Family and generally have a Type parameter available for use in the schedules. However, instead of using Family and Type as provided by Revit, short code like descriptions are used for Type. To simplify matters, such descriptions should represent both the information of Family and Type.

QS TAG

QS Tag is a new parameter specially added to supplement the information of "Type". Select Add Parameter as shown above to open the Parameter Properties window. Select Shared parameter > Select to open another window. Select the Parameter Group QS if it already exists, otherwise create it as described for Shared Parameters. Select QS Tag if it already exists, otherwise create it as described for Shared Parameters by selecting Edit. Select OK to go back to the Parameter Properties window.

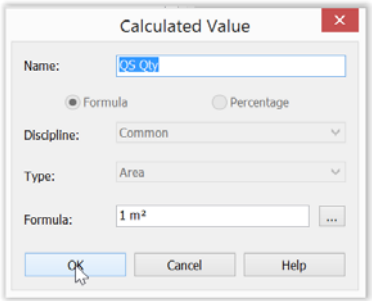


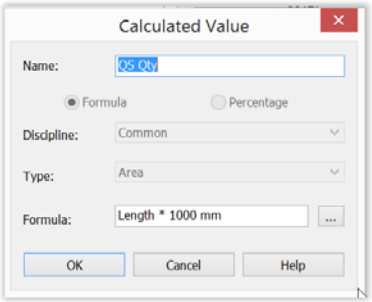
Check “Add to all elements in the selected categories” and Instance. Select Identity Data under “Group parameter under”, and select OK. Move up the newly created parameter on the menu to the desired position.



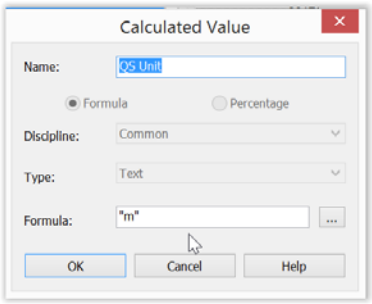
QS QTY AND QS UNIT

QS Qty is a calculated field borrowing its value from other parameters like Count, Length, Area, Volume, etc. The unit of QS Qty can be Nr, m, m2, m3, i.e. a mixture. However, Revit does not permit a column of such mixed nature. Since QS Qty should have 2 decimal places, therefore, it is artificially defined as an Area parameter which according to the setting of the Project Units described earlier will show 2 decimal places. However, tricks have to be done as shown below to make their numerical values still correct even though they are recognised by Revit as Area parameters:





QS Unit is a calculated field specially used to tell the real unit of QS Qty. The formulae are simply “m” for “m”, “m2” for “m2”, “m3” for “m3”, “Nr” for “Nr”, etc.



It may be possible for some elements that we need their Count, Length, Area as QS Qty, e.g. Column Nr and Column Length, Wall Area and Wall Length. To handle this, the same

schedule can be duplicated only with the QS Qty and QS Unit suitably adjusted to suit while all other fields can be kept the same

WALL SCHEDULES

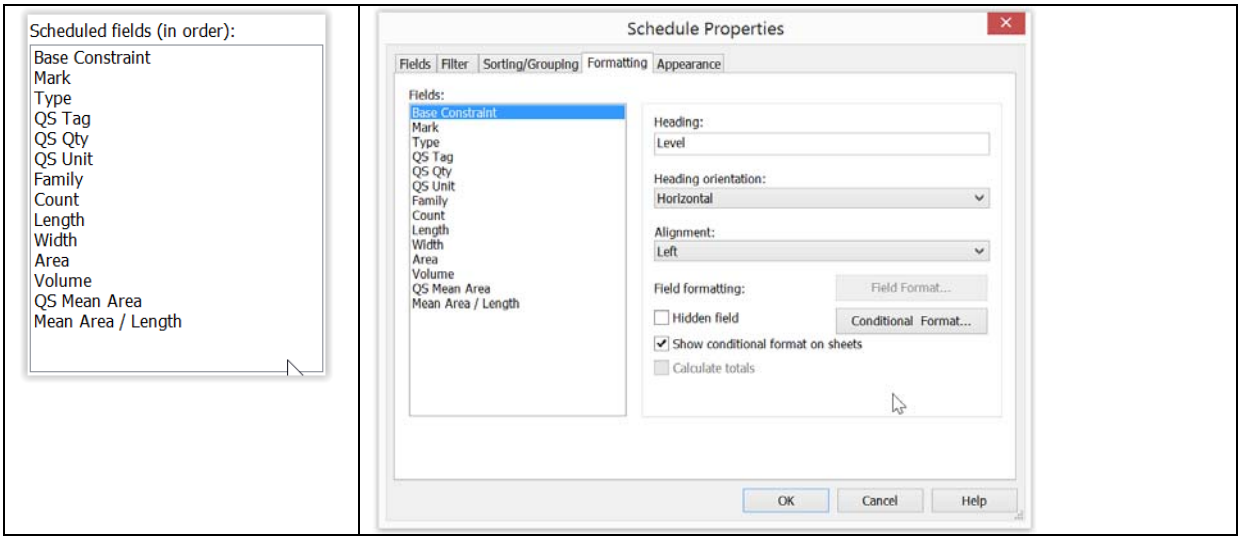
A Wall Area schedule gives wall areas as QS Qty for the generation of concrete volume and formwork area:

A	B	C	D	E	F	G	H	I	J	K	L	M
1	Dim - Wall Area											
2	Level	Mark	Type : QS Tag : QS Unit	QS Qty	Family	Count	Length	Width	Area	Volume	QS Mean Area	Mean Area / Length
3			: :									
4	Level 1		IW-BLK-100 : : m2	12.13	Basic Wall	1	4550	100	12.13	1.213	12.13	2665
5	Level 1		EW-C20-125 : : m2	9.69	Basic Wall	1	4250	125	9.69	1.212	9.69	2281
6	Level 1		EW-C20-125 : : m2	13.60	Basic Wall	1	4400	125	13.60	1.700	13.60	3091
7	Level 1		EW-C20-125 : : m2	13.60	Basic Wall	1	4250	125	13.60	1.700	13.60	3200
8	Level 1		EW-C20-125 : : m2	13.12	Basic Wall	1	4100	125	13.12	1.640	13.12	3200

A Wall Length schedule gives wall lengths as QS Qty for the processing of quantities at junctions with slabs:

A	B	C	D	E	F	G	H	I	J	K	L	M
1	Dim - Wall Length											
2	Level	Mark	Type : QS Tag : QS Unit	QS Qty	Family	Count	Length	Width	Area	Volume	QS Mean Area	Mean Area / Length
3			: :									
4	Level 1		IW-BLK-100 : : m	4.55	Basic Wall	1	4550	100	12.13	1.213	12.13	2665
5	Level 1		EW-C20-125 : : m	4.25	Basic Wall	1	4250	125	9.69	1.212	9.69	2281
6	Level 1		EW-C20-125 : : m	4.40	Basic Wall	1	4400	125	13.60	1.700	13.60	3091
7	Level 1		EW-C20-125 : : m	4.25	Basic Wall	1	4250	125	13.60	1.700	13.60	3200
8	Level 1		EW-C20-125 : : m	4.10	Basic Wall	1	4100	125	13.12	1.640	13.12	3200
9	Level 1		EW-C20-125 : : m	4.40	Basic Wall	1	4400	125	12.41	1.551	12.41	2819
10	Level 1		EW-C20-125 : : m	4.55	Basic Wall	1	4550	125	13.07	1.634	13.07	2873
11	Level 1		EW-C20-125 : : m	4.63	Basic Wall	1	4625	125	12.83	1.604	12.83	2774
12	Level 1		EW-C20-125 : : m	5.15	Basic Wall	1	5150	125	14.51	1.814	14.51	2818
13	Level 1		EW-C20-125 : : m	5.00	Basic Wall	1	5000	125	14.30	1.787	14.30	2859
14	Level 1	WA1	WA-C35-400 : SL150-Edge : m	4.85	Basic Wall	1	4850	400	16.27	6.278	15.70	3236
15	Level 1	WA2	WA-C35-300 : SL150-Edge : m	4.65	Basic Wall	1	4650	300	19.25	5.371	17.90	3850
16	Level 1	WA2A	WA-C35-300 : SL150 : m	0.66	Basic Wall	1	658	300	2.77	0.658	2.19	3338
17	Level 1	WA3	WA-C35-300 : SL150 : m	4.80	Basic Wall	1	4800	300	17.71	5.313	17.71	3690
18	Level 1	WA4	WA-C35-300 : SL150-Edge : m	4.70	Basic Wall	1	4700	300	18.13	5.094	16.98	3612
19	Level 1	WA5	WA-C35-300 : SL150 : m	4.80	Basic Wall	1	4800	300	17.30	4.843	16.14	3363
20	Level 1	WA6	WA-C35-300 : SL150 : m	4.79	Basic Wall	1	4788	300	19.25	5.429	18.10	3780
21	Level 1	WA7	WA-C35-300 : SL150 : m	4.79	Basic Wall	1	4788	300	19.25	5.428	18.10	3780
22	Level 1	WA8	WA-C35-300 : SL150-Edge : m	9.88	Basic Wall	1	9875	300	38.50	11.550	38.50	3899

Level is actually the base constraint renamed in the schedule heading. This is not absolutely necessary for billing purposes.



QS Tag is for entering information like the slab thickness and whether the wall is at slab edge to facilitate adjustment for slab and wall junctions.

Length, Width (i.e. thickness), Area and Volume are system parameters.

Note that Area is not always equal to the elevation areas along the centre line as explained later and would need special treatment.

Wall height is not available probably because the height can vary for a wall.

Lengths at wall ends are not available. This is still to be resolved.

Lengths around openings are not available. This is to be resolved through giving more information on the Door and Window Schedules. For blank openings, this is still to be resolved.

QS Mean Area is equal to Volume / Width.

"Mean Area / Length" is equal to QS Mean Area / Length.

REPRESENTATION OF WALL LENGTH, AREA AND VOLUME

Note the wall length, area and volume have the following representations:

	Wall length given	Area given	Volume given
Straight wall e.g. 12 m long x 4 m high x 300 mm thick	Centre line of wall = 12 m	Elevation area on one face = 12 x 4 m = 48 m2	Area along centre line x wall thickness = 12 x 4 x 0.3 m = 14.4 m3
Wall L-shaped on plan with **mitre** joint e.g. 400 mm wall 8 m long + 300 mm wall 5 m long, both x	Centre line of wall with the corner shared between the two wings = 7.85 + 4.80 m	Elevation area based on the **extreme** length of each wing = 8 x 4 + 5 x 4 m = 32 + 20 m2	Area along centre line **with the corner shared** x wall thickness = 7.85 x 4 x 0.4 +

	Wall length given	Area given	Volume given
4 m high	= 12.65 m	= 52 m2	4.80 x 4 x 0.3 = 12.56 + 5.76 m3 = 18.32 m3
Wall L-shaped on plan with **butt** joint e.g. 8 m x 400 mm wall + 4.6 m x 300 mm wall, both x 4 m high	Same as above	Elevation area based on the **self** length of each wing = 8 x 4 + 4.6 x 4 m = 32 + 18.4 m2 = 50.4 m2	Area based on **self** length x wall thickness = 8 x 4 x 0.4 + 4.6 x 4 x 0.3 = 32 x 0.4 + 18.4 x 0.3 = 12.80 + 5.52 m3 = 18.32 m3
Wall L-shaped on plan with **butt** joint e.g. 7.7 m x 400 mm wall + 5 m x 300 mm wall, both x 4 m high	Same as above	Elevation area based on the **self** length of each wing = 7.7 x 4 + 5 x 4 m = 30.8 + 20 m2 = 50.8 m2	Area based on **self** length x wall thickness = 7.7 x 4 x 0.4 + 5 x 4 x 0.3 = 30.8 x 0.4 + 20 x 0.3 = 12.32 + 6 m3 = 18.32 m3
Wall T-off from another wall e.g. 4.6 x 300 mm wall T-off from 400 mm wall	Centre line of T-off wall measured to centre line of main wall = 4.6 + 0.2 = 4.8 m	Elevation area based on the **self** length of T-off wall = 4.6 x 4 m = 18.4 m2	Area based on **self** length x wall thickness = 4.6 x 4 x 0.3 = 5.52 m3

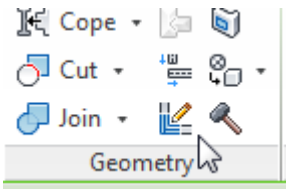
The Area and Volume do not make deduction at the junction with floor slab.

Whether the L-shaped wall is mitre or butt jointed, it does not make any difference to the Volume, but the Areas are different for the three cases.

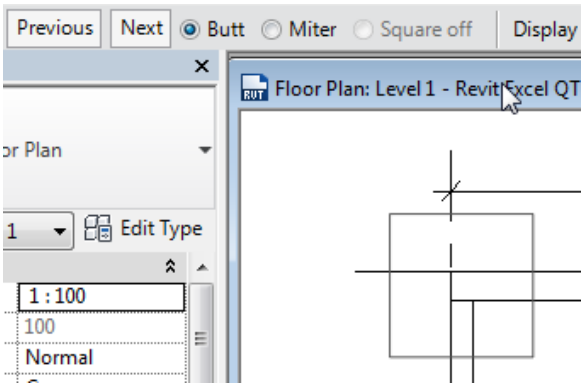
The area of formwork to sides of wall should be equal to the elevation area along the centre line x height x 2 = (7.85 + 4.8) x 4 x 2 = 50.6 m2 x 2. The Areas given by Revit for the three cases of L-shaped wall are useless for this purpose.

Wall junctions on plan:

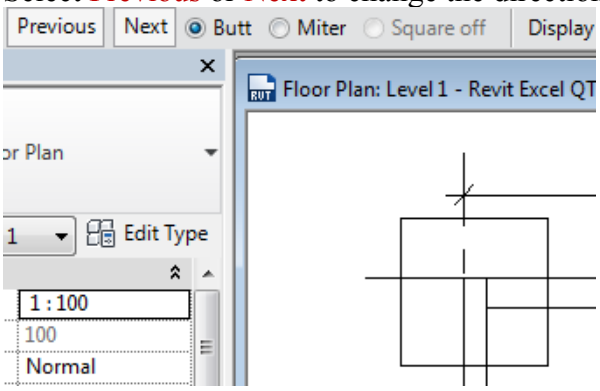
- It is preferred to use mitre joints at corner junction of walls.
- To change the type of wall junctions on plan: select a wall > Wall Joins.



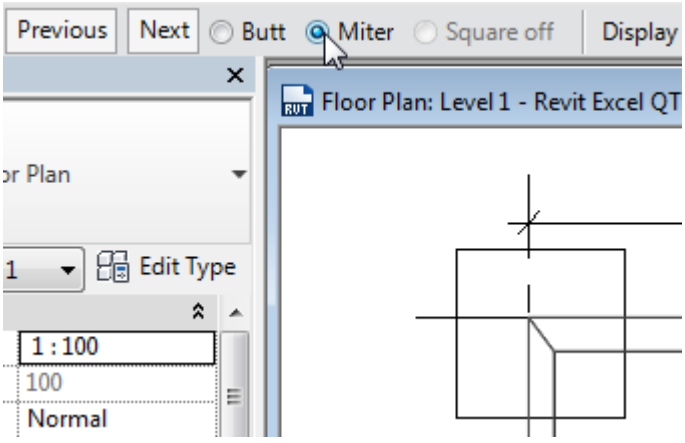
- Select Butt to give a butt joint:



- Select Previous or Next to change the direction of the joint:



- Select Miter for a mitre joint:



After all these discussions, it seems that Volume is a more reliable value to use than Area. Therefore, QS Qty using Wall Area for Estimate and BQ purposes takes the value of Volume / Width, i.e. the QS Mean Area.

QS Qty using Wall Length is to be used for adjustments for the wall and slab junction, it appears that Length is not accurate enough but close enough and is the only choice available. It is tolerated for the time being.

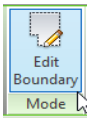
ARCHITECTURAL WALLS

Architectural walls can be modelled just like a structural wall. However, unlike structural walls which can be taken as going up to the floor level because they usually have stronger

concrete grade, architectural walls should go up to beam or ceiling soffit only and not the floor level. Revit does not have a feature to let architectural walls automatically go up and stop there. Therefore, architectural walls have to be modelled one by one to ensure correct height.

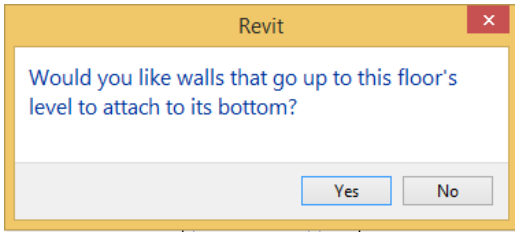
With so many architectural walls within a building, this is a time consuming process and is therefore prone to errors.

FLOOR SLAB AND STRUCTURAL WALL JUNCTIONS

When modelling, after selecting a floor slab > 

> 

> Yes, the following dialogue will appear and will only appear if there are structural walls underneath the slab:



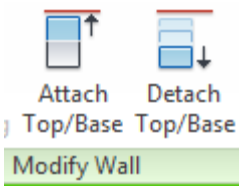
If “Yes” is selected, the volume of the structural walls below will be measured to the underside of the slab, as shown for the wall on the left below.

If “No” is selected, the volume of the structural walls below will be measured to the top of the slab, as shown for the wall on the right below, **but** the reported volume of the slab will not be reduced.

The reported height of the structural walls when defined to be to the top of the slab will not be changed in both cases.

In theory, when a structural wall is attached to the bottom of a floor slab, the wall top will move when the floor slab is moved up or down.

Furthermore, the Modify Wall menu also has the following choices.



However, the behaviour after attaching or detaching walls using the above slab or wall commands is not quite definite every time. Furthermore, since the slab and wall junctions will need to be adjusted in any case, when encountering the above dialogue when editing slab boundary, it is better to answer "No" to retain the default treatment.

STRUCTURAL COLUMN SCHEDULES

A Structural Column Length schedule gives column lengths as QS Qty for the generation of concrete volume and formwork area:

1	Dim - Structural Column Length												
2	Base Level	Column Location Mark	Type : QS Tag : QS Unit	QS Qty	Family	Count	QS App Slab Tk	Top Level	Length	QS Width	QS Depth	Volume	Vol / (Wi x Dp)
3			: :										
4	Level 1	B-2	CL-C40-450x600 : : m	4.00	Column	1	150	Level 2	4000	450	600	1.040	3850
5	Level 1	C-1	CL-C40-600x750 : SL130-EdgeL : m	4.00	Column	1	150	Level 2	4000	600	750	1.733	3850
6	Level 1	C-2	CL-C40-450x600 : SL150 : m	4.00	Column	1	150	Level 2	4000	450	600	1.040	3850
7	Level 1	C-3	CL-C40-450x600 : SL150 : m	4.00	Column	1	150	Level 2	4000	450	600	1.040	3850
8	Level 1	C-4	CL-C40-450x600 : SL150-EdgeS : m	4.00	Column	1	150	Level 2	4000	450	600	1.040	3850

A Structural Column Number schedule gives column numbers as QS Qty for the processing of quantities at junctions with slabs:

1	Dim - Structural Column Nr												
2	Base Level	Column Location Mark	Type : QS Tag : QS Unit	QS Qty	Family	Count	QS App Slab Tk	Top Level	Length	QS Width	QS Depth	Volume	Vol / (Wi x Dp)
3			: :										
4	Level 1	B-2	CL-C40-450x600 : : Nr	1.00	Column	1	150	Level 2	4000	450	600	1.040	3850
5	Level 1	C-1	CL-C40-600x750 : SL130-EdgeL : Nr	1.00	Column	1	150	Level 2	4000	600	750	1.733	3850
6	Level 1	C-2	CL-C40-450x600 : SL150 : Nr	1.00	Column	1	150	Level 2	4000	450	600	1.040	3850
7	Level 1	C-3	CL-C40-450x600 : SL150 : Nr	1.00	Column	1	150	Level 2	4000	450	600	1.040	3850
8	Level 1	C-4	CL-C40-450x600 : SL150-EdgeS : Nr	1.00	Column	1	150	Level 2	4000	450	600	1.040	3850

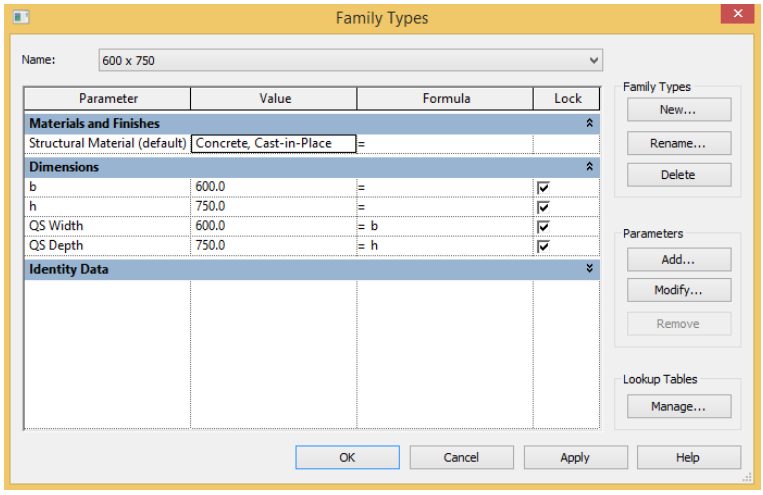
Column Location Mark is a system parameter which gives the grid line references. This is used here instead of the usual Mark.

QS Tag is for entering information like the slab thickness and whether the column is an edge or corner column to facilitate adjustment for slab and column junctions.

For a column defined to be of floor to floor height, Volume is a system parameter which gives the volume of concrete below slab, while Length is a system parameter which gives the floor to floor height.

Since the floor to floor height is needed, therefore "QS Qty" takes the value of Length for Column Length schedule.

It is strange that the column width (b) and depth (h) are not available to the properties window and schedules. Therefore, two shared parameters QS Width and QS Depth have been added to the Family Type parameters to make them available to the schedules to facilitate error checking.



QS App Slab Tk = Length - Volume / (QS Width * QS Depth) which is useful for indicating the approximate slab thickness for counter-checking any errors in positioning the columns.

Vol / (Wi x Dp) = Volume / (QS Width * QS Depth) gives the length of column below slab.

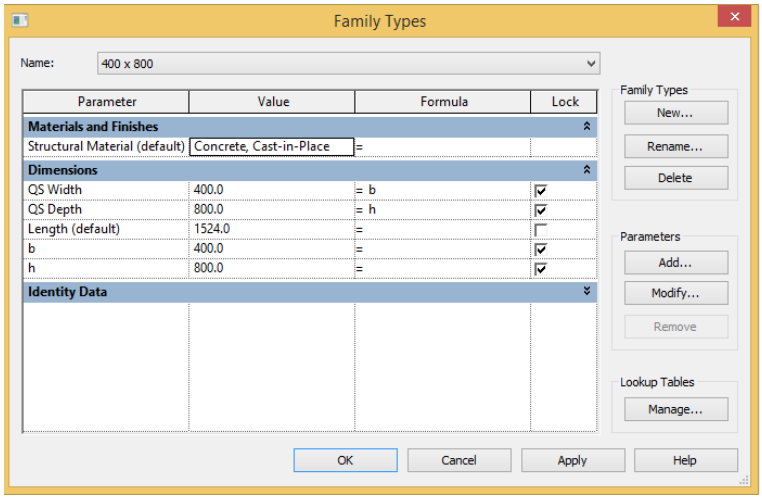
STRUCTURAL FRAMING LENGTH SCHEDULE

Structural Framing Length schedule is basically a beam length schedule:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1	Dim - Structural Framing Length											
2	2	Reference Level	Mark	Type : QS Tag : QS Unit	QS Qty	Family	Count	Approx SL Tk	Length	Cut Length	QS Width	QS Depth	Volume
3	3			: :									
4	4	Level 2	2B1	BM-C30-400x800 : SL130 : m	4.10	Beam	1	130	5000	4100	400	800	1.099
5	5	Level 2	2B2	BM-C30-400x800 : SL130 : m	4.40	Beam	1	130	5000	4400	400	800	1.179
6	6	Level 2	2B3	BM-C30-400x800 : I : m	4.40	Beam	1	0	5000	4400	400	800	1.408
7	7	Level 2	2B4	BM-C30-400x800 : I : m	4.10	Beam	1	0	5000	4100	400	800	1.312
8	8	Level 2	2B5	BM-C30-400x800 : SL150 : m	4.18	Beam	1	150	5000	4175	400	800	1.086

For beams, there are two parameters of Length and Cut Length. Only the Cut Length is the length between supporting columns or walls. QS Qty for Beam Length takes the value of Cut Length.

Similar to structural columns, the beam width (b) and depth (h) are not available to the properties window and schedules. Therefore, two shared parameters QS Width and QS Depth have been added to the Family Type parameters to make them available to the schedules to facilitate error checking.



FLOOR SLAB AREA SCHEDULE

A Floor Slab Area schedule can be:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	1	Dim - Floor Slab Area												
2	2	Level	Mark	Type : QS Tag : QS Unit	QS Qty	Family	Type Mark	Count	Perimeter	Area	Volume			
3	3			: :										
4	4	Level 2	2S1	SL-C30-150 : H3.5-5 : m2	200.00	Floor		1	60000	200.00	30.000			
5	5	Level 2	2S2	SL-C30-130 : H3.5-5 : m2	51.50	Floor		1	30600	51.50	6.695			
6	6	Grand total: 2		: :	251.50			2	90600	251.50	36.695			
7														

QS Tag is for the entry of information about the strutting height.

QS Qty takes Area.

DOOR AND WINDOW SCHEDULES

A Door Number schedule can be:

	A	B	C	D	E	F	G	H	I	J	K
1	1	Dim - Door Number									
2	2	Level	Mark	Type : QS Tag : QS Unit	QS Qty	Family	Door Type	Count	Width	Height	Thickness
3	3			: :							
4	4	Level 1	1	DR-D1-Double-1830x2134 : EW-CON-125 : Nr	1.00	M_Double-Glass 1	D1	1	1830	2134	51
5	5	Level 1	2	DR-D2-Single-915x2134 : IW-BLK-100 : Nr	1.00	M_Single-Flush	D2	1	915	2134	51
6	6	Level 1	3	DR-D2-Single-915x2134 : WA-CON-300 : Nr	1.00	M_Single-Flush	D2	1	915	2134	51
7	7	Grand total: 3		: :	3.00			3			

A Window Number schedule can be:

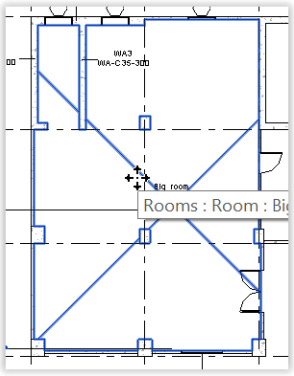
	A	B	C	D	E	F	G	H	I	J	K
1	1	Dim - Window Number									
2	2	Level	Mark	Type : QS Tag : QS Unit	QS Qty	Family	Type Mark	Count	Width	Height	Sill Height
3	3			: :							
4	4	Level 1	1	WD-W1-1220x1220 : WA-CONC-400 : Nr	1.00	M_Casement Dbl with Trim		1	1220	1220	11
5	5	Level 1	2	WD-W1-1220x1220 : WA-CONC-400 : Nr	1.00	M_Casement Dbl with Trim		1	1220	1220	1100
6	6	Level 1	3	WD-W1-1220x1220 : EW-CONC-125 : Nr	1.00	M_Casement Dbl with Trim		1	1220	1220	1100
7	7	Level 1	4	WD-FL2-0610x1830 : WA-CONC-300 : Nr	1.00	M_Fixed		1	610	1830	1100
8	8	Level 1	5	WD-FL3-0406x1830 : EW-CONC-125 : Nr	1.00	M_Fixed		1	406	1830	1100

QS Tag is for the entry of information about the walls housing the doors and windows to facilitate future measurement of formwork to jambs and soffit, boxing and lintels.

ROOM SCHEDULES

The default room schedules provided by Revit can give Floor Area and Perimeter, but not the Nett Ceiling and Beam surface areas, nor wall and column surface areas:

1	Room Schedule											
2	Level	Name	Floor Finish	Ceiling Finish	Wall Finish	Count	Area	Perimeter	Base Offset	Upper Limit	Limit Offset	Unbounded Height
3												
4	Level 1	Big room	FL-Ceramic Tile-Screed-50	CE-Plaster-Emulsion	WL-Plaster-Emulsion	1	139.42	61576	0	Level 2	-150	3850
5	Level 1	Middle room	FL-Ceramic Tile-Screed-50	CE-Plaster-Emulsion	WL-Ceramic Tile-Screed	1	69.44	39650	0	Level 2	-150	3850
6	Level 1	Small room	FL-Ceramic Tile-Screed-50	CE-Plaster-Emulsion	WL-Plaster-Emulsion	1	19.36	17600	0	Level 2	-150	3850
7	Grand total: 3					3	228.22	118826				



By defining the floor level above a room as the Upper Limit, and entering the slab thickness in negative value as the Limit Offset, the reported Unbounded Height will give the floor to ceiling soffit height, which should be good for generating the wall and column surface areas. Adjustment would need to be made for the beam surfaces and end junctions. Adjustment for window and door openings may be taken care of when processing the window and door quantities.

To provide QS Qty for floor, wall, ceiling and skirting, 4 separate schedules are adapted from the Room Schedule.

The Room Floor Finishes Area schedule:

1	Dim - Room Floor Finishes Area							
2	Level	Name	Type : QS Tag : QS Unit	QS Qty	Floor Finish	Count	Area	
3			: :					
4	Level 1	Big room	FL-Ceramic Tile-Screed-50 : : m2	139.42	FL-Ceramic Tile-Screed-50	1	139.42	
5	Level 1	Middle room	FL-Ceramic Tile-Screed-50 : : m2	69.44	FL-Ceramic Tile-Screed-50	1	69.44	
6	Level 1	Small room	FL-Ceramic Tile-Screed-50 : : m2	19.36	FL-Ceramic Tile-Screed-50	1	19.36	
7	Grand total: 3		: :	228.22		3	228.22	

The Room Ceiling Finishes Area schedule:

1	Dim - Room Ceilg Finishes Area								
2	Level	Name	Type : QS Tag : QS Unit	QS Qty	Ceiling Finish	Count	Area		
3			: :						
4	Level 1	Big room	CE-Plaster-Emulsion : : m2	139.42	CE-Plaster-Emulsion	1	139.42		
5	Level 1	Middle room	CE-Plaster-Emulsion : : m2	69.44	CE-Plaster-Emulsion	1	69.44		
6	Level 1	Small room	CE-Plaster-Emulsion : : m2	19.36	CE-Plaster-Emulsion	1	19.36		
7	Grand total: 3		: :	228.22			228.22		

The Room Wall Finishes Area schedule, with skirting area to be deducted from wall area:

	A	B	C	D	E	F	G	H	I	J	K	L
1	1	Dim - Room Wall Finishes Area										
2	2	Level	Name	Type : QS Tag : QS Unit	QS Qty	Wall Finish	Count	Perimeter	Base Offset	Upper Limit	Limit Offset	Unbounded Height
3	3			: :								
4	4	Level 1	Big room	WL-Plaster-Emulsion : : m2	237.07	WL-Plaster-Emulsion	1	61576	0	Level 2	-150	3850
5	5	Level 1	Middle room	WL-Ceramic Tile-Screed : : m2	152.65	WL-Ceramic Tile-Screed	1	39650	0	Level 2	-150	3850
6	6	Level 1	Small room	WL-Plaster-Emulsion : : m2	67.76	WL-Plaster-Emulsion	1	17600	0	Level 2	-150	3850
7	7	Grand total: 3		: :	457.48		3	118826				

The Room Perimeter Length schedule for skirting:

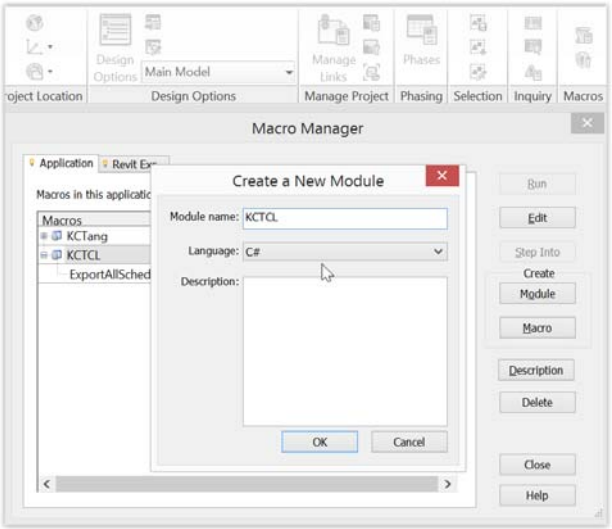
	A	B	C	D	E	F	G	H	I
1	1	Dim - Room Perimeter Length							
2	2	Level	Name	Type : QS Tag : QS Unit	QS Qty	Floor Finish	Count	Perimeter	
3	3			: :					
4	4	Level 1	Big room	FL-Ceramic Tile-Screed-50 : : m	61.58	FL-Ceramic Tile-Screed-50	1	61576	
5	5	Level 1	Middle room	FL-Ceramic Tile-Screed-50 : : m	39.65	FL-Ceramic Tile-Screed-50	1	39650	
6	6	Level 1	Small room	FL-Ceramic Tile-Screed-50 : : m	17.60	FL-Ceramic Tile-Screed-50	1	17600	
7	7	Grand total: 3		: :	118.83		3	118826	

Room elements do not have a Type parameter. A calculated field has been created for Type, which takes the names of the finishes as its values.

EXPORTING SCHEDULES TO EXCEL

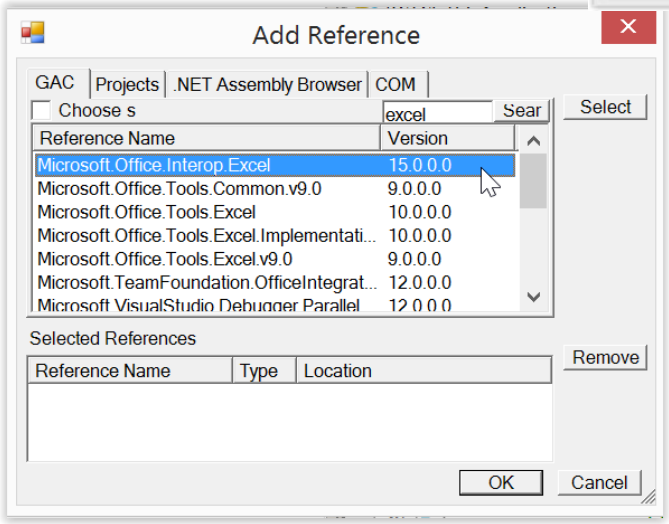
The following is a macro to export all schedules to one Excel file as described earlier.

To install, select: Manage ribbon > Macros - Macro Manager > Create - Module > enter KCTCL at Module name > OK > Edit to open the SharpDevelop coding screen.



Copy the codes below to the SharpDevelop coding screen and overwrite the existing codes generated by Revit.

Select SharpDevelop's Project > Add Reference. Search for Microsoft Office Interop Excel. Click Select > OK to confirm. Select Build > Build Solution.



Go back to Revit's macro manager menu. Select the newly created macro, and select Run.

```
/*
 * Created by SharpDevelop.
 * User: K C Tang
 * Date released for use : 31/12/2014
 * Date last revised: 19/1/2015
 *
 * To change this template use Tools | Options | Coding | Edit Standard Headers.
 */
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using Autodesk.Revit.DB;
using Autodesk.Revit.DB.Architecture;
using Autodesk.Revit.UI;
using Autodesk.Revit.UI.Selection;
using Excel = Microsoft.Office.Interop.Excel;

namespace KCTCL
{
   [Autodesk.Revit.Attributes.Transaction(Autodesk.Revit.Attributes.TransactionMode.Manual)]
   [Autodesk.Revit.DB.Macros.AddInId("E77FD3DE-05E8-4FD3-B85A-116F5B6F2EEF")]
   public partial class ThisApplication
   {
      private void Module_Startup(object sender, EventArgs e)
      {

      }

      private void Module_Shutdown(object sender, EventArgs e)
      {

      }

      #region Revit Macros generated code
      private void InternalStartup()
      {
         this.Startup += new System.EventHandler(Module_Startup);
         this.Shutdown += new System.EventHandler(Module_Shutdown);
      }
      #endregion

      public void ExportAllSchedulesToOneExcel()
      {
         // initilize variables
         char[] char_tab = new char[] { '\t' };
         char[] char_quote = new char[] { '"' };
         string text_line;
         string qs_desc;
         decimal qs_qty;
         List<string> text_line_list = new List<string>();
         SortedList<string, decimal> qs_desc_list = new SortedList<string, decimal>();
            object default_value = System.Reflection.Missing.Value;
           // select active document
         Document doc = this.ActiveUIDocument.Document;
            // get filename from doc.Title
         string filename_no_ext = doc.Title;
            // add ".rvt" temporarily doc.Title not ending with ".rvt"
            // because file explorer hides the extension
            if (!filename_no_ext.EndsWith(".rvt"))
            {
             filename_no_ext = filename_no_ext + ".rvt";
            }
         // get active folder name by removing the full file name
         // from the full pathname which contains the full file name
            string folder_name = doc.PathName.Replace(filename_no_ext, "");
         // change file extension to the current datetime string
         // to avoid overwriting existing files
         filename_no_ext = filename_no_ext.Replace(".rvt",
            DateTime.Now.ToString("-yyyyMMdd-HHmmss"));
```

```
    // initilize Excel variables
    Excel.Application xlApp;
   Excel.Workbook xlWorkBook;
  Excel.Worksheet xlWorkSheet;
  Excel.Worksheet xlWorkSheetAllDim;
  Excel.Range xlSelectedRange;
    xlApp = new Excel.Application();
    // check whether Excel is installed
    if (xlApp == null)
    {
     TaskDialog.Show("ExportAllSchedulesToOneExcel", "Excel is not installed!!");
     return;
    }
    // create new workbook which by default contains at least 1 worksheet
    xlWorkBook = xlApp.Workbooks.Add(default_value);
    // initialize 2 worksheet variables, all referring to Sheet1 for the time being
    xlWorkSheetAllDim = (Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
   xlWorkSheet = (Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
  // rename Sheet1 to contain contents of all future worksheets
  // with names starting with "Dim - "
    xlWorkSheetAllDim.Name = "All Dim";
    // make Excel file visible
  xlApp.Visible = true;
    // maximize workbook window
    xlApp.ActiveWindow.WindowState = Excel.XlWindowState.xlMaximized;
    // read viewschedules in active document
  ViewScheduleExportOptions opt = new ViewScheduleExportOptions();
  FilteredElementCollector col = new FilteredElementCollector(doc).OfClass(typeof(ViewSchedule));
    if (col.ToElementIds().Count == 0)
    {
     TaskDialog.Show("ExportAllSchedulesToOneExcel", "No schedule available!!");
     // close workbook without saving
     xlWorkBook.Close(false, default_value, default_value);
     xlApp.Quit();
     // release objects
     releaseObject(xlWorkSheet);
     releaseObject(xlWorkSheetAllDim);
     releaseObject(xlWorkBook);
     releaseObject(xlApp);
     return;
    }
  // sort elements in col in descending order
  IOrderedEnumerable<ViewSchedule> sorted_col =
      from ViewSchedule vs in col orderby vs.Name ascending select vs;
  // process schedule in ascending order
  int all_dim_row_num = 0;
  foreach (ViewSchedule vs in sorted_col)
    {
     // check if schedule name too long
     if (vs.Name.Length > 31 )
     {
        TaskDialog.Show("ExportAllSchedulesToOneExcel",
           vs.Name + "\n" + "Schedule name should not be more than 31 characters!!");
        // release objects
        releaseObject(xlWorkSheet);
        releaseObject(xlWorkSheetAllDim);
          releaseObject(xlWorkBook);
          releaseObject(xlApp);
        return;
     }
  }
  foreach (ViewSchedule vs in sorted_col)
    {
     if (vs.Name.StartsWith("<"))
     {
        // skip schedule with name beginning with "<", such as "<Revision Schedule>"
     } else
     {
        // reduce filename length longer than 31
        if (31 < vs.Name.Length )
        {
           vs.Name = vs.Name.Substring(0, 14) + " name length > 31";
        }
        // replace special character with "_"
        vs.Name = vs.Name
```

```
                .Replace( ':', '_' )
                .Replace( '*', '_' )
                .Replace( '?', '_' )
                .Replace( '/', '_' )
                .Replace( '\\', '_' )
                .Replace( '[', '_' )
                .Replace( ']', '_' );
            // export schedule to txt file
            vs.Export(folder_name, filename_no_ext + ".txt", opt);
            // add a worksheet
            xlWorkSheet = (Excel.Worksheet)xlWorkBook.Worksheets.Add(default_value);
            // move it to become the last worksheet
            xlWorkSheet.Move(default_value, xlWorkBook.Worksheets[xlWorkBook.Worksheets.Count]);
            // name worksheet as schedule name
            xlWorkSheet.Name = vs.Name;
            // initialize variables
            bool bold_flag = true;
            int row_num = 0;
            int array_count = 0;
            int[] QS_Desc_Col = new int[4];
            string value = "";
            // read txt file
            System.IO.StreamReader file = new System.IO.StreamReader(folder_name
                + "\\" + filename_no_ext + ".txt");
            // read process each line
            while((text_line = file.ReadLine()) != null)
            {
                // increase row_num
                row_num += 1;
                // convert line into list by:
                // splitting line when tab is encountered
                // trimming away quotation marks
                text_line_list = text_line.Split(char_tab).Select<string, string>(s => s.Trim(char_quote)).ToList();
                // if worksheet name starts with "Dim - "
                if (vs.Name.StartsWith("Dim - "))
                {
                    // check line 2 for 3 columns to be combined to form QS Desc
                    if (row_num == 2)
                    {
                        for (array_count = 0; array_count < text_line_list.Count; array_count++)
                        {
                            // store the column numbers of the desired columns
                            value = text_line_list[array_count];
                            switch (value)
                            {
                                case "Type" :
                                    QS_Desc_Col[0] = array_count;
                                    break;
                                case "QS Tag" :
                                    QS_Desc_Col[1] = array_count;
                                    break;
                                case "QS Unit" :
                                    QS_Desc_Col[2] = array_count;
                                    break;
                                case "QS Qty" :
                                    QS_Desc_Col[3] = array_count;
                                    break;
                                default:
                                    break;
                            }
                        }
                    }
                    // if not first row, and QS Unit and Type not empty
                    if ((row_num != 1) && (QS_Desc_Col[2] != 0) && (QS_Desc_Col[1] != 0))
                    {
                        // combine the 3 columns into "Type : QS Tag : QS Unit" format
                        qs_desc =
                            text_line_list[QS_Desc_Col[0]] + " : " +
                            text_line_list[QS_Desc_Col[1]] + " : " +
                            text_line_list[QS_Desc_Col[2]];
                        text_line_list[QS_Desc_Col[0]] = qs_desc;
                        // assign qs_qty
                        qs_qty = 0;
                        if (QS_Desc_Col[3] != 0)
                        {
```

```
                    if (decimal.TryParse(text_line_list[QS_Desc_Col[3]], out qs_qty)){}
                }
                // if not equal to "Type : QS Tag : QS Unit" and not " :  : "
                if ((qs_desc != "Type : QS Tag : QS Unit") && (qs_desc != " :  : ") )
                {
                    // add qs_desc and qs_qty to list if not already there
                    // add qs_qty if already there
                    if (!qs_desc_list.ContainsKey(qs_desc))
                    {
                        qs_desc_list.Add(qs_desc, qs_qty);
                    }
                    else
                    {
                        qs_desc_list[qs_desc] += qs_qty;
                    }
                }
                // remove QS Tag and QS Unit
                text_line_list.RemoveAt(QS_Desc_Col[2]);
                text_line_list.RemoveAt(QS_Desc_Col[1]);
            }
        }
        // input into individual worksheet
        // assign value of first column with row num
        xlWorkSheet.Cells[row_num, 1] = row_num;
        // process other columns
        for (array_count = 0; array_count < text_line_list.Count; array_count++)
        {
            value = text_line_list[array_count];
            // array_count + 2 so that the first column is column B instead of A
            xlWorkSheet.Cells[row_num, array_count + 2] = value;
            xlSelectedRange = xlWorkSheet.get_Range(cellAddress(row_num, array_count + 2));
            if (value.Contains("."))
            {
                xlSelectedRange.NumberFormat = numFormat(value);
            }
            // bold first 2 lines, otherwise stop bold
            if (row_num <= 2)
            {
                xlSelectedRange = xlWorkSheet.get_Range("A" + row_num, "B" + row_num);
                xlSelectedRange.EntireRow.Font.Bold = true;
            } else
            {
                bold_flag = false;
            }
        }
        // input into All Dim worksheet for schedules with names starting with "Dim - "
        if (vs.Name.StartsWith("Dim - "))
        {
            all_dim_row_num += 1;
            // assign value of first column with row num
            xlWorkSheetAllDim.Cells[all_dim_row_num, 1] = all_dim_row_num;
            // process other columns
            for (array_count = 0; array_count <= text_line_list.Count - 1; array_count++)
            {
                value = text_line_list[array_count];
                // array_count + 2 so that the first column is column B instead of A
                xlWorkSheetAllDim.Cells[all_dim_row_num, array_count + 2] = value;
                xlSelectedRange = xlWorkSheetAllDim.get_Range(cellAddress(all_dim_row_num, array_count + 2));
                if (value.Contains("."))
                {
                    xlSelectedRange.NumberFormat = numFormat(value);
                }
            }
            // bold first 2 lines from each individual worksheet
            if (bold_flag)
            {
                xlSelectedRange = xlWorkSheetAllDim.get_Range("A" + all_dim_row_num, "B" + all_dim_row_num);
                xlSelectedRange.EntireRow.Font.Bold = true;
            }
        }
    }
    // autofit column widths
    xlWorkSheet.Columns.EntireColumn.AutoFit();
    // close txt file
    file.Close();
```

```
            // delete txt file
            System.IO.File.Delete(folder_name + "\\" + filename_no_ext + ".txt");
        }
    }
    // autofit column widths of All Dim worksheet
    xlWorkSheetAllDim.Columns.EntireColumn.AutoFit();
    // move it to become the first worksheet
    xlWorkSheetAllDim.Move(xlWorkBook.Worksheets[1]);
    // add and name a worksheet to contain unique QS Desc
    xlWorkSheet = (Excel.Worksheet)xlWorkBook.Worksheets.Add(default_value);
    xlWorkSheet.Name = "QS Desc";
    int row_num2 = 1;
    xlWorkSheet.Cells[row_num2, 1] = "Type : QS Tag : QS Unit";
    xlWorkSheet.Cells[row_num2, 2] = "QS Qty";
    xlSelectedRange = xlWorkSheet.get_Range("A1","B1");
    xlSelectedRange.Font.Bold = true;
    xlSelectedRange = xlWorkSheet.get_Range("B1");
    xlSelectedRange.EntireColumn.NumberFormat = "0.00";
    // input each qs_desc into column A;
    foreach (KeyValuePair<string, decimal> kvp in qs_desc_list)
    {
        row_num2 += 1;
        xlWorkSheet.Cells[row_num2, 1] = kvp.Key;
        xlWorkSheet.Cells[row_num2, 2] = kvp.Value;
    }
    // autofit column widths of QS Desc worksheet
    xlWorkSheet.Columns.EntireColumn.AutoFit();
    // move it to become the first worksheet
    xlWorkSheet.Move(xlWorkBook.Worksheets[1]);
    // save workbook
    xlWorkBook.SaveAs(folder_name + "\\" + filename_no_ext,
        Excel.XlFileFormat.xlWorkbookNormal,
        default_value, default_value, default_value, default_value,
        Excel.XlSaveAsAccessMode.xlExclusive,
        default_value, default_value, default_value, default_value, default_value);
    // release objects
    releaseObject(xlWorkSheet);
    releaseObject(xlWorkSheetAllDim);
    releaseObject(xlWorkBook);
    releaseObject(xlApp);
}

private    string numFormat(string value)
{
    switch (value.Length - value.LastIndexOf("."))
    {
        case 2:
            // 1 decimal place
            return "0.0";
        case 3:
            // 2 decimal places
            return "0.00";
        case 4:
            // 3 decimal places
            return "0.000";
        default:
            return "";
    }
}

private    string cellAddress(int row, int col)
{
    // change cell address from (100, 1) to (A100) style
    string prompt = (row + "\n\t" + col + "\n\t");
    if (row < 1 || row > 1048576)
    {
        TaskDialog.Show("Excel Row Number", "Error - must be within 1 - 1048576!!");
        return null;
    }
    if (col < 1 || col > 16384)
    {
        TaskDialog.Show("Excel Column Number", "Error - must be within 1 - 16384!!");
        return null;
    }
    // convert column number to alphabetical reference
```

```
        int remainder = 0;
        string result = "";
        for (int i = 0; i < 3; i++)
        {
            // get remainder after division by 26
            remainder = ((col - 1) % 26) + 1;
            if (remainder != 0)
            {
                // match the remainder to alphabets A to Z where A is char 65
                // precede the alphabet to the previous result
                result = Convert.ToChar(remainder + 64).ToString() + result;
            }
            col = ((col - 1) / 26);
            // do it three times
        }
        // append row number to alphabetical column reference
        result = result + row.ToString();
        return result;
    }

    private void releaseObject(object obj)
    {
        try
        {
            System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);
            obj = null;
        }
        catch (Exception ex)
        {
            obj = null;
            TaskDialog.Show("Excel file created","Exception Occurred while releasing object " + ex.ToString());
        }
        finally
        {
            GC.Collect();
        }
    }

  }
}
```

CLOSING

Having set up the Revit schedules and the corresponding Excel worksheets once, they can be re-used as a set of templates for other Revit models. The number of chains of QS Desc -> secondary calculations -> Codes can be expanded as and when they are encountered and retained in the templates to serve future use to reduce the burden of re-defining every time.

Get into modelling which is easy and powerful as soon as possible. Understand it, identify the limitations and suggest solutions to make it really productive. Increase our user base and join force to push the software vendors to meet our need.

End