

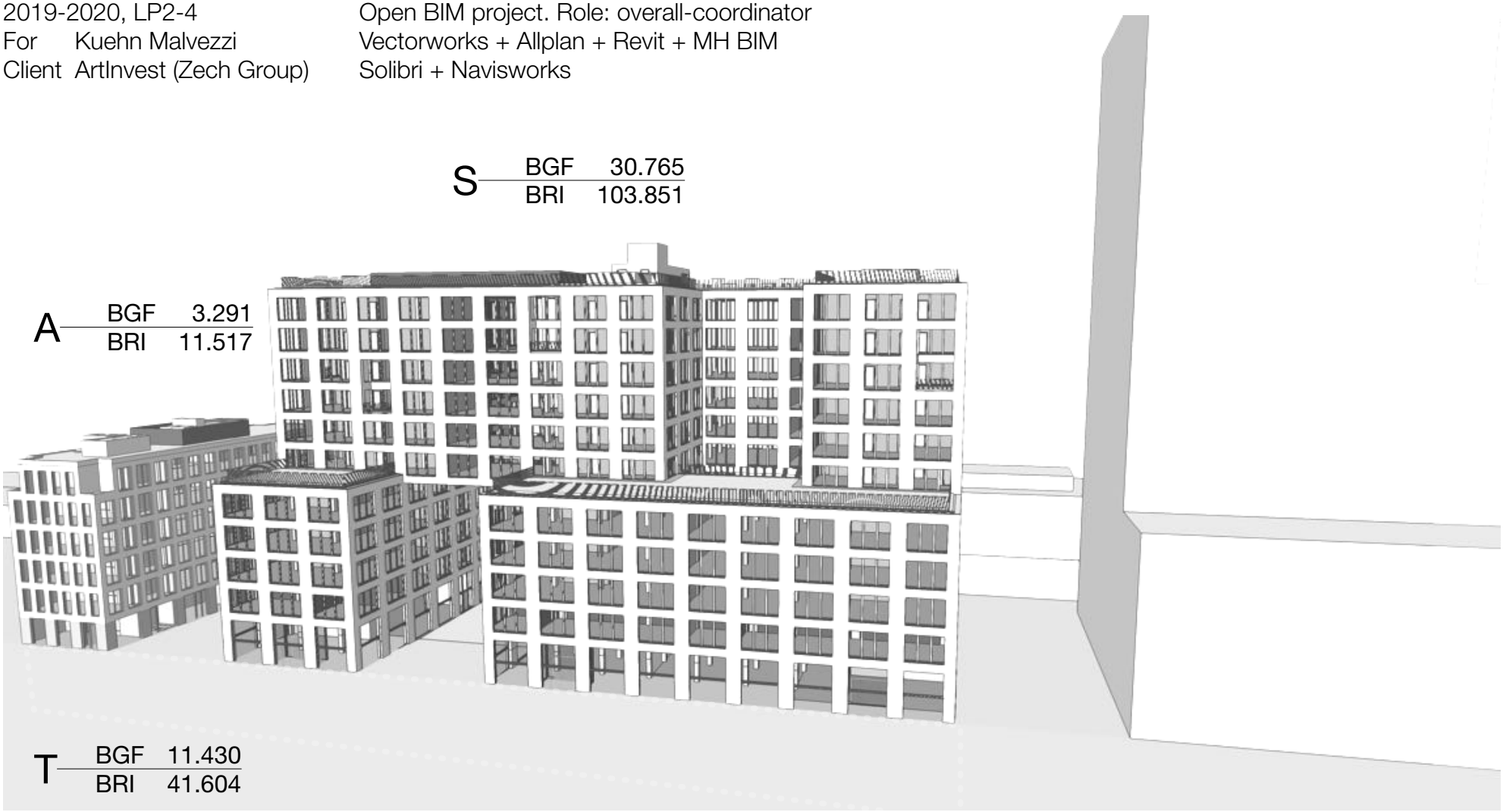
BIM Overall Coordination

JAHO

Jannowitzbrücke: 2 buildings, 2 underground levels

2019-2020, LP2-4
For Kuehn Malvezzi
Client ArtInvest (Zech Group)

Open BIM project. Role: overall-coordinator
Vectorworks + Allplan + Revit + MH BIM
Solibri + Navisworks



Team

1 project director
1 senior architect (assistent)
1 senior architect (details)
1 architect (plans)
2 trainees
1 bim inform. coordinator

	Walls	Slabs	Stairs	Columns	Windows	Doors
S	3076	623	52	1541	765	851
A	292	118	12	216	156	111
T	757	68	16	857	0	181
	4125	**809**	**80**	**2614**	**921**	**1143**

BIM role

VDI 2552-1

Information coordinator - construction project
Informationskoordinator Bauprojekt / BIM-Gesamt-Koordinator LPH 1-4

External tasks

Technical feasability EIR AIA > BEP BAP
Agreements over IFC parameter list with planning partners

Establish a 10 steps BIM-Worklow
Team schooling, 6 persons
Model-checking (Solibri): project, static, norm, collisions
Represent the office in the BIM Allianz

Internal tasks

write new Windows
write new Doors
put "intelligence" in other parametric tools

create supporting custom records
create custom psets
create a custom IFC mapping (xml)
create controlling tables for all ifc elements
create an accompanying script (efficiency, reduce mistakes)
clean up project every night before office time

8 months

Auftraggeber-Informations-Anforderungen (AIA)

-JAHO-
Version 1.00 | 09.08.2019

BIM-Abwicklungs-Plan (BAP)

-JAHO-
Version 1.01 | 04.05.2020

Accompanying IFC script: > 5000 lines

```
(**)
ForEachObject(SetBeam, (NOTINREFDLVP & NOTINDLVP & ((PON='StructuralMember') & ('Structural

cnt := Count((PON='FramingMember'));
IF cnt > 0 THEN BEGIN
    ALLOCATE labelList[1..cnt, 1..2];
    arrCnt := 0;
    strt := 1;
    ForEachObject(SetBeam, (NOTINREFDLVP & NOTINDLVP & (PON='FramingMember')));
END;
{}

ForEachObject(SetColumn, (NOTINREFDLVP & NOTINDLVP & ((PON='StructuralMember') &
cturalMember'.'StructuralUse'='Column'))));
ForEachObject(SetColumn, (NOTINREFDLVP & NOTINDLVP & (PON='Pilaster2')));
{ ForEachObject(SetColumn, (NOTINREFDLVP & NOTINDLVP & (ST=PILLAR))); THE PILASTER SHOULD B
}

ForEachObject(SetColumn, (NOTINREFDLVP & NOTINDLVP & (PON='Column2')));

ForEachObject(SetStair, (NOTINREFDLVP & NOTINDLVP & (PON='Simple Stair')));
ForEachObject(SetStair, (NOTINREFDLVP & NOTINDLVP & (PON='Custom Stair')));

ForEachObject(SetRailing, (NOTINREFDLVP & NOTINDLVP & (PON='Railing CW')));
ForEachObject(SetRailing, (NOTINREFDLVP & NOTINDLVP & (PON='Handrail (Straight)')));
ForEachObject(SetRailing, (NOTINREFDLVP & NOTINDLVP & (PON='Handrail (Curved)')));

ForEachObject(SetCeiling, (NOTINREFDLVP & NOTINDLVP & (PON='Ceiling Grid')));
ForEachObject(SetRoof, (NOTINREFDLVP & NOTINDLVP & (T=71))); { roof face }
ForEachObject(SetRoof, (NOTINREFDLVP & NOTINDLVP & (T=83))); { roof }

ForEachObject(SetSpace, (NOTINREFDLVP & NOTINDLVP & (PON='Space')));
RenumberSpaces;

ForEachObject(SetRamp, (NOTINREFDLVP & NOTINDLVP & (PON='Ramp')));

ForEachObject(SetDoor, (NOTINREFDLVP & NOTINDLVP & (PON='Door')));
ForEachObject(SetWindow, (NOTINREFDLVP & NOTINDLVP & (PON='Window')));
ForEachObject(SetCBMSkylight, (NOTINREFDLVP & NOTINDLVP & (PON='CBM-Flat Roof Window')));

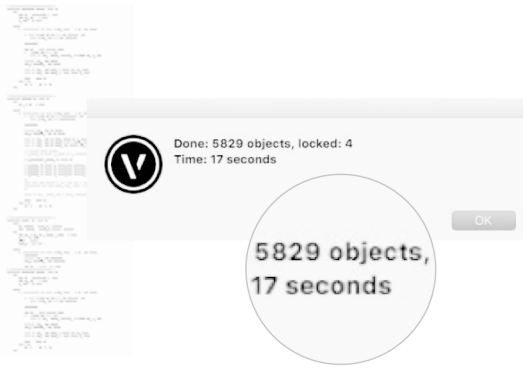
ForEachObject(SetStair, (NOTINREFDLVP & NOTINDLVP & (PON='Stairs CW')));

ForEachObject(SetSlab, (NOTINREFDLVP & NOTINDLVP & (PON='Slab')));
ForEachObject(SetSanitary, (NOTINREFDLVP & NOTINDLVP & (S='JAHO-S-*')));
ForEachObject(SetWall, (NOTINREFDLVP & NOTINDLVP & ((T=WALL) OR (T=ROUNDWALL))));


ForEachObject(SetCBMWindow, (NOTINREFDLVP & NOTINDLVP & (PON='CBM-Window')));
ForEachObject(SetCBMDoor, (NOTINREFDLVP & NOTINDLVP & (PON='CBM-Door')));

tick2 := GetTickCount;

AlrtDialog(Concat(
'Done: ', gCnt, ' objects, locked: ', gError, Chr(13),
'Time: ', (tick2 - tick1) DIV 60, ' seconds'
));
```



Delivering requirements: IFC parameter list

What is not in IFC, must be brought in as custom pset.

Determine workflow: can the task be coded or not:

- Y implement code
- N implement tools for the CAD drafters (minimize mistakes)

Bauelement	Eigenschaften set	Parameter IFC	Verfügbarkeit	Datentyp	Farbe	Nur visualisieren	Zeichenfolgen wert(e)	Einheit	Beispielwert	Parameter	Parameter CAD	Fachmodell	Lieferant	Bearbeiter	Pflege	Data Drops								
																0	1.1a	2.1a	2.2a	2.3a	2.4a	2.5a	3.1a	3.2a
Fenster	IfcSite	IfcSite	Must exist	Text				-	Standort	Grundstück	IfcSite	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	IfcBuilding	IfcBuilding	Must exist	Text				-	Stadthaus	Gebäude	Datei	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	IfcBuildingStorey	IfcBuildingStorey	Must exist	Text				-	U1, 00	Geschoss	Story	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	-	GlobalID	Must exist	Text				-	0tfBdiyhbAu9MoSQ6k8t$O	ElementID	-	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	-	-	Must exist	Numeric				m	2,14	Höhe	-	Objektplanerm	Objektplaner	Objektplaner	IFC Viewer									
Fenster	-	-	Must exist	Numeric				m	0,89	Breite	-	Objektplanerm	Objektplaner	Objektplaner	IFC Viewer									
Fenster	-	-	Must exist	Numeric				m	0,24	Wanddicke	-	Objektplanerm	Objektplaner	Objektplaner	IFC Viewer									
Fenster	-	-	Must exist	Numeric				m²	2,00	Fläche (Rohbauöffnung)	-	Objektplanerm	Objektplaner	Objektplaner	IFC Viewer									
Fenster	IfcWindow	IfcWindow	Must exist	Text		TRUE	*	-	Fenster	Bauteil	Window/CBM-Window	Objektplanerm	Objektplaner	Objektplaner	Vectorworks		x	x	x	x				
Fenster	IfcWindow	Description	Must exist	Text		TRUE	*	-	Fenster-2Flg	Bauteilname	Description	Objektplanerm	Objektplaner	Objektplaner	Vectorworks			x	x	x				
Fenster	Classification	Reference	Must exist	Text		TRUE	*	-	334	Zuordnung nach DIN 276	ItemReference	Objektplanerm	Objektplaner	Objektplaner	Vectorworks			x	x	x				
Fenster	Ifc Dimensions	Overall Width	Must exist	Numeric		TRUE	*	-	1,01	Rohbauhöhe (NM DIN 18100)	OverallWidth	Objektplanerm	Objektplaner	Objektplaner	Vectorworks		x	x	x	x				
Fenster	Ifc Dimensions	Overall Height	Must exist	Numeric		TRUE	*	-	1,01	Rohbaubreite (NM DIN 18100)	OverallHeight	Objektplanerm	Objektplaner	Objektplaner	Vectorworks		x	x	x	x				
Fenster	ePset_KM_Win	Nr	Must exist	Text		TRUE	*	-	xxx	Fensternummer	Nr	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	ePset_KM_Win	Fensterteile	Must exist	Numeric		TRUE	*	-	4	Fensterteile	Fensterteile	Objektplanerm	Objektplaner	Objektplaner	Vectorworks			x	x	x				
Fenster	ePset_KM_Win	Fluegelanzahl	Must exist							Flügelanzahl	Fluegelanzahl	Objektplanerm	Objektplaner	Objektplaner	Vectorworks		x	x	x	x				
Fenster	ePset_KM_Win	Schallschutzklasse	Must exist	Text		TRUE	*	dB	Rw, R = 37	Schallschutzklasse	Schallschutzklasse	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	ePset_KM_Win	Feuerwiderstandsklasse	Must exist	Text		TRUE	*	-	nb	Feuerwiderstandsklasse	Feuerwiderstandsklasse	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
~~Fenster~~	~~ePset_KM_Win~~	~~Rauchschutzklasse~~	~~Must exist~~	~~Text~~		TRUE	*	-	RS	~~Rauchschutzklasse~~	~~Rauchschutzklasse~~	~~Objektplanerm~~	~~Objektplaner~~	~~Objektplaner~~	~~Vectorworks~~									
Fenster	ePset_KM_Win	Sonnenschutz	Must exist	Text		TRUE	*	-	Verglasung	Sonnenschutz	Sonnenschutz	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	ePset_KM_Win	Rahmenmaterial	Must exist	Text		TRUE	*	-	Aluminium	Rahmenmaterial	Rahmenmaterial	Objektplanerm	Objektplaner	Objektplaner	Vectorworks			x	x	x				
Fenster	ePset_KM_Win	Antrieb_Oeffnung	Must exist	Text		TRUE	*	-	motorisch	Antrieb_Öffnung	Antrieb_Oeffnung	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	ePset_KM_Win	Blendschutz	Must exist	Boolean		TRUE	*	-	Ja	Blendschutz	Blendschutz	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	ePset_KM_Win	Fensterbankausladung	Must exist	Numeric		TRUE	*	m	0,05	Ausladung	Fensterbankausladung	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	ePset_KM_Win	Lichtdurchlaessigkeit	Must exist	Text		TRUE	*	-	52/29	Lichtdurchlässigkeit	Lichtdurchlaessigkeit	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	ePset_KM_Win	Glasflaeche	Must exist	Numeric		TRUE	*	m	1,50	Glasfläche	Glasflaeche	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									
Fenster	ePset_KM_Win	Beschlag	Must exist	Text		TRUE	*	-	Abschließbar	Beschläge	Beschlag	Objektplanerm	Objektplaner	Objektplaner	Vectorworks									

Inhaltsverzeichnis | Hinweise | Wand | Decke | Balken | Stütze | Treppe | Rampe | Fundament | Dach | Raum | Vorhangfassade | Abhangdecke | Fußböden | Tür | **Fenster** | Geländer | Ausstattung | Durchbrüche | Sanitärtechnik | Kä

Example: Glas Surface

Used in Facility Management for quantifying cleaning surfaces. Weirdly, IFC doesn't directly support a neat glas surface

Example: Lockable Window Catch

Not in IFC, needs manual entry depending on position in the building.

Example: Deliver parameters needing manual entry

3 IFC buildings:

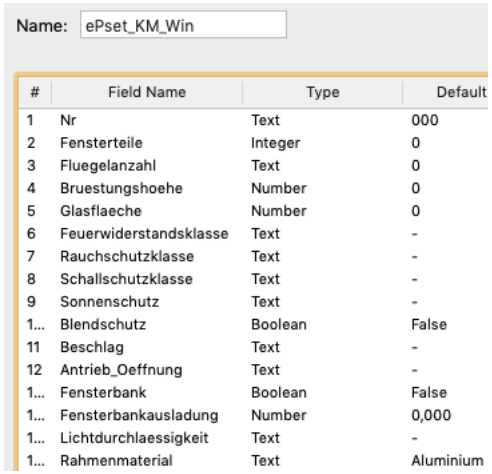
S 13 floors
A 5 floors
T 2 underground floors

921 Windows:

S 765
A 156
T -

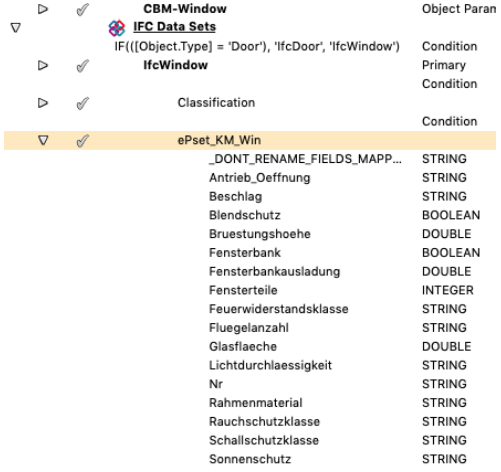
Chosen Workflow: implement controlling tools and helper script:

1- Create helper database (record in VW)
2- Create a linked IFC's custom pSet
3- Create a controlling table
4- Add some helpers to the overall accompanying script

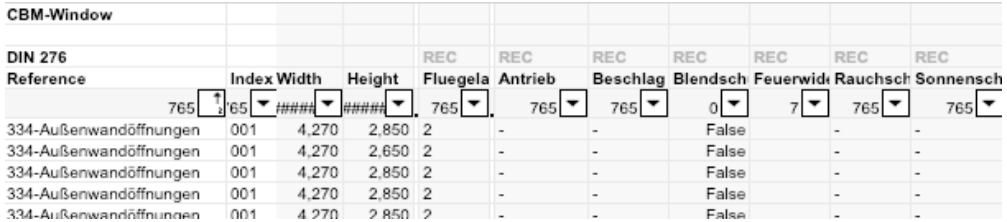


Name: ePset_KM_Win

#	Field Name	Type	Default
1	Nr	Text	000
2	Fensterteile	Integer	0
3	Fluegelanzahl	Text	0
4	Bruestungshoehe	Number	0
5	Glasflaeche	Number	0
6	Feuerwiderstandsklasse	Text	-
7	Rauchschutzklasse	Text	-
8	Schallschutzklasse	Text	-
9	Sonnenschutz	Text	-
1...	Blendschutz	Boolean	False
11	Beschlag	Text	-
12	Antrieb_Oeffnung	Text	-
1...	Fensterbank	Boolean	False
1...	Fensterbankausladung	Number	0,000
1...	Lichtdurchlaessigkeit	Text	-
1...	Rahmenmaterial	Text	Aluminium



	CBM-Window	Object Paran
	IFC Data Sets	
	IF(([Object.Type] = 'Door'), 'IfcDoor', 'IfcWindow')	Condition
	IfcWindow	Primary
		Condition
	Classification	
		Condition
	ePset_KM_Win	
	_DONT_RENAME_FIELDS_MAPP...	STRING
	Antrieb_Oeffnung	STRING
	Beschlag	STRING
	Blendschutz	BOOLEAN
	Bruestungshoehe	DOUBLE
	Fensterbank	BOOLEAN
	Fensterbankausladung	DOUBLE
	Fensterteile	INTEGER
	Feuerwiderstandsklasse	STRING
	Fluegelanzahl	STRING
	Glasflaeche	DOUBLE
	Lichtdurchlaessigkeit	STRING
	Nr	STRING
	Rahmenmaterial	STRING
	Rauchschutzklasse	STRING
	Schallschutzklasse	STRING
	Sonnenschutz	STRING



CBM-Window

DIN 276				REC	REC	REC	REC	REC	REC	REC
Reference	Index	Width	Height	Fluegela	Antrieb	Beschlag	Blendsch	Feuerwid	Rauchsch	Sonnensch
765	765	#####	#####	765	765	765	0	7	765	765
334-Außenwandöffnungen	001	4,270	2,850	2	-	-	False		-	-
334-Außenwandöffnungen	001	4,270	2,650	2	-	-	False		-	-
334-Außenwandöffnungen	001	4,270	2,850	2	-	-	False		-	-
334-Außenwandöffnungen	001	4,270	2,850	2	-	-	False		-	-
334-Außenwandöffnungen	001	4,270	2,850	2	-	-	False		-	-

```
{ ***************************** }
PROCEDURE SetCBMWindow(h: HANDLE);
    VAR
        config : DYNARRAY[] OF CHAR;
        styleN, str, recName : STRING;
        i, cnt : INTEGER;

    BEGIN
        IF GetObjectVariableBoolean(h, 700) = FALSE THEN BEGIN

            str := GetRField(h, 'CBM-Window', '3Don');

            IF GetClass(h) <> '30 BT-FENSTER-OBJ' THEN
                SetClass(h, '30 BT-FENSTER-OBJ');

            Set276(h);

            styleN := GetPluginStyle(h);
            IF Len(styleN) > 7 THEN
                SetRField(h, rec276, 'Index', Copy(styleN, 5, 3));

            IF GetRField(h, 'CBM-Window', 'Type') = 'Window' THEN
                recName := recIFCWin
            ELSE
                recName := recIFCtuer;

            SetRecord(h, recName);
            Rec_CleanUp(h, recName);

            cnt := 0;
            config := '';
            config := GetRField(h, 'CBM-Window', 'Config');
            i := 0;
            WHILE i < Len(config) DO BEGIN
                i := i +1;

                CASE config[i] OF
                '-', '?', 'P' :;
                'B' : cnt := cnt +2;
                OTHERWISE
                    cnt := cnt +1;
                END;
            END;

            SetRField(h, recName, 'Fluegelanzahl', Concat(cnt));

            gCnt := gCnt +1;
        END ELSE
            gError := gError +1;
    END;
```

Example: Deliver calculated parameters

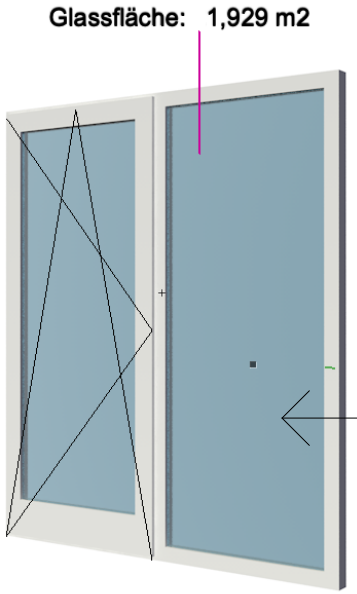
3 IFC buildings:

S 13 floors
A 5 floors
T 2 underground floors

921 Windows:

S 765
A 156
T -

Chosen Workflow: add functionality in Window Object
1- Add Glas Panes functionality
2- Add Glazing area and area fraction functionality
3- Add IFC automapping for the involved fields




```
PROCEDURE RSash3D;
    BEGIN
        IF CircleCircleInters(sOutL, gPioArcCen, sl, gO_rad +sOffL.y, p1, p2) THEN
            pL := p2;
        IF CircleCircleInters(sOutR, gPioArcCen, sr, gO_rad +sOffL.y, p1, p2) THEN
            pR := p1;

        BeginXtrd(gBotSash +gSW +gSB, gTopSash -gSW);
            ArcPolyRadial(pL, pR, gO -offY, gSH);
        EndXtrd;
        clipper := LNewObj;

        BeginXtrd(gBotSash, gTopSash);
            ArcPolyRadial(sOutL, sOutR, gO -offY, gSH);
        EndXtrd;
        obj := LNewObj;
```

```
    'B': { biparting simmetric }
        BEGIN
            Jamb2D(jOutL, jl, gJH, 1, TRUE, g3DON); { left }
            Jamb2D(jOutR, jr, gJH, -1, TRUE, g3DON); { right }

            IF optionDoor = FALSE THEN BEGIN
                BeginGroupN(gSashGr);
                    Sash2D(jOutL, sOffL, sl, 1, gThisOut); { left }
                    SashMid_Prof(gElemCen); { left part of the middle profile }
                    Sash2D(jOutR, sOffR, sr, -1, gThisOut); { right }
                EndGroup;
            END;

            IF gIsRoundWin THEN BEGIN
                BeginGroup;
                GlasPane(sInL, _V(gElemCen.x -sg_SmidW, sInR.y),
                    -glOut, gSH, gGlasSZ -botOff, gGlasSDZ +botOff, NOT optionDoor
                EndGroup;
                H_MirrorGroup(LNewObj, TRUE, gElemCen, gPioArcCen);

            END ELSE BEGIN
                GlasPane(sInL, _V(gElemCen.x -sg_SmidW, sInR.y),
                    -glOut, gSH, gGlasSZ -botOff, gGlasSDZ +botOff, NOT optionDoor
                GlasPane(_V(gElemCen.x +sg_SmidW, sInL.y), sInR,
                    -glOut, gSH, gGlasSZ -botOff, gGlasSDZ +botOff, NOT optionDoor
            END;

            IF g3DON THEN BEGIN
                Jamb3D;
                Sash3D(sOutL, gElemCen, sl, sg_SmidW);
                Sash3D(gElemCen, sOutR, sg_SmidW, sr);
            END;
        END;

    OTHERWISE { '>', '<' }
        BEGIN
            Jamb2D(jOutL, jl, gJH, 1, TRUE, g3DON); { left }
            Jamb2D(jOutR, jr, gJH, -1, TRUE, g3DON); { right }

            BeginGroupN(gSashGr);
                Sash2D(jOutL, sOffL, sl, 1, gThisOut); { left }
                Sash2D(jOutR, sOffR, sr, -1, gThisOut); { right }
            EndGroup;
```

```
IF IFC_DMAddPSetFld(gPio_N, entryN, pSetN, 'ThermalTransmittance', cIfcReal, TRUE, TRUE, TRUE) THEN
    temp_b := IFC_DMSetPSetFldMap(gPio_N, entryN, pSetN, 'ThermalTransmittance', '[Object.UValue]');

IF IFC_DMAddPSetFld(gPio_N, entryN, pSetN, 'GlazingAreaFraction', cIfcReal, TRUE, TRUE, TRUE) THEN
    temp_b := IFC_DMSetPSetFldMap(gPio_N, entryN, pSetN, 'GlazingAreaFraction', '[Object.GlazingAreaFraction]');
    { Glasflächenanteil: Anteil der verglasten Fläche an der Gesamtfläche des Fensters.
    Es ist der Reziprokwert des Rahmenanteils. }

IF IFC_DMAddPSetFld(gPio_N, entryN, pSetN, 'SmokeStop', cIfcBoo, TRUE, TRUE, TRUE) THEN
    temp_b := IFC_DMSetPSetFldMap(gPio_N, entryN, pSetN, 'SmokeStop', '[Object.Smoke Stop]');
```