



زمین مرجع کردن ترسیمات در
Revit
و
نمایش آن‌ها در پلت فرم CesiumJS
با بکارگیری استانداردهای
CityGML (LOD3) و ۳D Tiles

انجمن ملی
زیرساخت‌های اطلاعات مکانی

نگارنده: علیرضا امیری
(کارشناس سامانه‌های اطلاعات مکانی)

مهر ۱۴۰۲

فهرست مطالب

۱- مقدمه	۱
۲- نگاهی به استاندارد CityGML	۲
۳- زمین مرجع (ژئورفرنس) کردن ترسیمات سه بعدی	۳
۳-۱- زمین مرجع کردن نقشه در Revit با استفاده از مختصات UTM معلوم یک نقطه (روش اول).....	۳
۳-۲- زمین مرجع کردن نقشه در Revit با استفاده از Shapefile دارای مختصات UTM معلوم (روش دوم).....	۱۴
۴- تبدیل فایل Revit به فرمت IFC، CityGML و ۳D و نمایش آن در CeisumJS	۲۴
۴-۱- تبدیل فرمت Revit به فرمت IFC.....	۲۴
۴-۲- تبدیل فرمت IFC به CityGML.....	۲۵
۴-۳- تبدیل فرمت CityGML به ۳D۳D	۳۳

۱- مقدمه

در این دستورالعمل، ابتدا نحوه زمین مرجع کردن ساختمانی که نرم افزار Revit ترسیم شده بیان می شود. سپس نحوه تبدیل این فایل ترسیم شده در Revit - که فرمت آن .rvt است- به فرمت های CityGML و IFC با استفاده از نرم افزار FME توضیح داده می شود. در نهایت، روش نمایش فایل با فرمت IFC، در نرم افزار تحت وب CesiumJS شرح داده می شود. IFC، یک ساختار و فرمت داده مکانی است که شامل مجموعه ای از تایل ها است که برای رندر کردن سریع داده های سه بعدی و در اختیار قرار دادن آنها در بستر وب است. این داده ها می توانند شامل داده های فتوگرامتری؛ ساختمان های سه بعدی، BIM/CAD و ابر نقطه باشد. در این بخش از دستورالعمل، برای آشنایی با نرم افزارهایی که بکار گرفته شده اند، توضیح مختصری راجع به هر کدام از آنها آورده شده است.

نرم افزار Revit، ابزاری قدرتمند برای مدل سازی سه بعدی و ترسیم جزئیات ساختمانی است که از قدرت بالایی در ترسیمات پیچیده مهندسی برخوردار است. این نرم افزار این امکان را می دهد که علاوه بر ترسیم جنبه های معماری ساختمان مانند کف، سقف، درب و دیوار، جنبه های دیگر اعم از تاسیسات، برق، گاز، آب و دیگر اجزای ساختمان را به صورت سه بعدی مدل کرد. این نرم افزار که محصول شرکت Autodesk است، جزو نرم افزارهای BIM (مدل سازی اطلاعات ساختمان) دسته بندی می شود.

نرم افزار FME که هم نسخه Desktop و هم Server آن موجود است، نرم افزاری است که توسط شرکت Safe Software ارائه شده است. این نرم افزار ابزاری است که برای استخراج، پالایش، تبدیل فرمت و بارگذاری اطلاعات در زمان ساخت انبار داده ها استفاده می شود، به این صورت که اطلاعات را از یک یا چند منبع مختلف جمع آوری پالایش و تبدیل فرمت کرده و در نهایت، در انبار داده بارگذاری می کند. در این نرم افزار تمامی عملیات بدون نیاز به کد نویسی و تنها با استفاده از کشیدن و رها کردن قابل انجام هستند و می توان سریع، راحت و بدون نوشتن حتی یک خط کد بیش از ۴۷۵ نوع از فرمت داده ها از جمله GIS، CAD، Spatial databases & Databases، Spreadsheets، Cloud & Web services، XML، Rasters، Vectors، Point clouds، IFC را بین اپلیکیشن های مختلف جا به جا کرده، ویرایش و تفسیر کرد. با نرم افزار FME می توان داده های حجیم نظیر داده های مکانی را ایجاد کرد و سپس آنها را در نرم افزار CesiumJS با سطوح بالایی از جزئیات نمایش داد.

CesiumJS نیز، یک کتابخانه یا یک محیط اجرایی اجاوا اسکریپتی متن باز است که برای مشاهده (ویژوالیزاسیون) و تحلیل داده های دوبعدی و سه بعدی در یک مرورگر وب بکار می رود. یعنی بعد از اینکه داده ها آماده سازی و تایل شدند (به IFC تبدیل شدند)، می توان از پلت فرم قدرتمند CesiumJS استفاده کرد و داده های IFC را مشاهده (ویژوالیزه) کرد. فرمت های ورودی به Cesium می تواند شامل موارد زیر باشد:

- glTF (.gltf, .glb)
- CityGML (.citygml, .xml, .gml)
- KML/COLLADA (.kml, .kmz)
- LASer (.las, .laz)
- COLLADA (.dae)
- Wavefront OBJ (.obj)

مرجع: <https://olosh.dev/library/21913>

۲- نگاهی به استاندارد CityGML

CityGML یک استاندارد بین المللی برای ممکن ساختن نمایش قابل تعامل و تبادل مدل های مجازی سه بعدی شهرها و مناظر است. به بیان دیگر، CityGML، یک مدل داده استاندارد باز است و همچنین یک فرمت تبادل داده نیز محسوب می شود که ذخیره سازی مدل های سه بعدی رقومی شهرها و مناظر را انجام می دهد و توسط OGC ارائه شده است. این مدل داده، هندسه سه بعدی، توپولوژی، معنا و ظاهر اشیای محیط های شهری را تعریف می کند. این تعاریف، در سطوح مختلف از جزئیات^۱ (مدل چند رزولوشنی) ارائه می شود. CityGML مبتنی بر زبان GML است که یک مدل هندسی استاندارد ارائه می دهد. CityGML قابلیت تعامل پذیری داده ها را در زمینه خدمات وب مکانی و زیرساخت های داده های مکانی تسهیل می کند.

این استاندارد، عوارض شهری را به چند طبقه تقسیم بندی می کند که یکی از این طبقات، سازه ها (مانند ساختمان ها، پل ها و تونل ها) می باشند. در این دستورالعمل به دلیل اهمیت این طبقه و پرهیز از طولانی شدن بحث، صرفاً به کلاس عارضه ساختمان و نحوه تبدیل و نمایش آن در بستر وب پرداخته شده است.

CityGML از ۵ سطح از جزئیات (LoD) پشتیبانی می کند. در سطح اول یا LoD1، یک ساختمان به صورت یک منشور یا مکعب مستطیل (بلوک) ساده نمایش داده می شود و به هیچ وجه جزئیات ساختمان مدل نمی شوند. یعنی قسمت های دارای حجم و مساحت پوسته بیرونی ساختمان، با بدنه ساختمان ادغام می شوند. در LoD1، پشت بام ها به صورت تخت، مدل می شوند.

در سطح دوم یا LoD2، پوسته بیرونی ساختمان و اجزای آن یعنی عوارض کوچکتر ساختمان (تاسیسات و تجهیزات بیرونی ساختمان) نه تنها از نظر هندسی، بلکه به صورت معنایی مدل می شوند. مثال های بارز این عوارض عبارتند از دودکش ها، پنجره های روی شیروانی، تراس ها، پله های فرار. ساختار سقف ساختمان در LoD2 پیشرفته تر است و از هندسه سقف های شیب دار و شیروانی ها پشتیبانی می کند.

LoD3، مشابه با LoD2 است، اما LoD3 برای رسیدن به دقت هندسی بالاتر و عوارض با ابعاد کوچکتر مانند آنتن ها و ناودان های روی نمای ساختمان و نیز مدل کردن درب ها و پنجره های ساختمان مورد استفاده قرار می گیرد. در LoD3 در مورد سقف ساختمان، می توان حتی بیرون زدگی های جزئی شیروانی از کناره های ساختمان را نیز مدل کرد.

^۱- Level-of-Detail (Multiresolution model)

LoD4 در مورد پوسته خارجی ساختمان، تفاوتی با LoD3 ندارد، لیکن LoD4 این امکان را فراهم کرده که ساختار داخلی یک ساختمان با اضافه شدن دیوارها، پله‌ها، دکوراسیون و مبلمان داخلی مدل شود. همچنین در همه LoDها، اطلاعات ظاهری و نما مانند جنس مصالح و بافت‌های با رزولوشن بالا را می‌توان بر روی سازه‌ها نگاشت. توضیحات فوق را می‌توان بطور خلاصه در شکل زیر ملاحظه کرد.



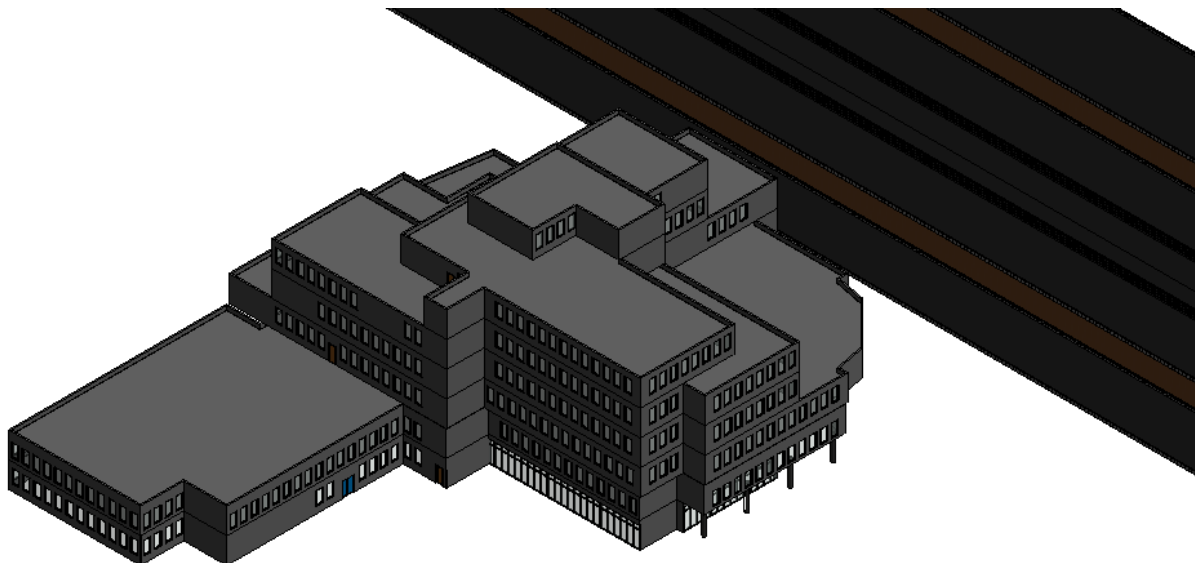
مرجع: <http://www.opengis.net/spec/citygml/2.0>

۳- زمین مرجع (ژئورفرنس) کردن ترسیمات سه بعدی

ترسیمات سه بعدی که در Revit انجام می‌شوند، زمین مرجع نیستند و پس از نمایش آنها در بستر CesiumJS، روی محل واقعی خود قرار نمی‌گیرند. لذا باید آنها را قبل از تبدیل به CityGML یا CityJSON، ژئورفرنس کرد. در اینجا فرض بر این است که می‌خواهیم یک تک ساختمان را زمین مرجع کنیم و در این دستورالعمل، دو روش را برای انجام این کار شرح خواهیم داد.

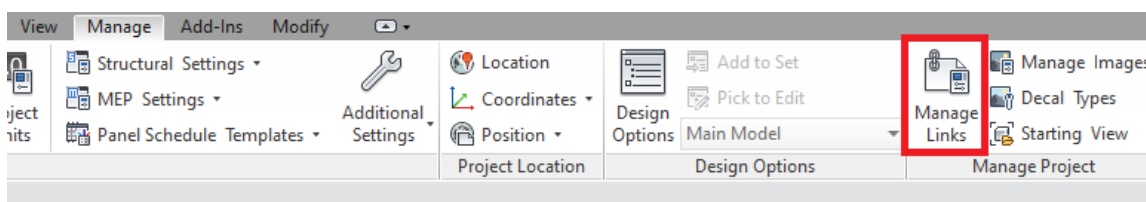
۳-۱- زمین مرجع کردن نقشه در Revit با استفاده از مختصات UTM معلوم یک نقطه (روش اول)

فرض بر این است که یک تک ساختمان در Revit ترسیم شده است و باید زمین مرجع شود. همچنین مختصات UTM (Zone 39) حداقل یکی از گوشه‌های این تک ساختمان معلوم است. در این شرایط می‌توان این تک ساختمان را با طی چند مرحله، زمین مرجع نمود. البته برای زمین مرجع نمودن یک عارضه (یا نقشه)، باید مختصات حداقل دو نقطه از آن عارضه، معلوم باشد، اما در اینجا فرض بر این است که دیوارهای شرقی و غربی ساختمان موردنظر در جهت شمال جغرافیایی قرار دارند و با یک نقطه، عمل زمین مرجع از دقت کافی برخوردار خواهد بود. نمونه‌ای ارائه شده در این بخش که قرار است زمین مرجع شود، ساختمان مرکزی شهرداری منطقه ۹ تهران واقع در خیابان آزادی، سر بلوار استادمعین است که در Revit ترسیم شده است (شکل زیر).

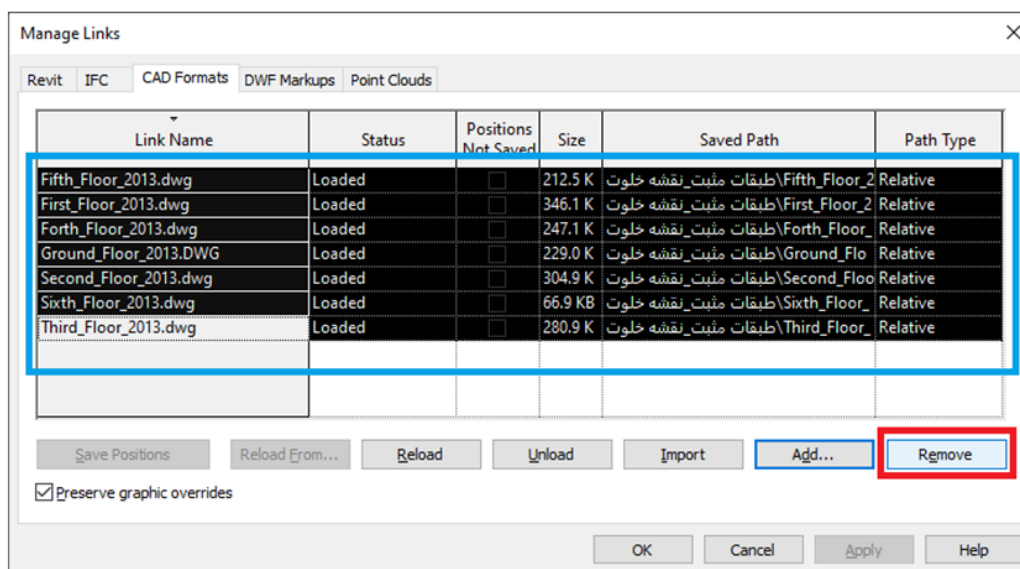


مراحل انجام این کار به شرح ذیل می‌باشند:

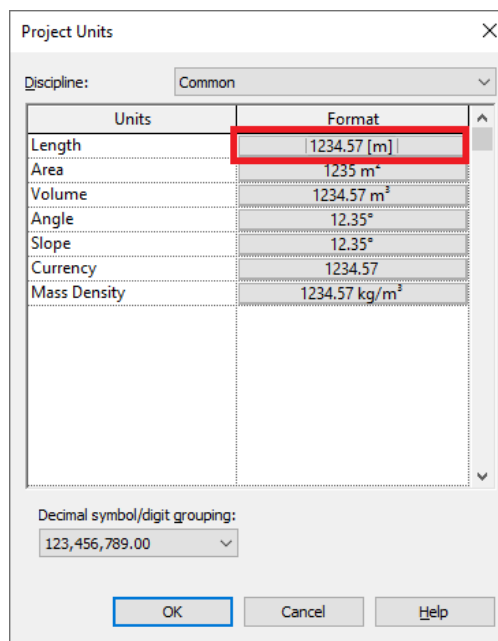
۱- فایل RVTی مربوط به تک ساختمان را در Revit باز می‌کنیم و از طریق تب Manage و سپس کلیک بر روی آیکن Manage Links (کادر قرمز شکل زیر)، اقدام به حذف کلیه فایل‌های اتوکدی که قبلاً به فایل RVT، لینک شده‌اند، می‌کنیم.



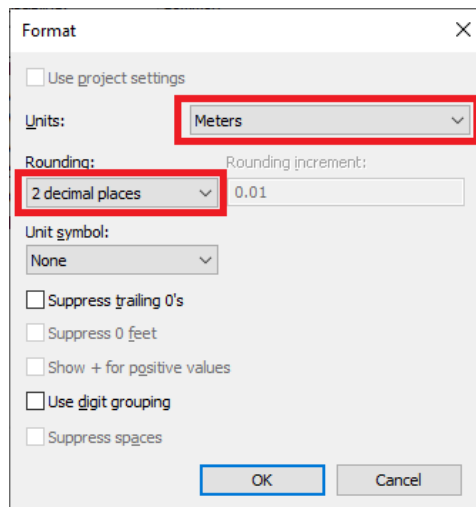
نحوه حذف کردن فایل‌های اتوکدی به این نحو است که بعد از باز شدن پنجره Manage Links، ردیف‌های مربوط به فایل‌های اتوکد لینک شده را انتخاب کرده (کادر آبی شکل زیر) و سپس با کلیک بر روی دکمه Remove (کادر قرمز شکل زیر) و OK کردن پنجره، اقدام به حذف آنها می‌کنیم.



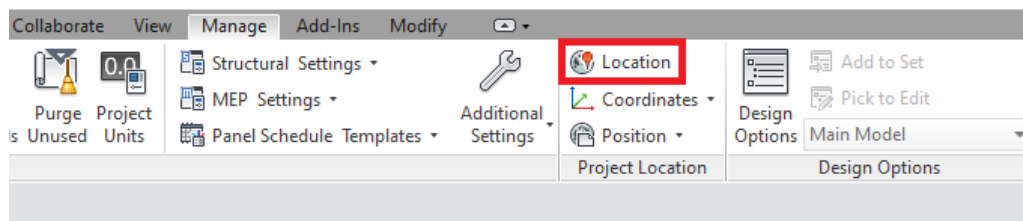
۲- از آنجائیکه سیستم مختصات موردنظر در فرآیند زمین مرجع کردن، UTM است، واحد در Revit روی "متر" تنظیم می شود. لذا با تایپ عبارت UN در محیط Revit، پنجره Project Units را باز می کنیم (شکل زیر).



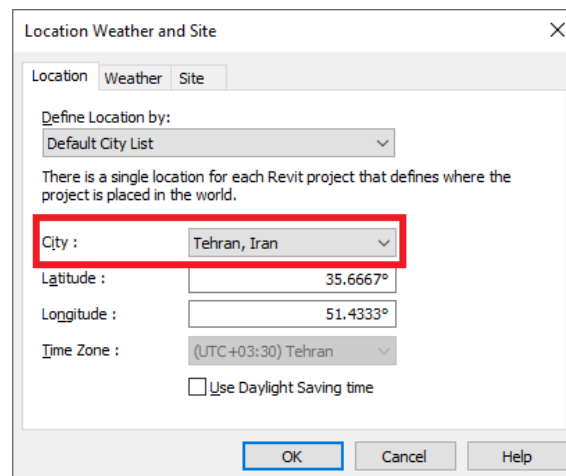
با کلیک روی کادر قرمز شکل بالا، پنجره Format باز می شود (شکل زیر). از کامبوباکس Units، گزینه Meters و در قسمت Rounding، گزینه 2 decimal places را انتخاب می کنیم. با OK کردن پنجره های Project Units و Format، واحد، روی "متر" و ارقام اعشار آن روی دو رقم تنظیم می شود.



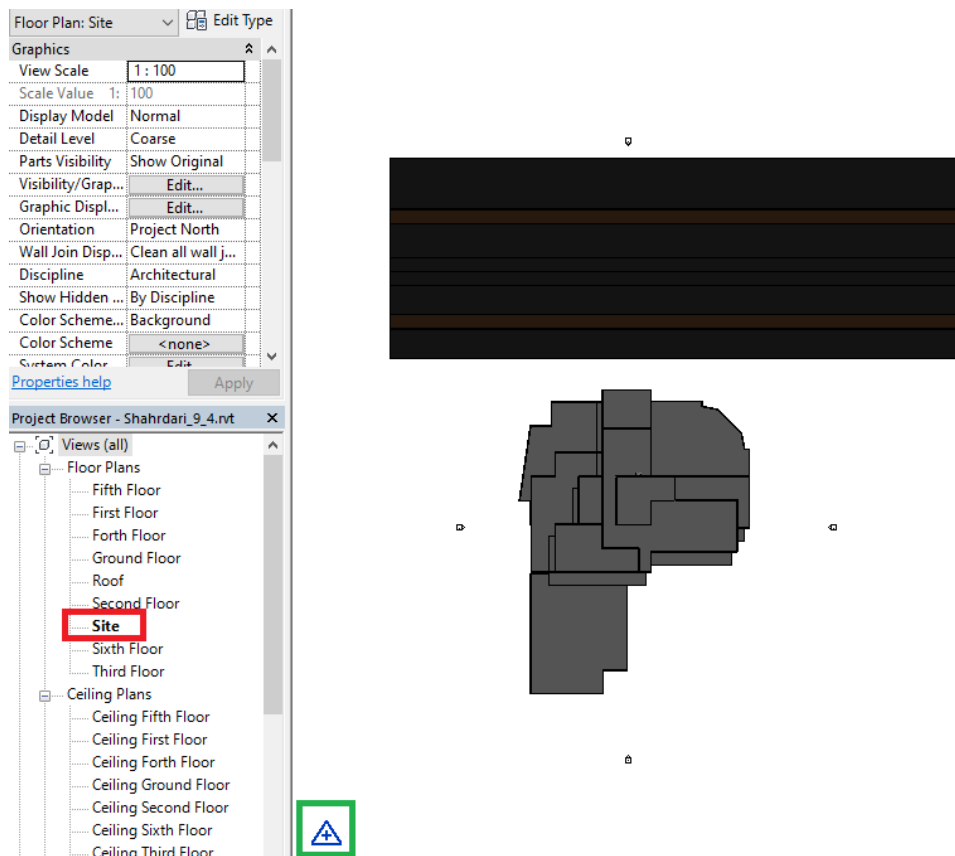
۳- بهتر است قبل از اقدام به زمین مرجع کردن، محل جغرافیایی در Revit را روی کشور ایران تنظیم کنیم. برای انجام این کار، در تب Manage، روی آیکن Location (کادر قرمز شکل زیر)، کلیک می کنیم.



بعد از باز شدن پنجره زیر، کامبوباکس City را روی Tehran, Iran تنظیم می کنیم.

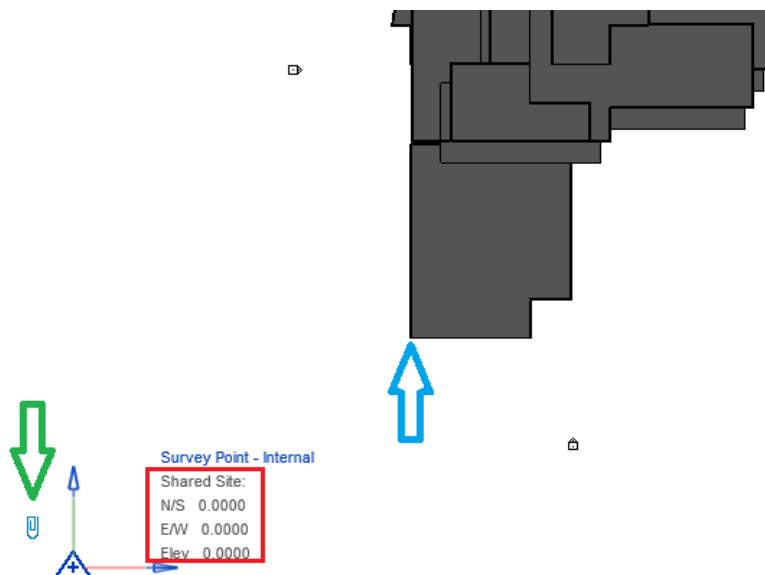


۴- اگر در Revit، روی نمای Site (کادر قرمز شکل زیر)، دابل کلیک کنیم، شکلی مشابه شکل زیر قابل مشاهده است. اگر کمی با zoom out، نقشه را کوچکتر کنیم، Survey Point مشخص می شود (کادر سبز شکل زیر).

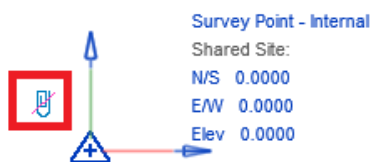


Surey Point، یک نقطه مرجع در Revit است که مبدا و توجیه سیستم مختصات پروژه یا نقشه را تعریف می‌کند. در واقع، این نقطه، پل ارتباطی میان مدل ساختمان ترسیم شده (با مختصات محلی) و دنیای واقعی (با مختصات صحیح) است. Survey Point را هم می‌توان به صورت دستی و هم از طریق لینک کردن یک فایل DWG (با مختصات صحیح) به Revit، تنظیم کرد که در این بخش، مختصات آن را به صورت دستی، معرفی می‌کنیم. اما در بخش بعد (روش دوم زمین مرجع کردن فایل Revit)، از یک فایل DWG برای تعریف مختصات این نقطه استفاده خواهیم نمود.

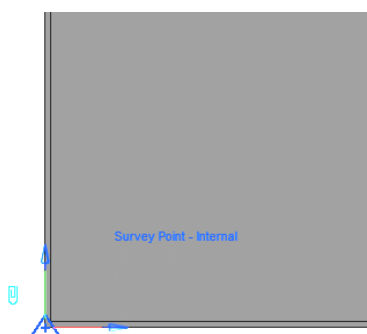
۵- Survey Point را به نقطه‌ای از ساختمان که مختصات UTM آن را در اختیار داریم، منتقل می‌کنیم. فرض بر این است که این نقطه، گوشه سمت چپ و پایین ساختمان شهرداری است (فلش آبی شکل زیر). بدین منظور، با درگ ماوس، Survey Point را Select می‌کنیم. با Select شدن آن، مختصات فعلی این نقطه، یعنی (۰،۰،۰)، نمایش داده می‌شود (کادر قرمز شکل زیر) و علامت یک سنجاق در کنار آن ظاهر می‌شود (فلش سبز رنگ شکل زیر). علامت سنجاق بیانگر این موضوع است که این نقطه، اصطلاحاً در حالت Clipped است و قابل جابجا شدن (move) نیست.



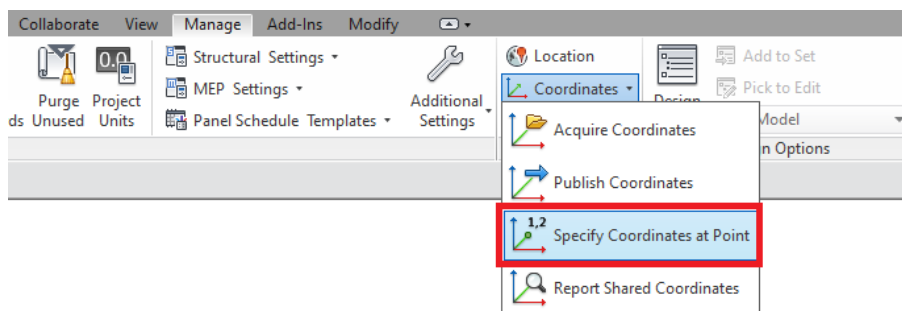
لذا ابتدا با کلیک روی علامت سنجاق، آن را Unclipped می‌کنیم تا از حالت قفل خارج شده و قابل جابجا شدن باشد. در اینصورت، یک خط قرمز مورب بر روی آن ظاهر می‌شود (کادر قرمز شکل زیر). اکنون Survey Point آماده انتقال است.



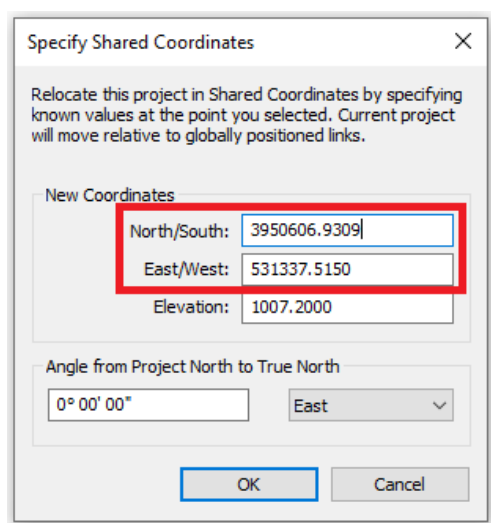
سپس با دستور move، Survey Point را انتقال داده و به گوشه سمت چپ و پایین ساختمان ترسیم شده منتقل می‌کنیم (شکل زیر) و مجدداً با کلیک روی علامت سنجاق، آن را Clipped می‌کنیم.



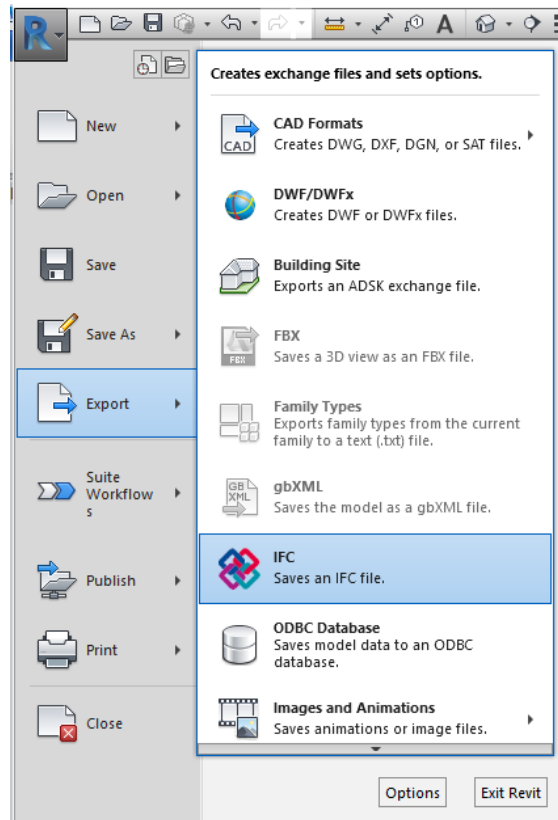
۶- با کلیک روی دستور Specify Coordinates at Point که در تب Manage (کادر قرمز شکل زیر) قرار دارد، می‌توان پروژه را به مکان صحیحش منتقل کرد یا به عبارت دیگر، زمین مرجع نمود.



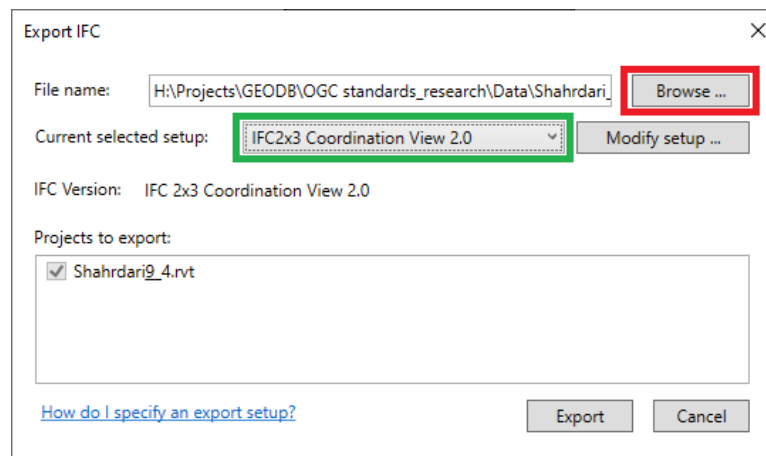
با کلیک بر روی Survey Point، پنجره Specify Shared Coordinates (شکل زیر) باز می‌شود که مختصات صحیح UTM گوشه ساختمان را در آن درج می‌کنیم (کادر قرمز شکل زیر) و OK می‌کنیم. مشاهده می‌کنیم که مکان Survey Point عوض می‌شود. وارد کردن ارتفاع یا Elevation در این قسمت، به دلیل اینکه صرفاً زمین مرجع کردن پلانیمتریک مد نظر است، ضروری نیست.



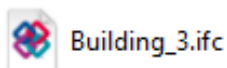
۷- از پروژه یا ساختمان در Revit، به فرمت IFC اکسپورت می‌گیریم (مطابق شکل زیر).



بعد از کلیک روی دستور IFC در شکل بالا، پنجره زیر باز می‌شود. با کلیک روی دکمه Browse (کادر قرمز شکل زیر)، مسیر و نام فایل خروجی با فرمت IFC را مشخص می‌کنیم و در کامیوباکس Current selected setup (کادر سبز شکل زیر)، گزینه IFC2x3 Coordination View 2.0 را که معمولاً در تبدیلات استفاده می‌شود، انتخاب می‌کنیم.



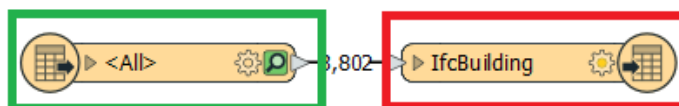
با کلیک بر روی دکمه Export، فایل خروجی با فرمت IFC ایجاد خواهد شد.



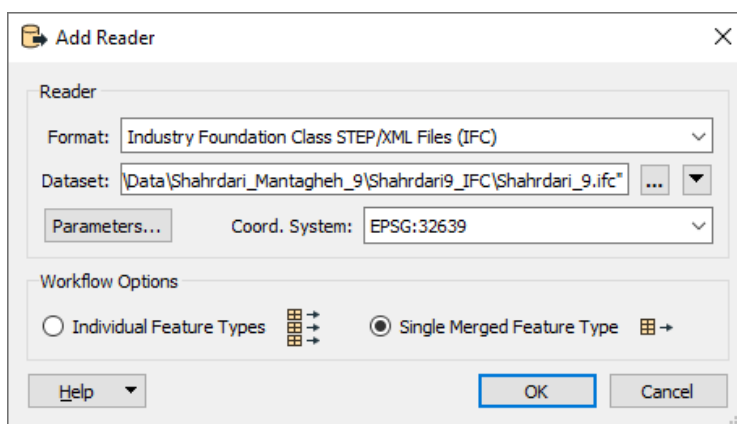
مرجع:

<https://help.bimtrack.co/hc/en-ca/articles/360041522712-Export-a-Revit-Model-to-IFC>

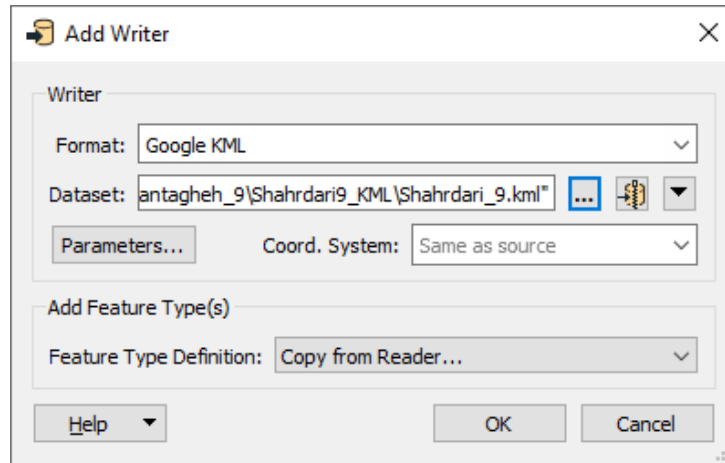
۹- سپس برای کنترل صحت نتیجه فرآیند زمین مرجع کردن و حصول اطمینان از صحت مختصات در فایل IFC تولید شده و قرار گرفتن آن در مکان درست خود، فایل IFC را در نرم افزار FME به فرمت KML تبدیل می کنیم. بدین منظور یک Reader (کادر سبز رنگ شکل زیر) و یک Writer (کادر قرمز شکل زیر) را در FME فراخوانی می کنیم و مدل را در FME به صورت زیر ایجاد می کنیم. Reader را از منوی Readers و سپس دستور Add Reader فراخوانی می کنیم. بطور مشابه، فراخوانی یک Writer نیز از منوی Writers و سپس، کلیک روی دستور Add Writer انجام قابل انجام است.



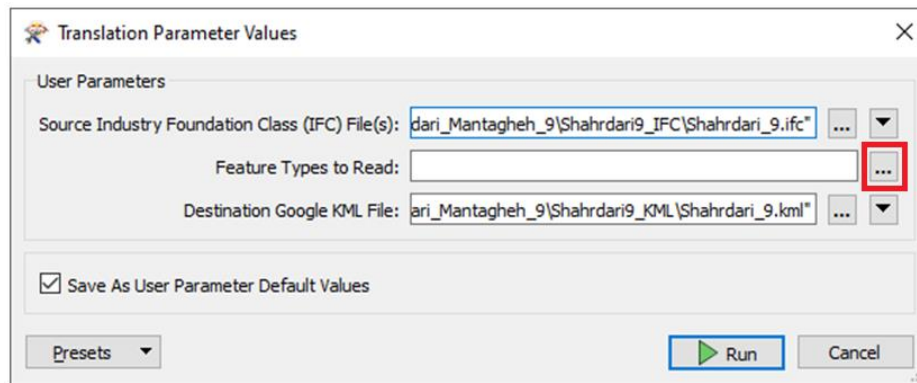
به محض کلیک روی دستور Add Reader پنجره شکل زیر باز می شود. در قسمت Format، گزینه Industry Foundation Class STEP/XML Files (IFC) را انتخاب می کنیم. در قسمت Dataset نیز مسیر و نام فایل IFC را مشخص می کنیم. در قسمت Coord. System، سیستم مختصات UTM (Zone 39) را باید انتخاب کنیم که گزینه EPSG:32639 متناظر با سیستم مختصات مزبور است. در نهایت، روی آپشن Single Merged Feature Type کلیک می کنیم و پنجره زیر را OK می کنیم.



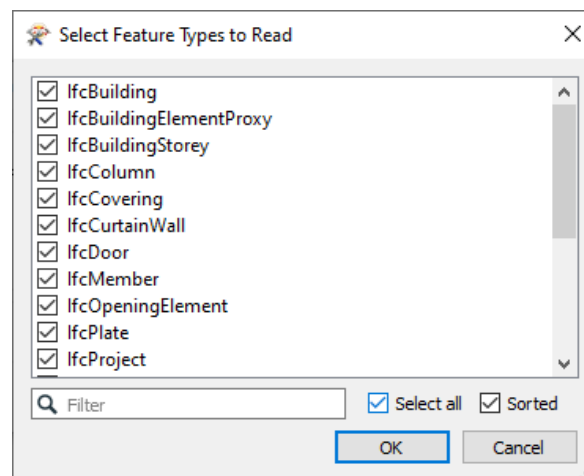
بعد از اضافه کردن Writer نیز پنجره زیر باز می شود که در قسمت Format، گزینه Google KML را انتخاب می کنیم. در قسمت Dataset نیز مسیر و نام فایل خروجی (فایل KML) را مشخص می کنیم و OK می کنیم.



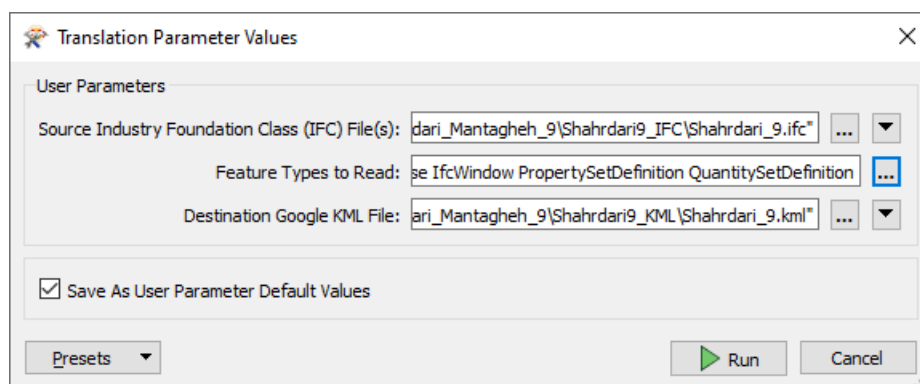
بعد از کلیک بر روی اجرای مدل در FME (دکمه Run)، پنجره زیر باز می‌شود.



با کلیک روی دکمه سه نقطه گزینه 'Feature Types to Read' (کادر قرمز شکل بالا)، پنجره زیر باز می‌شود. در این پنجره، چک باکس 'Select all' را فعال می‌کنیم (با فرض اینکه کلیه آبجکت‌های فایل IFC برای تبدیل شدن به KML مدنظر باشند).



با تایید پنجره فوق، مجدداً به پنجره زیر برمی‌گردیم. با کلیک روی دکمه Run، مدل اجرا شده، تبدیل فرمت از IFC به KML انجام می‌شود و فایل KML خروجی ایجاد می‌شود.



۱۷- فایل KML ایجاد شده را در نرم‌افزار Google Earth باز می‌کنیم و چک می‌کنیم که در در سر جای درست خود قرار گرفته باشد. همانطور که مشخص است، ساختمان شهرداری در محل صحیح خود (تقاطع خیابان آزادی و بلوار استادمعین) قرار گرفته است (شکل زیر). می‌توان باز کردن فایل KML در Google Earth را از مسیر File Menu>Import انجام داد.

حال می‌توان فایل IFC ایجاد شده در مرحله قبل که از صحت زمین‌مرجع بودن آن اطمینان حاصل شده به فرمت CityGML و سپس به فرمت ۳D تبدیل نمود و در CesiumJS مشاهده کرد. همچنین می‌توان مستقیماً فایل IFC را به ۳D تبدیل نمود که ما در اینجا، همین کار را انجام خواهیم داد.

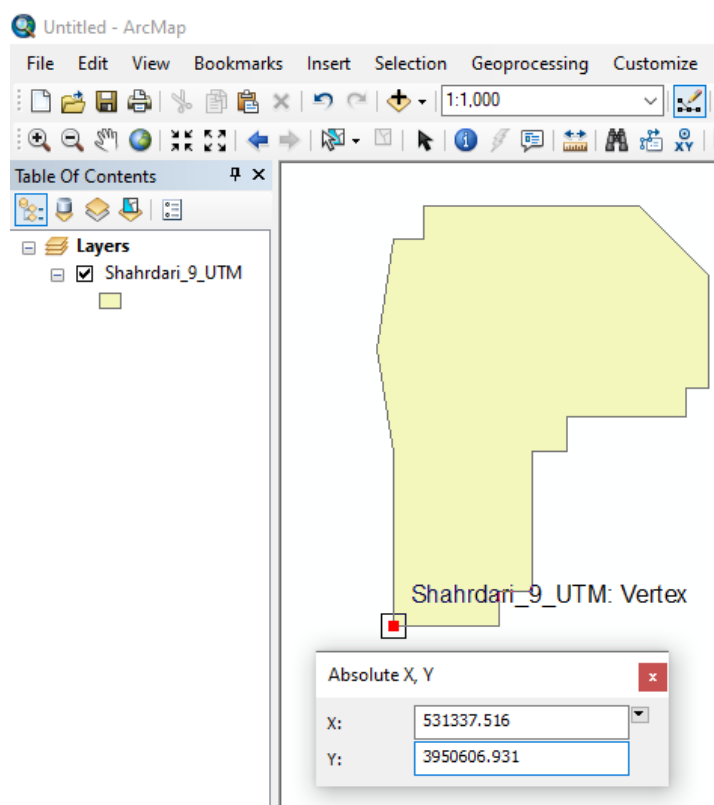


۲-۳- زمین مرجع کردن نقشه در Revit با استفاده از Shapefile دارای مختصات UTM

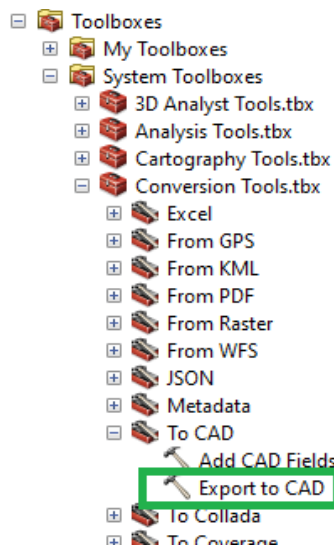
معلوم (روش دوم)

مجدداً فرض بر این است که تک ساختمان مربوط به شهرداری منطقه ۹ تهران در Revit ترسیم شده و در اختیار است و باید زمین مرجع شود. همچنین Shapefile پای این تک ساختمان که زمین مرجع شده و مختصات صحیح UTM دارد، موجود است (شکل زیر). در این شرایط می توان این تک ساختمان را با طی چند مرحله، زمین مرجع نمود. مراحل انجام این کار به شرح ذیل می باشند:

۱- ابتدا Shapefile مربوط به پای ساختمان را در ArcGIS باز می کنیم و صحت نقطه یا نقاط موردنظر از گوشه های ساختمان را که قرار است به عنوان نقاط رفرنس، به منظور زمین مرجع کردن، مورد استفاده قرار گیرند، کنترل می کنیم (شکل زیر).

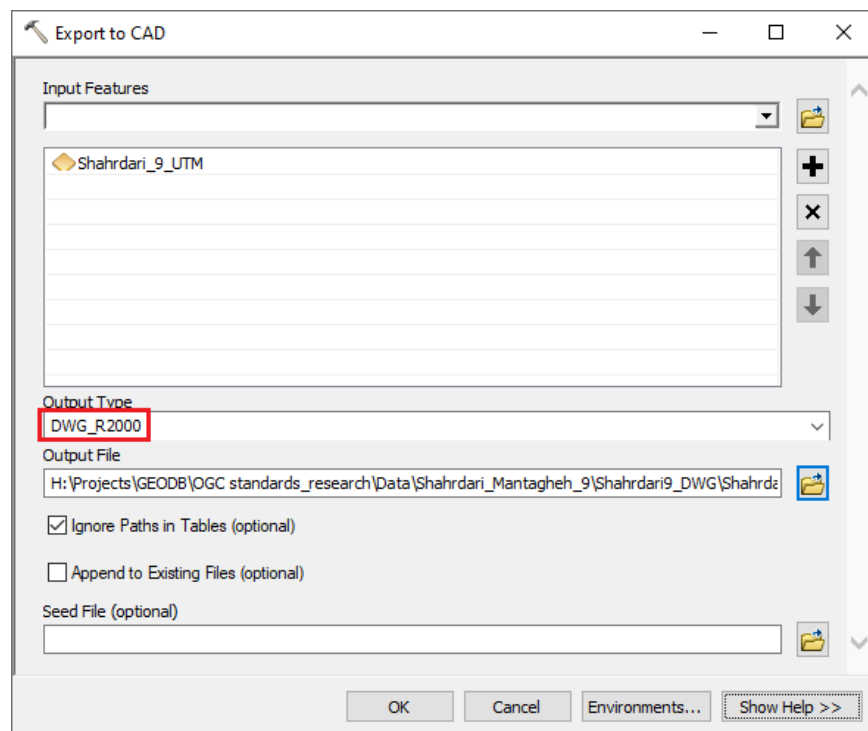


۲- با ابزار Export to CAD، شیپ فایل مذکور را به فرمت DWG تبدیل می کنیم (کادر سبز شکل زیر). مسلماً این فایل DWG نیز دارای مختصات صحیح خواهد بود.



نکته: برای پرهیز از بروز اشکال در باز شدن فایل DWG در Revit، نباید نسخه این فایل، بالاتر از نسخه Revit باشد. لذا در پنجره ابزار Export to CAD، نسخه فایل DWG را روی نسخه ۲۰۰۰ تنظیم کردیم (کادر قرمز شکل زیر). می‌توان پایین آوردن نسخه فایل DWG را بعد از تولید شدن آن، از طریق نرم‌افزار اتوکد نیز انجام داد.

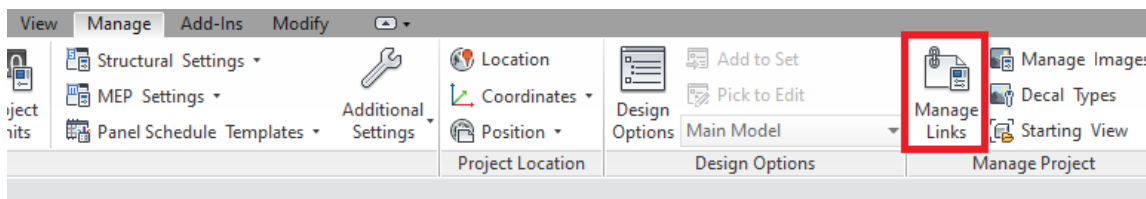
نکته: توصیه می‌شود واحد اندازه‌گیری در فایل DWG در نرم‌افزار اتوکد بر روی "متر" تنظیم شده باشد (با دستور UN).



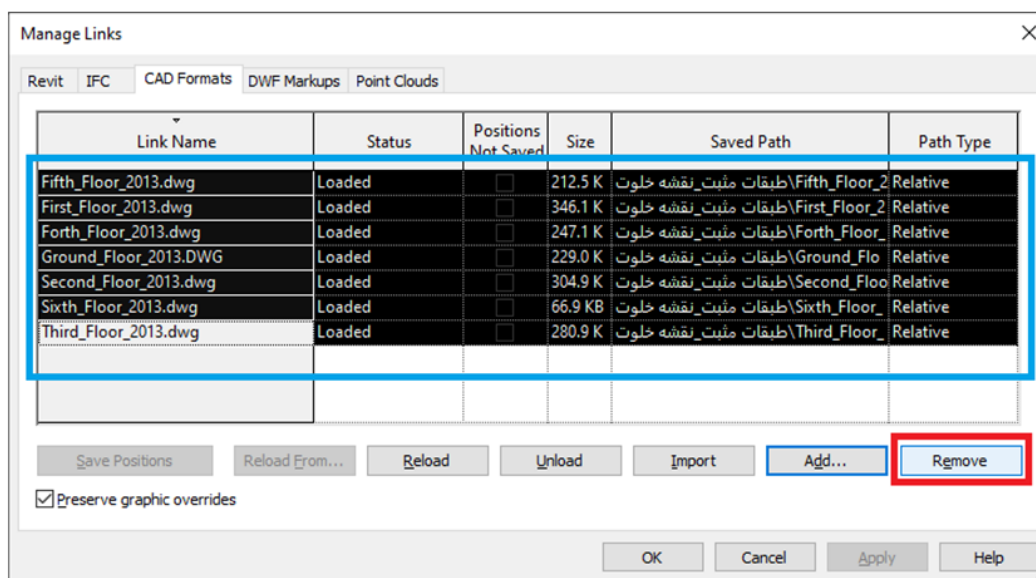
چنانچه نسخه فایل DWG بالاتر از نسخه Revit باشد، در هنگام لینک کردن این فایل به Revit، پیغام خطای زیر نمایش داده خواهد شد:

Some elements in the CAD file could not imported into Revit, such as ActiveX or proprietary components that is not recognized by Revit.

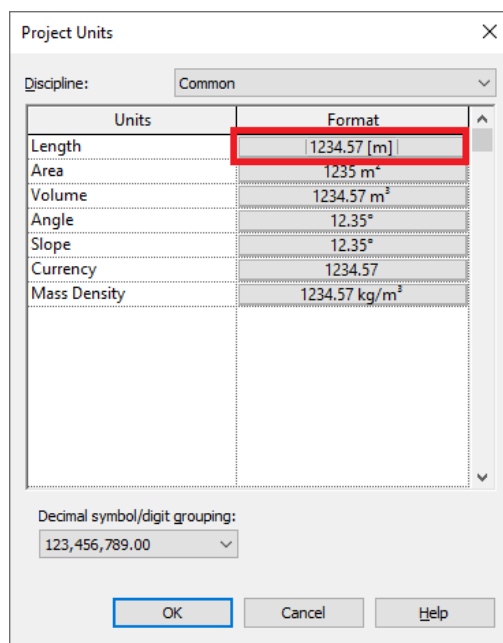
۳- فایل RVT ی مربوط به تک ساختمان را در Revit باز می کنیم و از طریق تب Manage و سپس کلیک بر روی آیکن Manage Links (کادر قرمز شکل زیر)، اقدام به حذف کلیه فایل های اتوکدی که قبلاً به فایل RVT لینک شده اند، می کنیم.



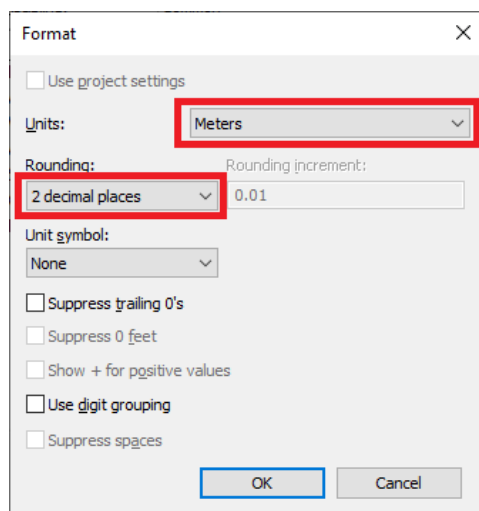
به منظور حذف فایل های اتوکد لینک شده، بعد از باز شدن پنجره Manage Links، ردیف های مربوط به فایل های مزبور را انتخاب کرده (کادر آبی شکل زیر) و سپس با کلیک بر روی دکمه Remove (کادر قرمز شکل زیر) و OK کردن پنجره زیر، اقدام به حذف آنها می کنیم.



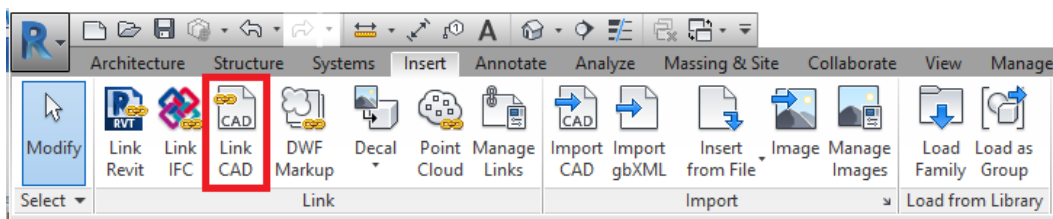
۴- از آنجاییکه سیستم مختصات Shapefile اولیه و به تبع آن، فایل DWG، UTM است، واحد در Revit باید روی "متر" تنظیم شود. لذا با تایپ عبارت UN در محیط Revit، پنجره Project Units را باز می کنیم (شکل زیر).



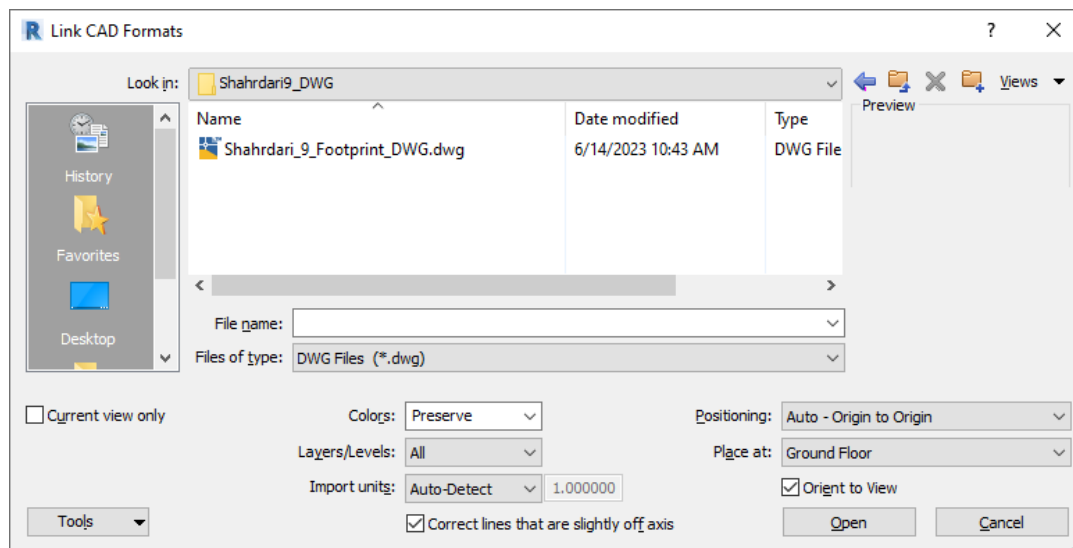
با کلیک روی کادر قرمز شکل بالا، پنجره Format باز می‌شود (شکل زیر). از کامبوباکس Units، گزینه Meters و در قسمت Rounding، گزینه 2 decimal places را انتخاب می‌کنیم. به این ترتیب، واحد روی "متر" و ارقام اعشار آن روی دو رقم تنظیم می‌شود.



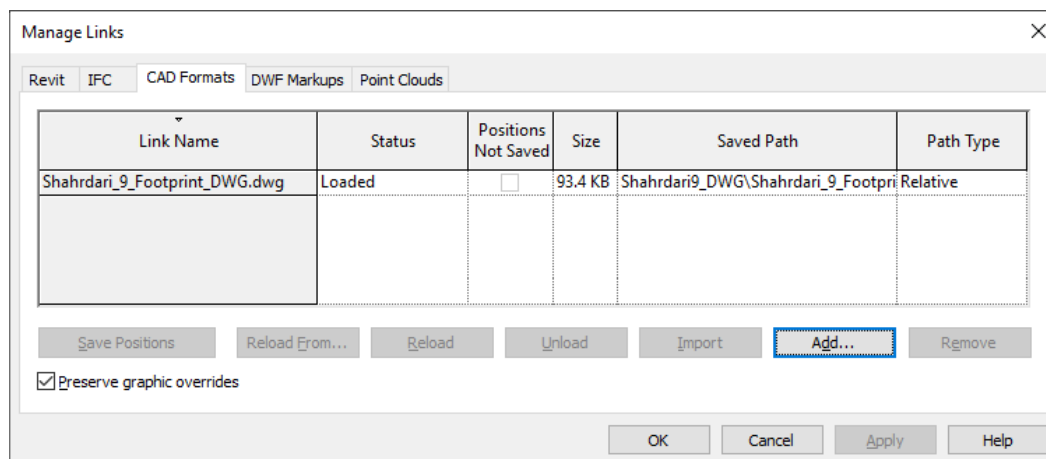
۵- فایل DWG مربوط به پای ساختمان را که پیشتر ایجاد شده است، به فایل RVT لینک می‌کنیم. بدین منظور، با کلیک بر روی تب Insert و سپس کلیک بر روی آیکن Link CAD (کادر قرمز شکل زیر)،



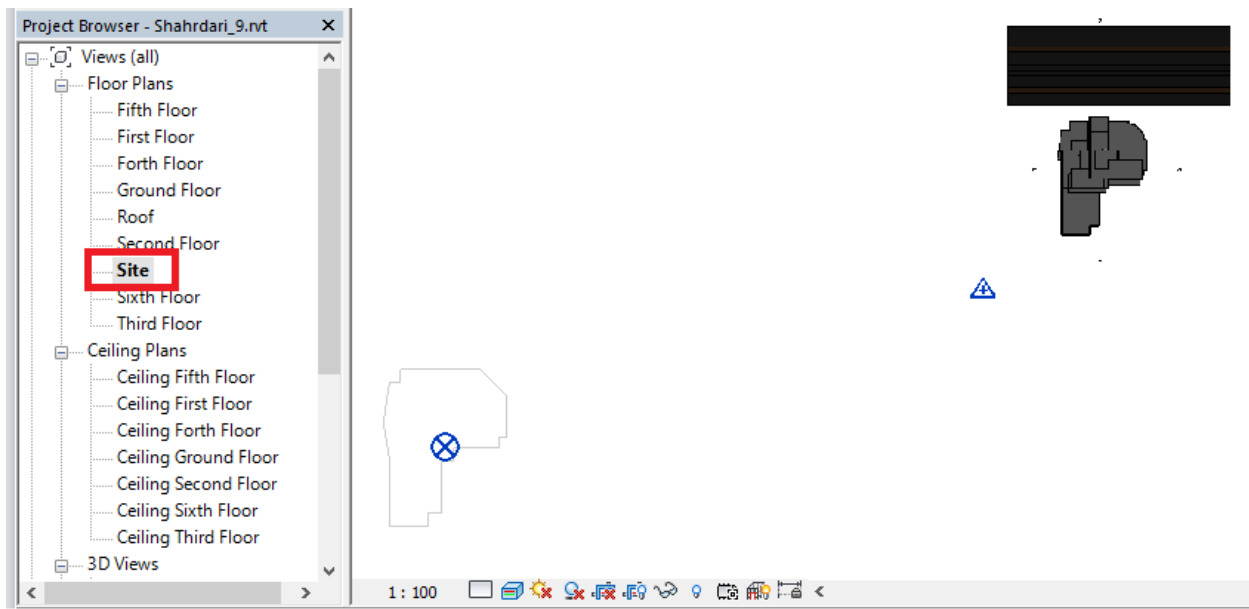
پنجره Link CAD Formats باز می‌شود (شکل زیر). فایل DWG ایجاد شده را انتخاب کرده و بقیه گزینه‌های این پنجره را نیز مطابق شکل زیر، تنظیم می‌کنیم. در نهایت، پنجره زیر، OK می‌شود.



۶- بعد از تایید پنجره فوق، فایل DWG به فایل RVT، لینک می‌شود و این امر، در پنجره Manage Links که قبلاً با آن آشنایی یافته‌ایم، قابل مشاهده است (شکل زیر).

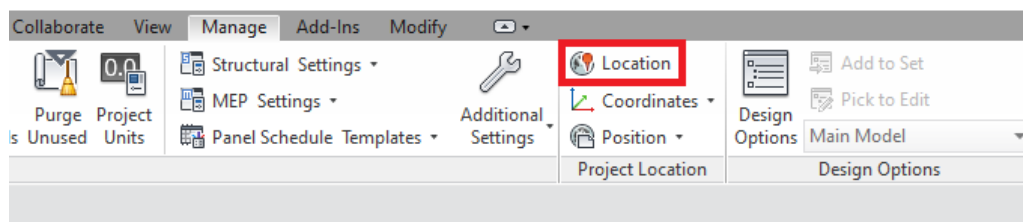


۷- اگر در Revit، روی نمای Site (کادر قرمز شکل زیر)، دابل کلیک کنیم، شکلی مشابه شکل زیر قابل مشاهده است. اگر در نگاه اول، شکل زیر قابل رویت نبود، می‌توان با چند بار زوم کردن، به شکل زیر رسید.

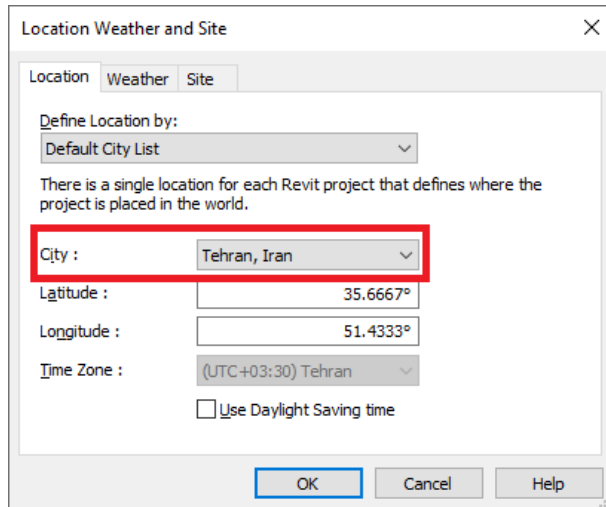


شکل سمت راست و بالا، همان فایل RVT مربوط به ساختمان شهرداری منطقه ۹ تهران به همراه قسمتی از خیابان آزادی است که از نمای بالا نشان داده شده است. شکل پایین و سمت چپ نیز فایل DWG پای ساختمان شهرداری است که به فایل RVT، لینک شده است. هرچند فایل DWG مزبور (پروژه لینک شده)، ذاتاً دارای مختصات UTM صحیح است، اما وقتی به پروژه میزبان (ساختمان ترسیم شده در Revit) لینک می‌شود، مختصات محلی پروژه میزبان را اتخاذ می‌کند که باید در مراحل بعدی، این مورد، معکوس شده و مختصات پروژه میزبان به مختصات پروژه لینک شده، تغییر وضعیت دهد.

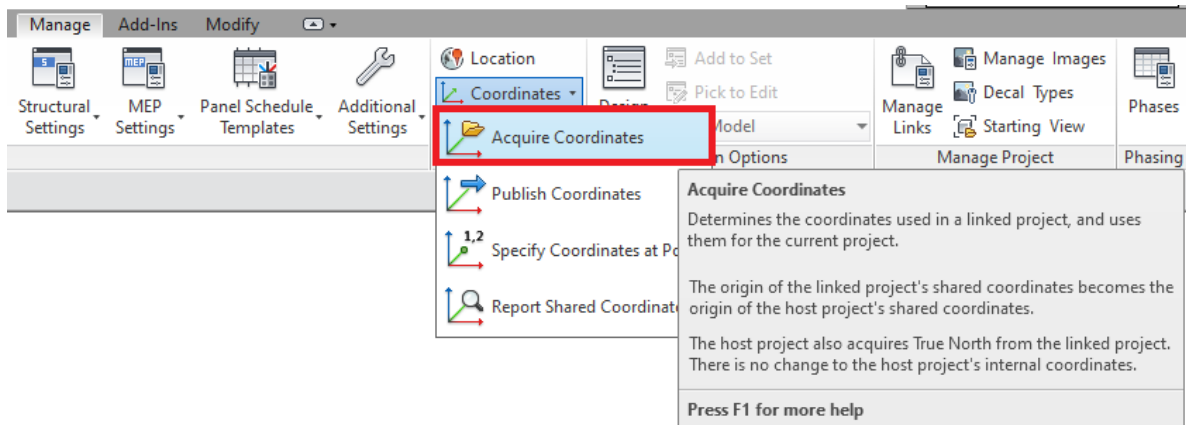
۸- برای اطمینان بیشتر از صحت انجام فرآیند زمین مرجع کردن، محل جغرافیایی در Revit را روی کشور ایران تنظیم می‌کنیم. برای انجام این کار، در تب Manage، روی آیکن Location (کادر قرمز شکل زیر)، کلیک می‌کنیم.



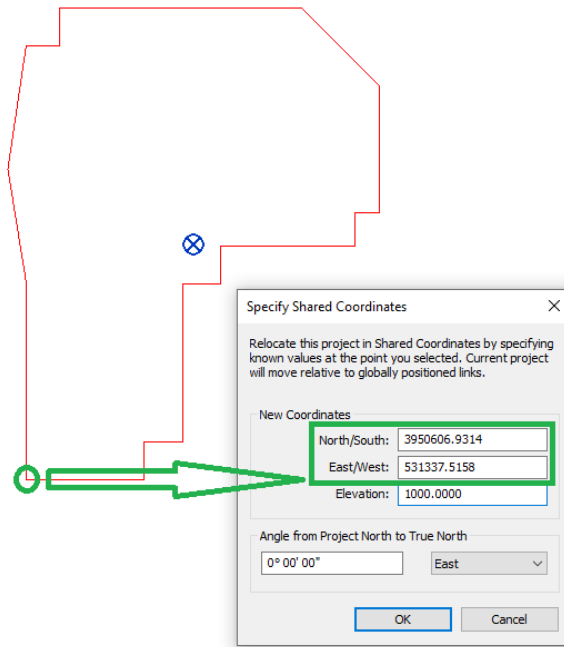
بعد از باز شدن پنجره زیر، کامبوباکس City را روی Tehran, Iran تنظیم می‌کنیم.



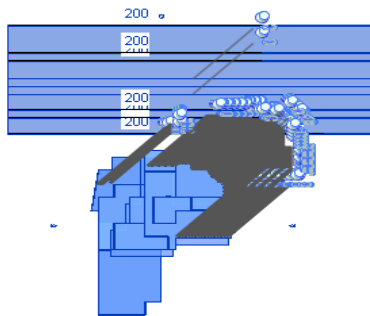
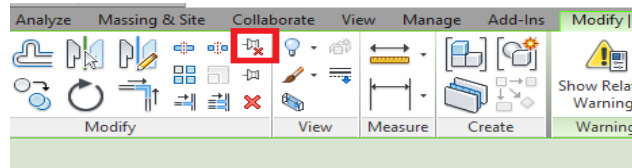
۹- سپس با دستور Acquire Coordinates اقدام به منطبق کردن مبدا مختصات فایل اتوکد لینک شده بر روی مبدا مختصات ساختمان موجود در Revit می‌کنیم. به عبارت دیگر، مختصات پروژه لینک شده (فایل DWG) مربوط به پای ساختمان، مختصات پروژه میزبان (ساختمان موجود در Revit) خواهد شد. لیکن هیچ تغییری در مختصات داخلی پروژه میزبان ایجاد نخواهد شد.



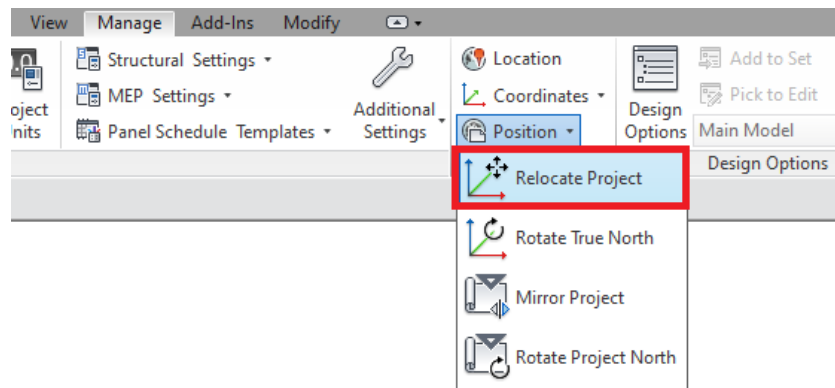
بعد از کلیک روی دستور شکل فوق، روی پای ساختمان (فایل اتوکد لینک شده) که در شکل زیر نشان داده شده، کلیک می‌کنیم. به این ترتیب، مختصات UTM پای ساختمان، ملاک مختصات پروژه میزبان (ساختمان) قرار می‌گیرد و کل پروژه دارای سیستم مختصات UTM خواهد شد. برای کنترل صحت این امر، می‌توان با دستور Specify Coordinates at Point که قبلاً اشاره شد، مختصات گوشه سمت چپ و پایین پای ساختمان را ملاحظه کرد (شکل زیر).



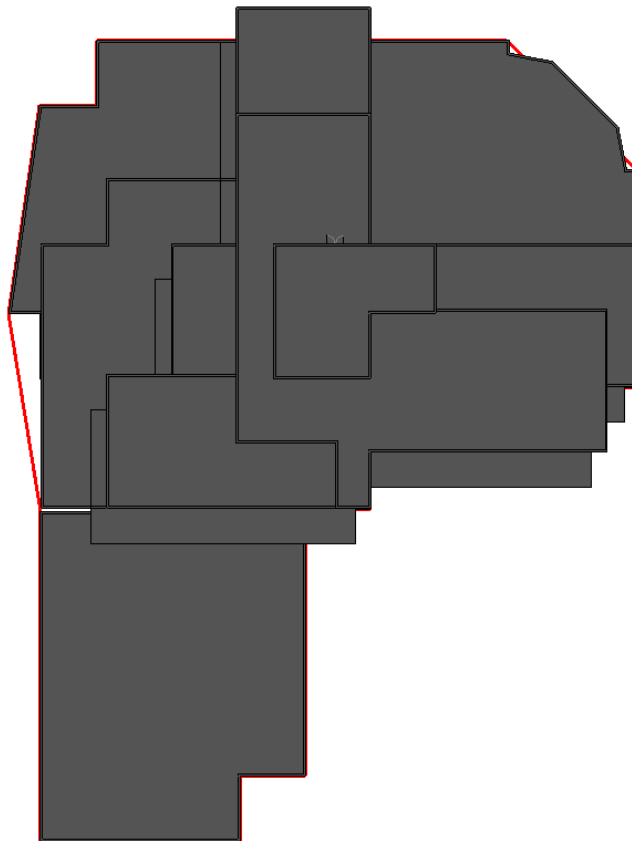
۱۰- درست است که مختصات UTM پای ساختمان، ملاک مختصات کل پروژه قرار گرفته، لیکن هنوز مکان ساختمان با مکان پای ساختمان، از همدیگر فاصله دارند و باید بر روی یکدیگر منطبق شوند. بدین منظور، با دستور Relocate Project (شکل زیر)، ساختمان ترسیم شده در Revit را بر روی پای ساختمان اتوکدی لینک شده، منتقل می‌کنیم تا ساختمان مذکور نیز در جای صحیح خود قرار گیرد. برای Relocate کردن از همان نقطه گوشه سمت چپ و پایین استفاده می‌کنیم. برای این کار، ابتدا کل ساختمان و ملحقات آن را انتخاب می‌کنیم و آن را Unpin می‌کنیم تا قابل منتقل شدن باشد (شکل زیر).



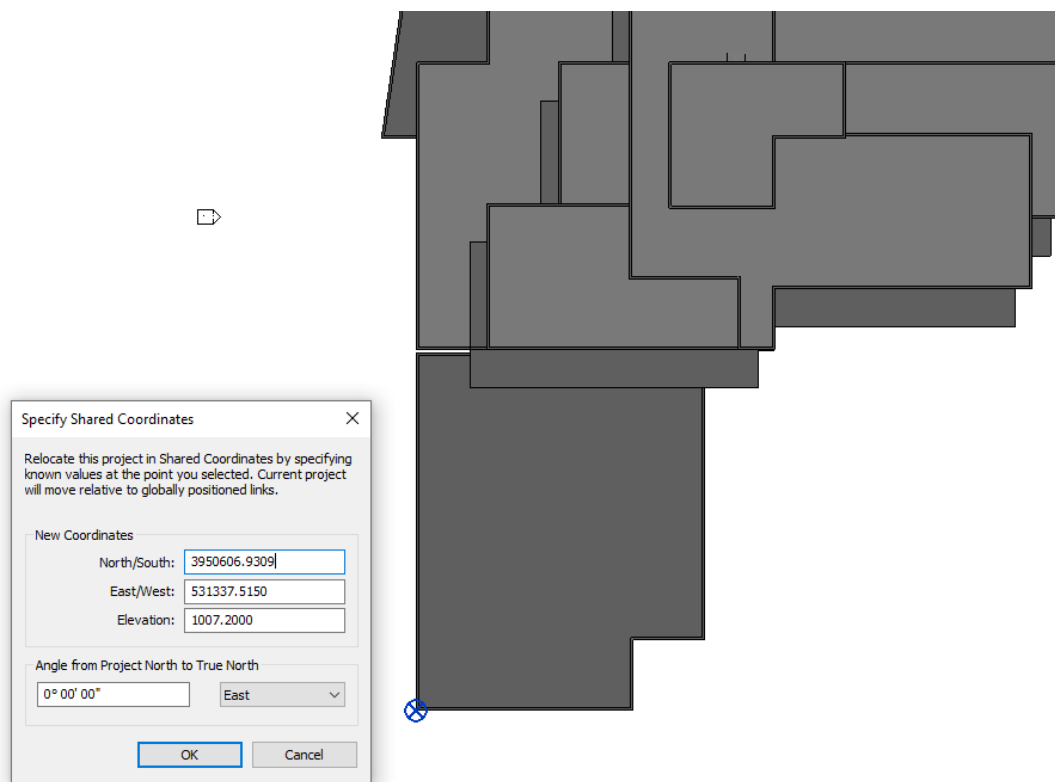
۱۱- سپس با دستور Relocate Project (شکل زیر)، اقدام به انتقال گوشه سمت چپ و پایین ساختمان بر روی گوشه سمت چپ و پایین پای ساختمان می‌کنیم.



نتیجه به صورت زیر خواهد بود. ساختمان ترسیم شده در Revit و پای ساختمان لینک شده، بر هم منطبق شده اند. در اینجا، مراحل زمین مرجع کردن، تکمیل شده است.



۱۲- برای حصول اطمینان، مختصات پلانیمتری یک گوشه ساختمان ترسیم شده در Revit را با دستور Specify Coordinates at Points چک می کنیم تا مختصات صحیح UTM را داشته باشد.



۱۳- از پروژه ترسیم شده در Revit، Export به فرمت IFC می‌گیریم. روش انجام این کار را قبلاً دیده‌ایم.

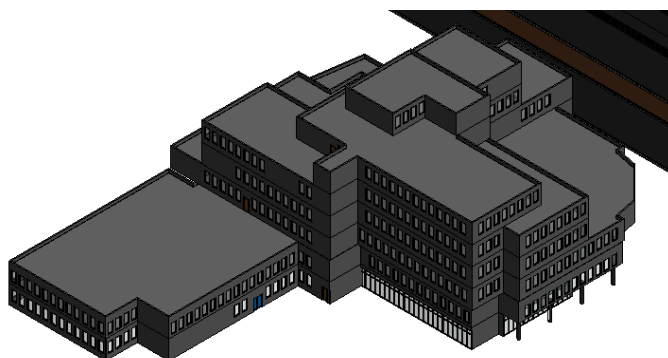
۱۴- برای کنترل صحت نتیجه فرآیند زمین‌مرجع کردن و حصول اطمینان از قرار گرفتن فایل در جای درست خود، فایل IFC را در نرم‌افزار FME به فرمت KML تبدیل می‌کنیم و در نهایت، فایل KML ایجاد شده را در Google Earth باز می‌کنیم و چک می‌کنیم که در در سر جای درست خود بنشیند. نحوه کنترل کردن مزبور نیز قبلاً در بخش ۱-۲ شرح داده شده است و از تکرار آن اجتناب می‌شود.

پس از اطمینان از صحت مکان جغرافیایی فایل IFC نوبت به تبدیل فرمت آن به فرمت استاندارد CityGML و سپس ۳۰۰۰۰۰۰۰ می‌شود. در ادامه، نحوه این تبدیل فرمت‌ها و در نهایت، نمایش فایل ۳۰۰۰۰۰۰۰ در بستر CesiumJS را خواهیم دید.

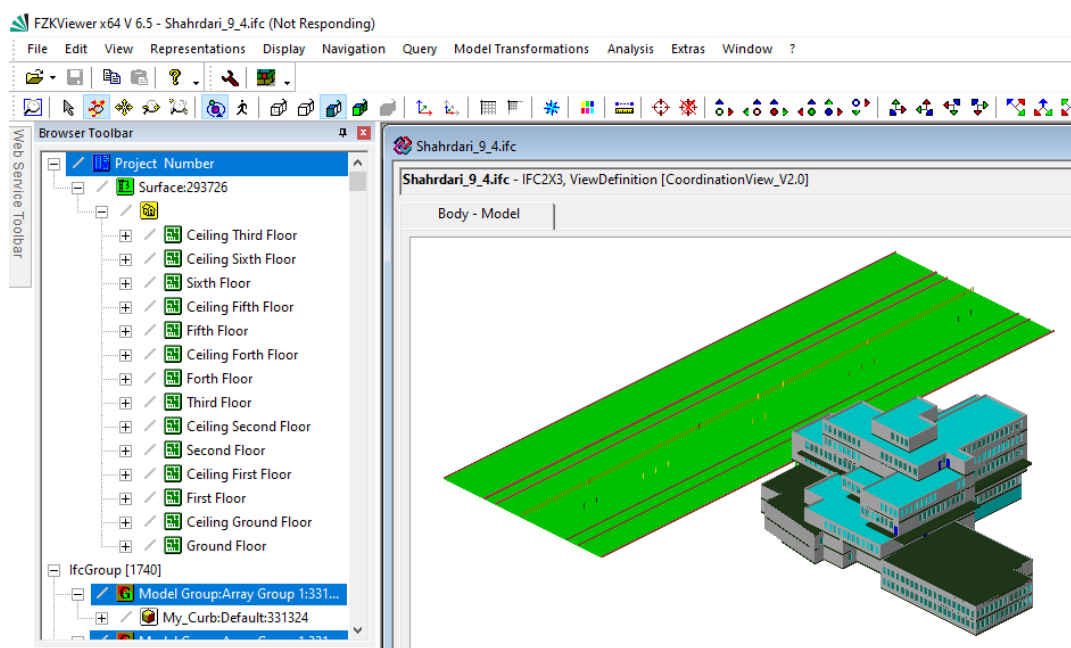
۴- تبدیل فایل Revit به فرمت IFC، CityGML و نمایش آن در CeisumJS

۴-۱- تبدیل فرمت Revit به فرمت IFC

تا اینجای کار، ساختمان شهرداری منطقه ۹ تهران (شکل زیر) در Revit ترسیم و زمین مرجع شده است. یعنی یک فایل RVTی زمین مرجع شده و همچنین فرمت IFCی آن را نیز -که پیشتر در نرم افزار Revit، اکسپورت گرفتیم-، در اختیار داریم.



می توان برای صحت سنجی انجام تبدیل فایل Revit به IFC، اقدام به پیش نمایش فایل IFCی تولید شده کرد. یک راه این است که از مسیر <https://www.iai.kit.edu/english/1302.php> اقدام به دانلود نرم افزار رایگان FZKViewer کرد و در آن، پیش نمایشی از فایل IFCی تولید شده را ملاحظه نمود (شکل زیر). اگر نمایش درستی از فایل IFC مورد نظر ارائه شود، یعنی عملیات تبدیل RVT به IFC به خوبی انجام شده است. نصب این نرم افزار به راحتی انجام می شود.

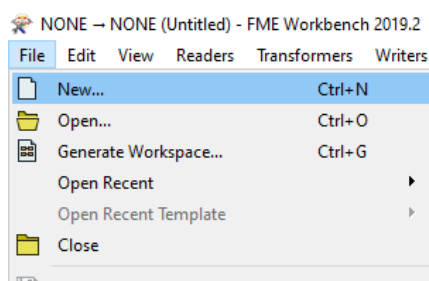


همچنین می‌توان از نرم‌افزارها و وبسایت‌های آنلاین مختلفی برای این کار بهره گرفت که یکی از این وبسایت‌ها، <https://wikiifc.com> است.

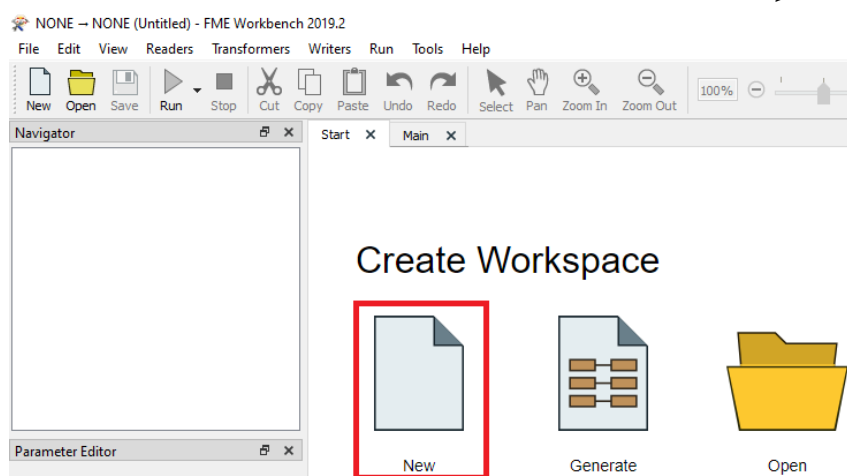
۲-۴- تبدیل فرمت IFC به CityGML

برای تبدیل فایل IFC ایجاد شده در مرحله قبل (با هر کدام از دو روش مذکور) به فرمت CityGML، از نرم‌افزار FME بهره می‌گیریم.

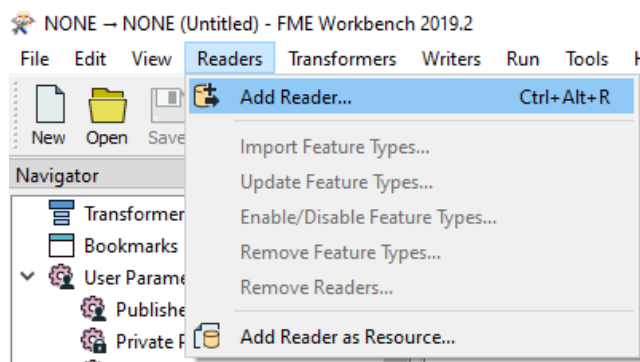
۱- ابتدا باید یک workspace ایجاد کرد. برای ایجاد آن باید از دستور New مطابق شکل زیر استفاده کرد.



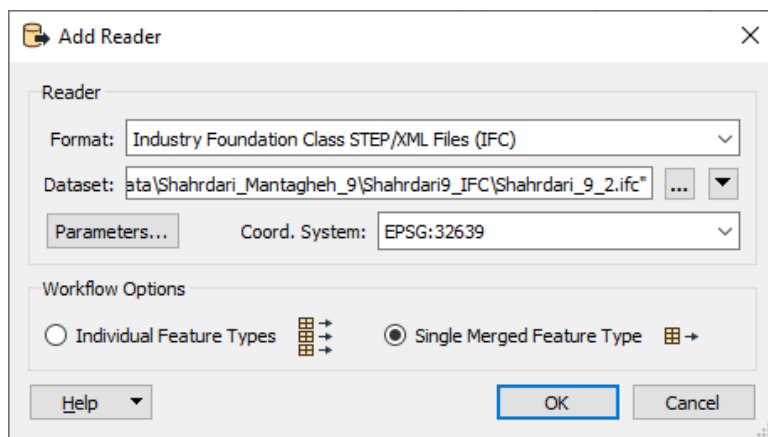
می‌توان دستور بالا را به محض باز کردن FME از طریق کلیک روی آیکن New (شکل زیر) نیز اجرا کرد. با کلیک روی آیکن بالا، یک workspace جدید ایجاد می‌شود (شکل زیر).



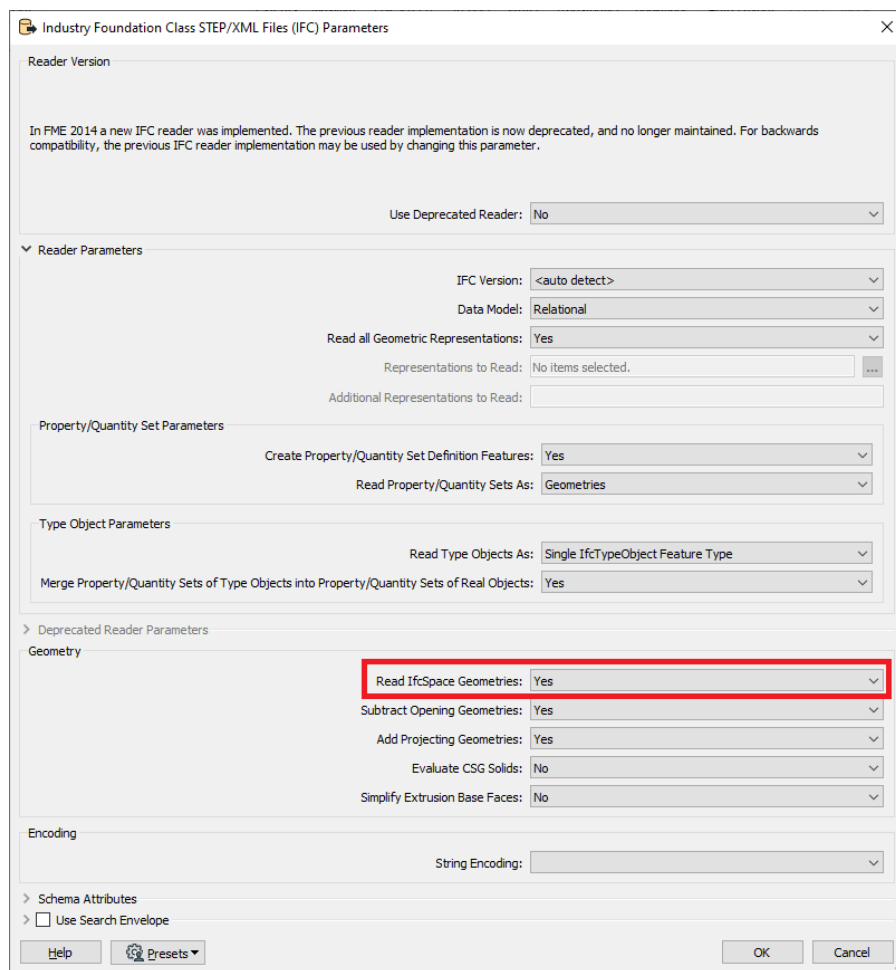
۲- در گام دوم، باید یک Reader را اضافه کرد.



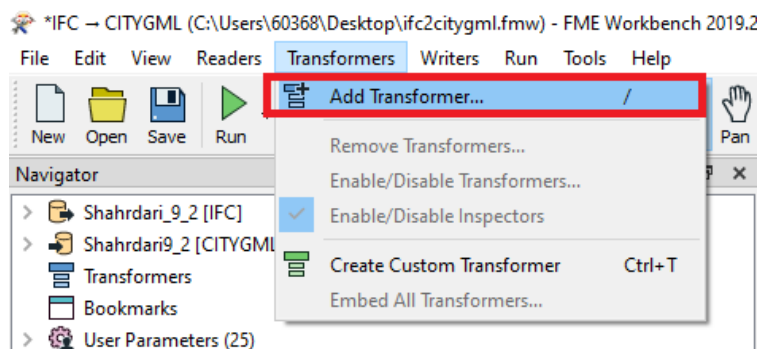
۳- بعد از کلیک روی دستور Add Reader، پنجره شکل زیر باز می‌شود. در پنجره شکل زیر، در قسمت Format، فرمت ورودی داده موردنظر را مشخص می‌کنیم. در اینجا، داده ورودی ما تحت فرمت IFC است، لذا گزینه Industry Foundation Class STEP/XML Files (IFC) در کامبوباکس Format، انتخاب می‌شود. در قسمت Dataset نیز فایل ورودی IFC موردنظر معرفی می‌شود (به عنوان مثال، در اینجا، نام فایل ورودی، Building_3.ifc است).



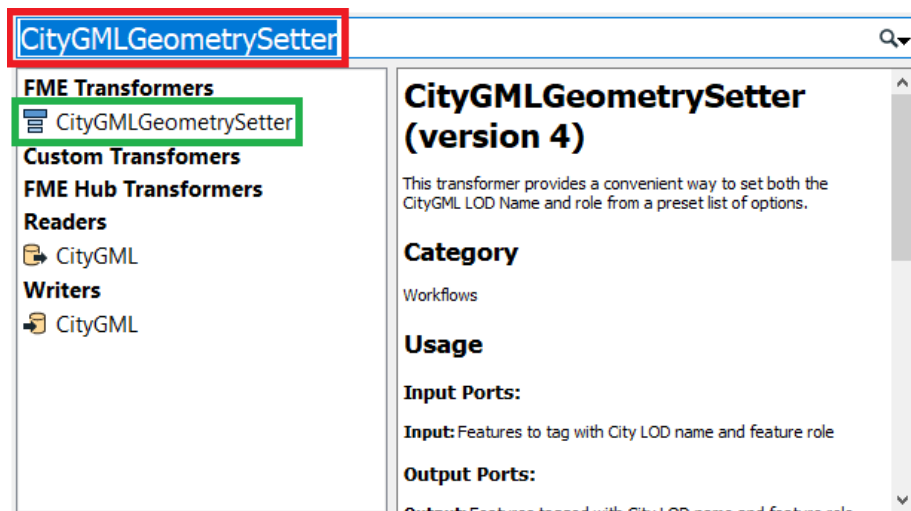
۴- سپس در پنجره فوق بر روی دکمه Parameters کلیک می‌کنیم تا پنجره زیر باز شود و گزینه نشان داده شده در کادر قرمز را روی وضعیت Yes قرار می‌دهیم.



۵- سپس به سراغ دستور Add Transformer می‌رویم تا ترانسفورمری به نام CityGMLGeometrySetter را به مدل اضافه کنیم (کادر قرمز شکل زیر).



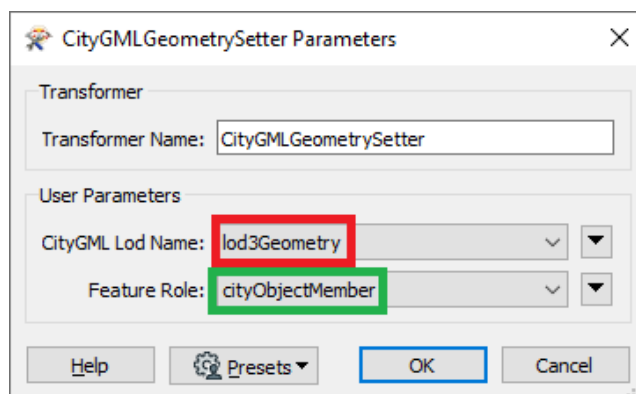
۶- با کلیک روی دستور Add Transformer در شکل فوق، پنجره شکل زیر باز می‌شود که شامل کلیه ترانسفورمرهای نرم‌افزار است. عبارت CityGMLGeometrySetter را در این پنجره تایپ می‌کنیم (کادر قرمز شکل زیر) تا این ترانسفورمور یافت شود. بعد از یافت شدن، روی آن (کادر سبز رنگ شکل زیر) دابل کلیک می‌کنیم تا به مدل اضافه شود.



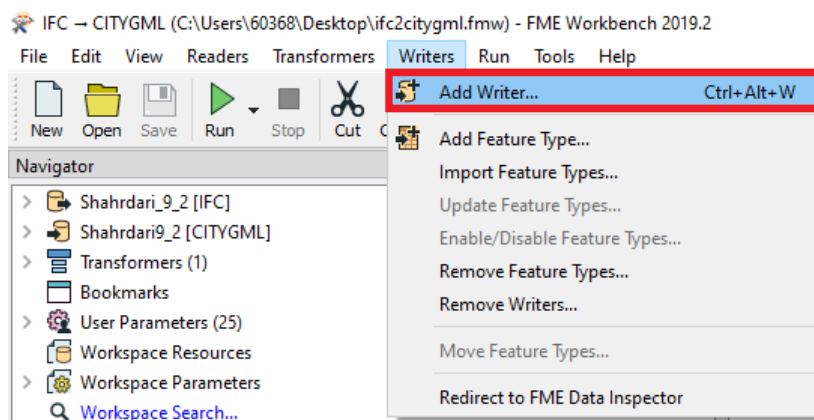
۷- مطابق شکل زیر (کادر قرمز)، ترانسفورمر CityGMLGeometrySetter به مدل اضافه می‌شود.



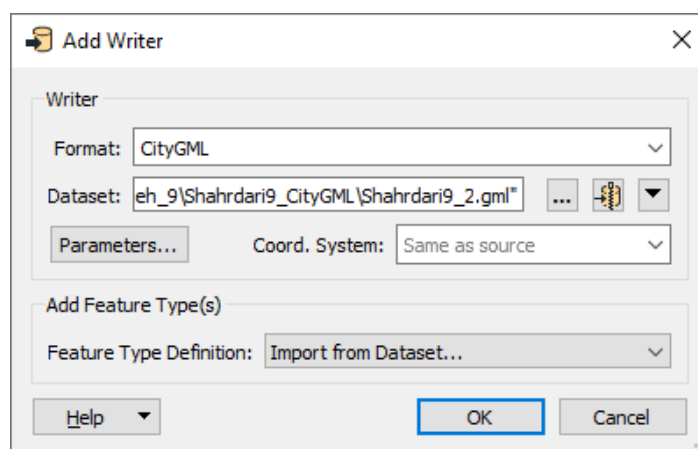
۸- در شکل فوق، Reader را به ترانسفورمر CityGMLGeometrySetter متصل می‌کنیم و روی ترانسفورمر مذکور دابل کلیک می‌کنیم تا پنجره زیر باز شود. قسمت CityGML Lod Name را -که مربوط به سطح جزئیات فایل CityGML است- روی گزینه lod3Geometry تنظیم می‌کنیم (کادر قرمز پنجره زیر). Feature Role را نیز روی گزینه cityObjectMember قرار می‌دهیم و OK می‌کنیم.



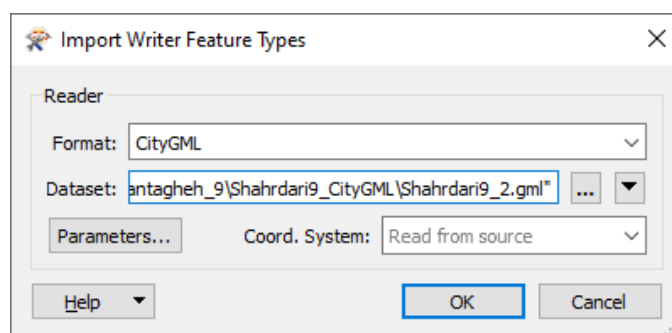
۹- در مرحله آخر نیز باید یک writer به مدل اضافه کنیم. این کار از طریق دستور Add Writer انجام می‌شود (کادر قرمز شکل زیر).



۱۰- با کلیک روی دستور فوق، پنجره شکل زیر باز می‌شود. در قسمت Format، فرمت CityGML را انتخاب می‌کنیم و مسیر و نام فایل CityGML خروجی را نیز در قسمت Dataset مشخص می‌کنیم.

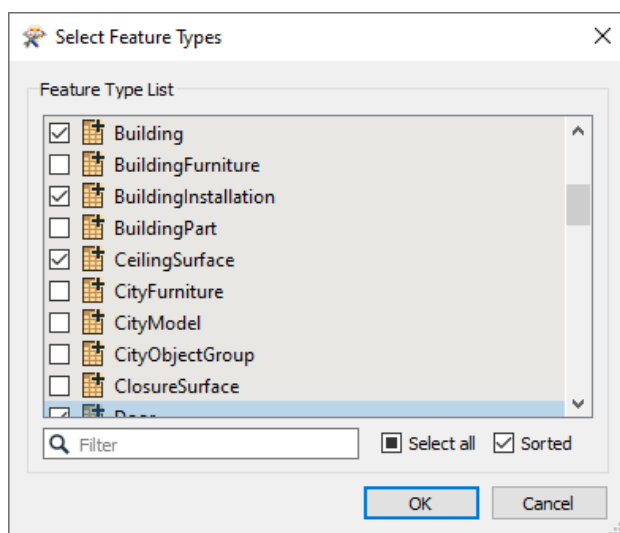


۱۱- با OK کردن پنجره بالا، پنجره شکل زیر باز می‌شود.

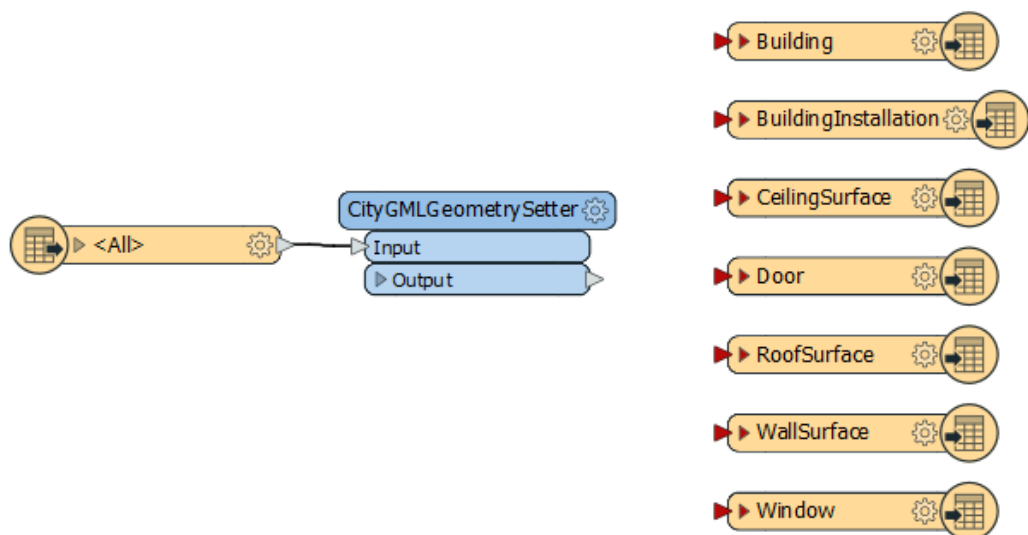


۱۲- با OK کردن پنجره بالا، پنجره شکل زیر باز می‌شود. با شناختی که از ساختمان ترسیم شده داریم، بهتر این است که در پنجره شکل زیر، فقط آن فیچرهایی را تیک بزنیم که در نرم‌افزار Revit برای ساختمان ترسیم شده‌اند (مانند درب، پنجره، پله و ...) و تیک بقیه فیچرها را که در ترسیم ما وجود خارجی ندارند، برداریم تا از حجم بالا و شلوغی فایل CityGML خروجی جلوگیری شود، هرچند می‌توان همه فیچرها را تیک زد، ولی زمان ایجاد فایل CityGML و حجم آن

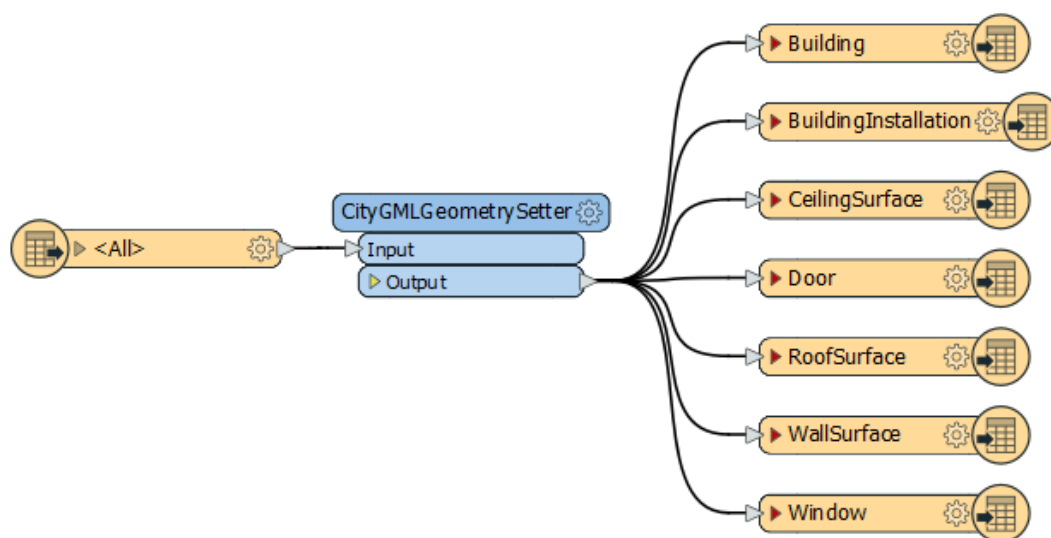
در ساختمان نمونه این مثال (شهرداری منطقه ۹ تهران)، به تعداد شش فیچر مشتمل بر: Building, BuildingInstallation, Door, RoofSurface, WallSurface, Window و CeilingSurface وجود دارند که باید در پنجره زیر، تیک بخورند. این انواع فیچر که در شکل زیر نشان داده شده‌اند، طبقه‌بندی عوارض در استاندارد CityGML است.



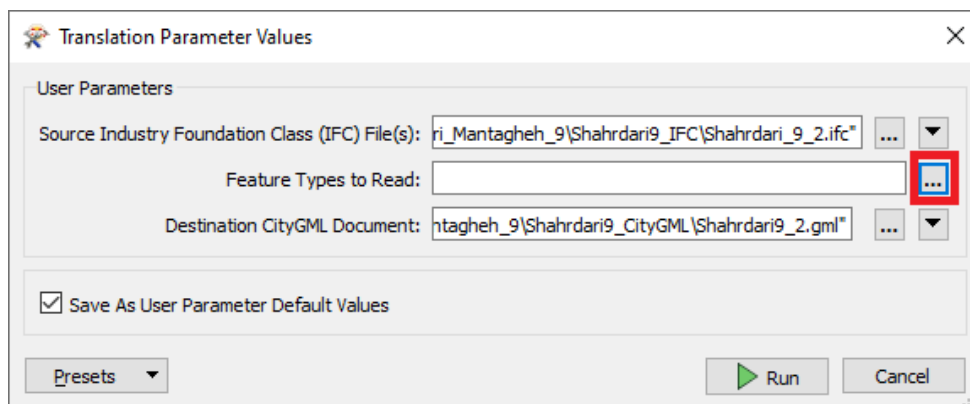
۱۳- با تایید پنجره شکل بالا، این هفت نوع فیچر که قرار است در فایل CityGML خروجی، منعکس شوند، به مدل اضافه می‌شوند و فقط باید تک تک آنها را به ترانسفورمر CityGMLGeometrySetter متصل کنیم. (شکل زیر).



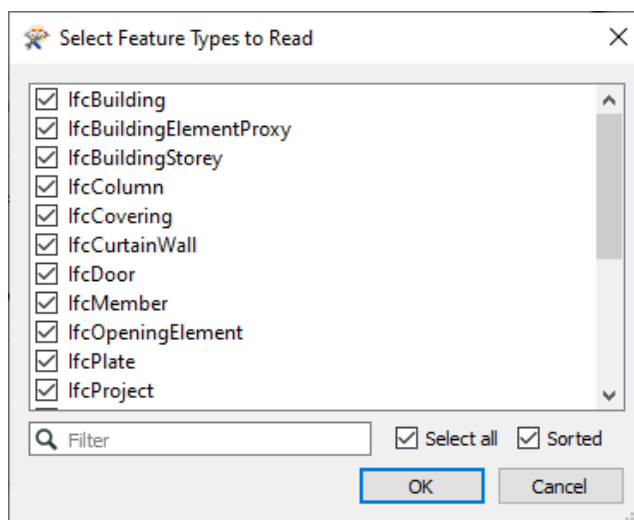
۱۴- با متصل کردن انواع فیچرهای فایل CityGML خروجی به ترانسفورمر CityGMLGeometrySetter، شکل زیر حاصل می‌شود:



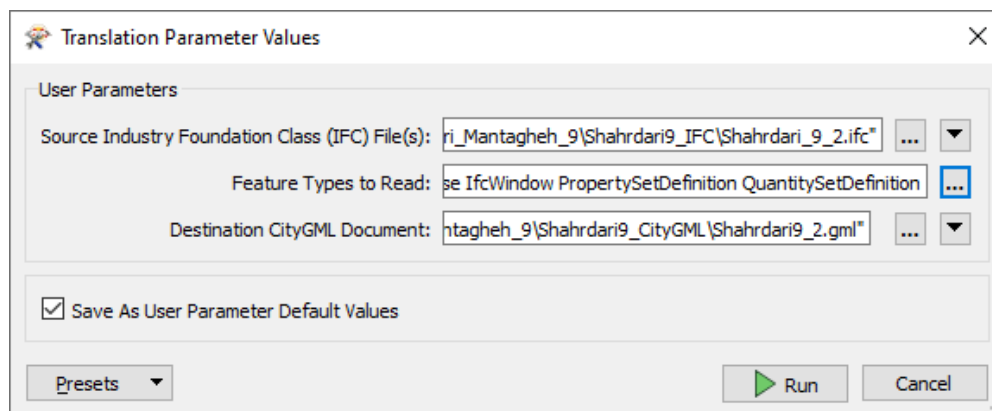
۱۵- با Run کردن مدل، پنجره شکل زیر باز می‌شود. روی دکمه مربوط به انتخاب نوع عوارضی که باید از فایل IFC خوانده شوند تا تبدیل فرمت روی آن‌ها انجام گیرد، کلیک می‌کنیم (کادر قرمز شکل زیر).



۱۶- با کلیک روی کادر قرمز شکل فوق، پنجره زیر باز می‌گردد که حاوی انواع مختلف عوارضی است که در استاندارد IFC تعریف شده‌اند. می‌توان همه نوع عارضه را در پنجره زیر تیک زده و انتخاب کرد.

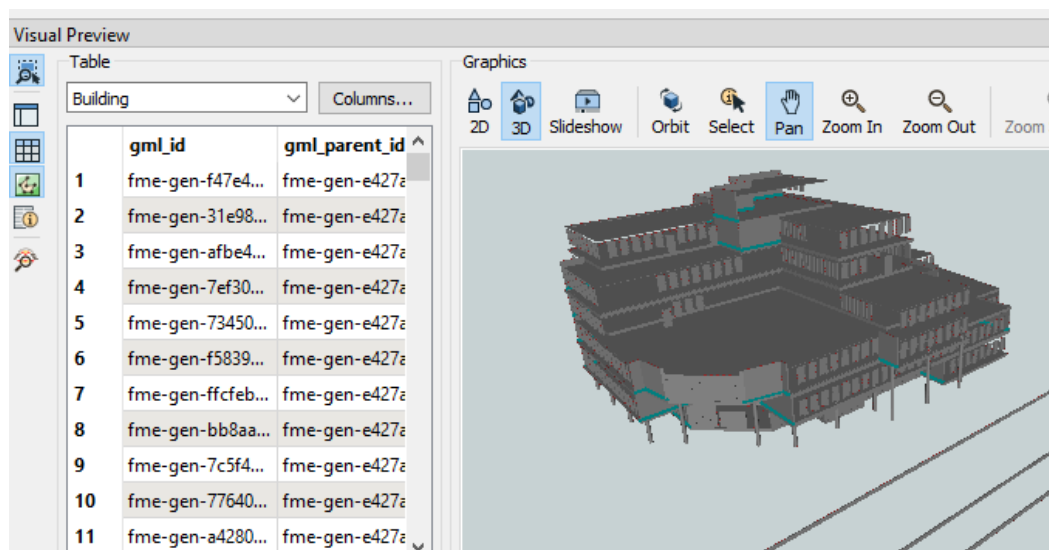


۱۷- با تایید پنجره بالا، مجدداً وارد پنجره شکل زیر می‌شویم:



۱۸- با کلیک روی دکمه Run، پنجره شکل بالا، مدل اجرا می‌شود و فایل CityGML خروجی ساخته می‌گردد.

۱۹- می‌توان با باز کردن پنجره Visual Preview از مسیر View Menu>Windows>Visual Preview و سپس درگ کردن فایل CityGML تولید شده از File Explorer و انداختن آن در پنجره Visual Preview، اقدام به پیش‌نمایش این فایل نمود. چنانچه پیش‌نمایش بدون مشکل نشان داده شود (مانند شکل زیر)، حاکی از صحت فرآیند تبدیل است. اکنون می‌توان به سراغ مرحله بعد رفته و اقدام به تبدیل این فایل به فرمت ۳D نمود.



می‌توانیم در اینجا یکی از نرم‌افزارهای Viewer فایل‌های CityGML، مانند نرم‌افزار رایگان FZKViewer را ذکر کنیم که پیشتر به آن اشاره شد (آدرس وب سایت برای دانلود: <https://www.iai.kit.edu/english/1310.php>). به این ترتیب، می‌توان صحت فایل CityGML تولید شده را بررسی کنیم.

نکته: چنانچه فایل CityGML اصلاً مورد نیاز نباشد، یکی از راه‌های معمول این است که فایل IFC ایجاد شده را ابتدا به OBJ و سپس به ۳D تبدیل کرد.

مرجع:

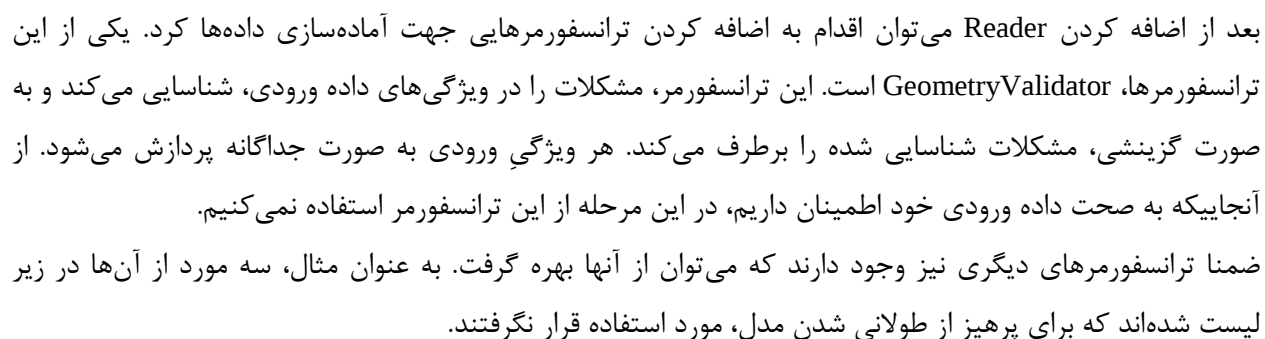
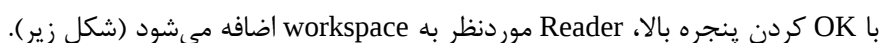
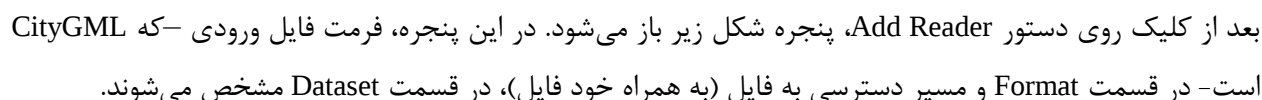
<https://community.safe.com/s/article/bim-to-gis-intermediate-ifc-lod-۳۰۰-to-lod-4-cityg>

<https://community.safe.com/s/article/bim-to-gis-basic-ifc-lod-۱۰۰-to-lod-۲-citygml>

۳-۴- تبدیل فرمت CityGML به ۳D

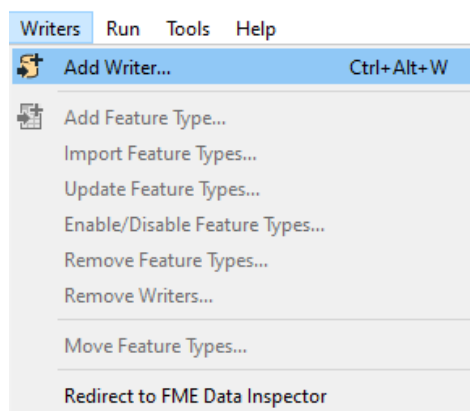
گام آخر در تبدیل فرمت، تبدیل CityGML به ۳D است تا داده موردنظر برای نمایش در Cesium مهیا شود. در اینجا نیز برای تبدیل فرمت CityGML به ۳D از نرم افزار FME استفاده می‌شود. مراحل انجام این کار به شرح ذیل است:

۱- در وهله اول، باید یک Reader را اضافه کرد.

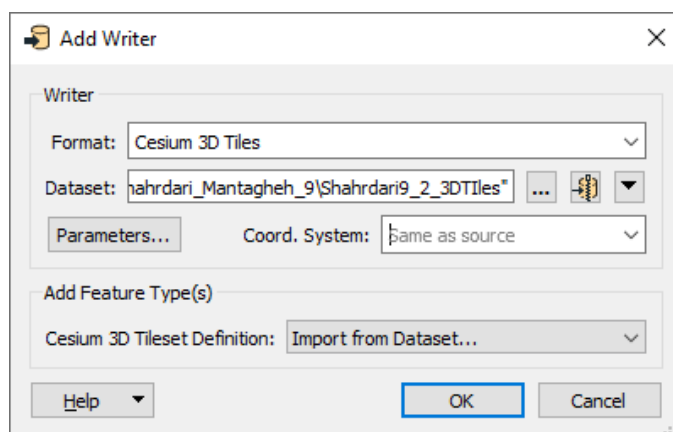


- Extruder و ۳۰۰۰۰۰۰۰۰ برای تبدیل داده های دوبعدی به سه بعدی
- AttributeManager برای مدیریت صفات توصیفی داده ها
- AppearanceSetter برای استایل گذاری بر روی هندسه داده ها

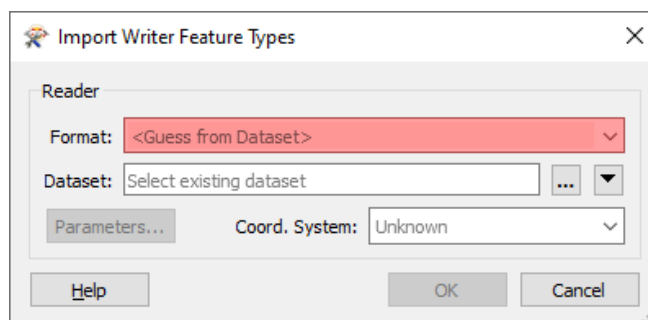
۲- در مرحله دوم، مطابق شکل زیر اقدام به اضافه کردن writer می‌کنیم.



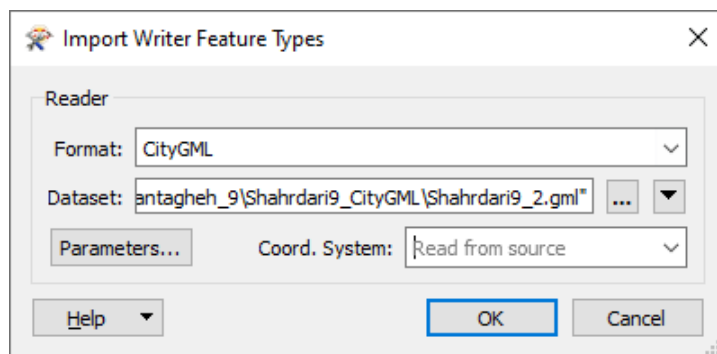
۳- بعد از کلیک روی دستور بالا، پنجره زیر باز می‌شود که در قسمت Format و Dataset آن به ترتیب، نوع فایل خروجی مطلوب (در اینجا Cesium 3D Tiles) و فولدیری که قرار است فایل خروجی در آن قرار گیرد، انتخاب می‌کنیم.



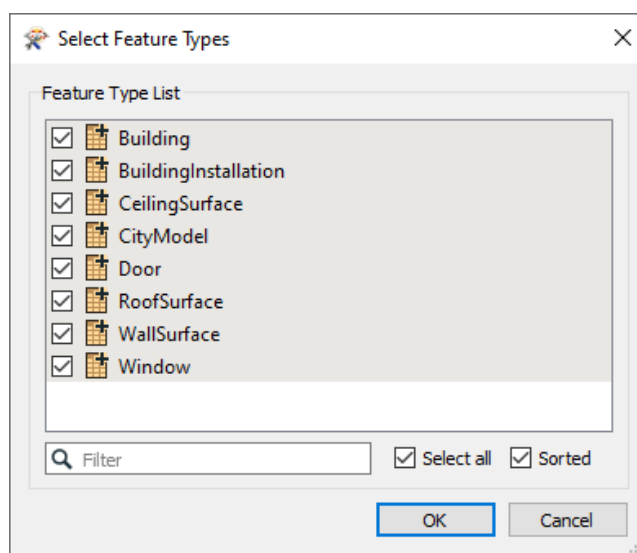
۴- با تایید کردن پنجره فوق، پنجره شکل زیر باز می‌شود:



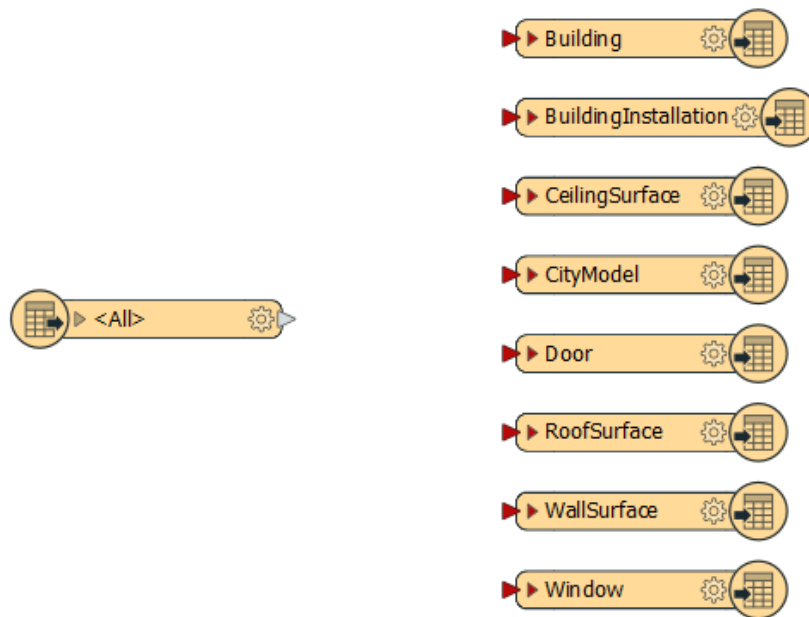
در قسمت Format، فرمت CityGML و در قسمت Dataset، فایل CityGML ای که در بخش قبل تولید شده، انتخاب می‌کنیم (شکل زیر).



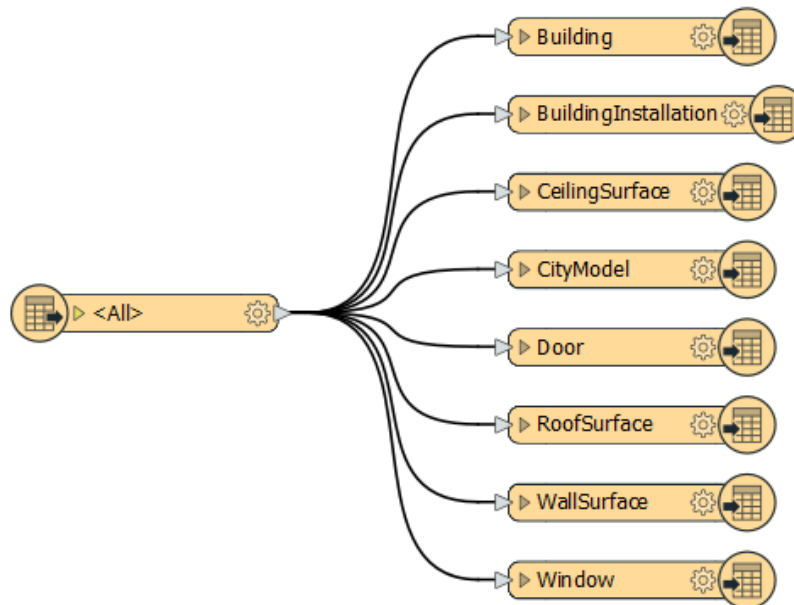
۵- با OK کردن پنجره بالا، پنجره شکل زیر باز می‌شود که شامل کلیه انواع عوارضی است که پیشتر در ترانسفورمر CityGMLGeometrySetter انتخاب شده بودند.



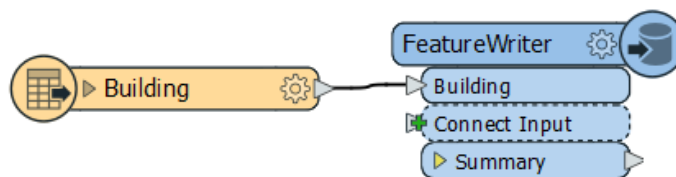
۶- با تایید پنجره فوق، مدل به صورت زیر درمی‌آید:



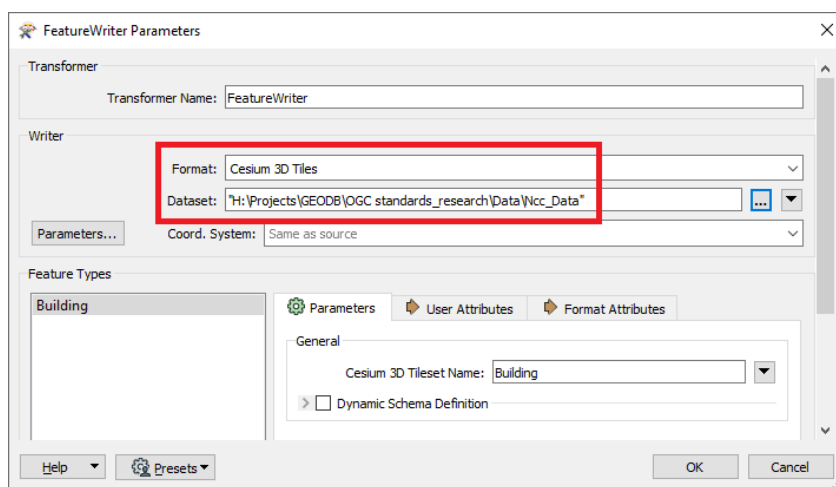
۷- با اتصال Reader سمت چپ به تک تک عوارض سمت راست در شکل فوق، شکل مدل به صورت زیر درمی آید:



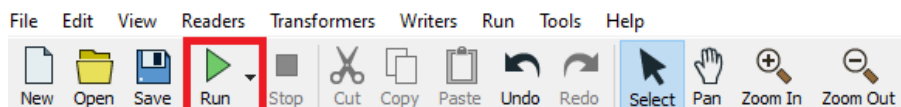
نکته: می توان از ترانسفورمر FeatureWriter به جای استفاده از Writer فوق الذکر جهت تولید فایل ۳D از CityGML استفاده کرد. بدین منظور باید اقدام به اضافه کردن ترانسفورمر FeatureWriter به مدل کرد و داده ورودی (در اینجا CityGML) را به فرمت های موردقبول Ceisum مانند LAS، OBJ و GeoTIFF تبدیل نمود.



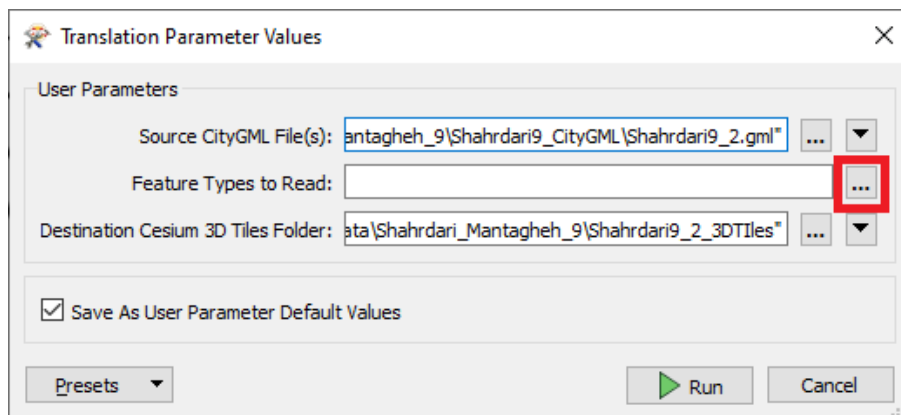
بعد از دابل کلیک روی FeatureWriter در شکل بالا، پنجره زیر باز می‌شود. در قسمت Format، گزینه Cesium 3D Tiles را انتخاب می‌کنیم و در قسمت Dataset، مسیر فولدیری که فایل خروجی در آنجا ایجاد می‌شود، مشخص می‌کنیم و OK می‌کنیم و در نهایت، مدل را اجرا می‌کنیم.



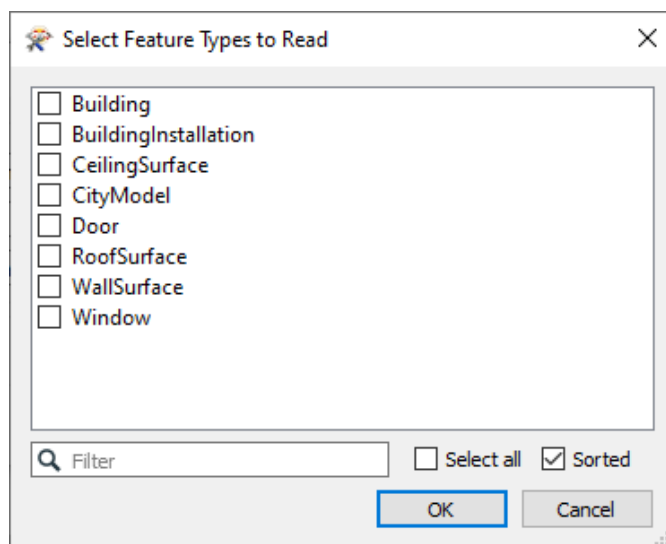
۸- مدل را Run می‌کنیم (کادر قرمز شکل زیر).



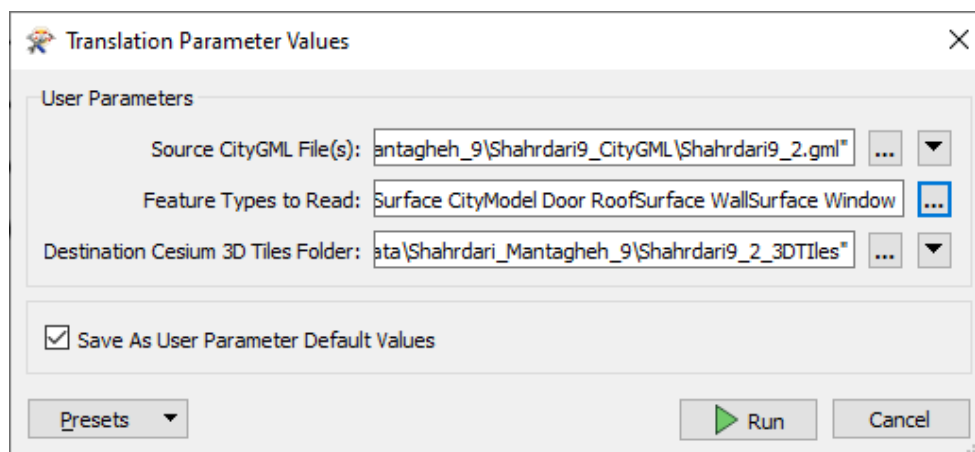
۹- با Run کردن مدل، پنجره شکل زیر باز می‌شود:



۱۰- با کلیک روی دکمه داخل کادر قرمز شکل بالا، پنجره زیر باز می‌شود که شامل همه انواع عوارضی منتخب است.



۱۱- با کلیک چک‌باکس Select all در پنجره شکل بالا، همه انواع عوارض مذکور را انتخاب می‌کنیم و مجدداً به پنجره زیر باز می‌گردیم و روی دکمه Run کلیک می‌کنیم تا مدل اجرا شود.



۱۲- منتظر می‌شویم تا فرآیند اجرا مدل تمام شود. بعد از اتمام مدل، پیغام موفقیت آمیز بودن کار داده می‌شود (کادر قرمز شکل زیر).

```

Translation Log
0 Errors 10 Warnings Information
281 Window 26615
282 =====
283 Total Features Written 212920
284 -----
285 ~~~~~
286 --
287 -- Feature caches have been recorded at every stage of the translation. --
288 -- To inspect the recorded features, --
289 -- click the feature cache icons next to the ports. --
290 -- ~~~~~
291 ~~~~~
292 Translation was SUCCESSFUL with 8 warning(s) (212920 feature(s) output)
293 FME Session Duration: 3 minutes 52.1 seconds. (CPU: 199.8s user, 13.1s system)
294 END - ProcessID: 2836 peak process memory usage: 1093988 kB, current process memory usage: 125676 kB
295 Translation was SUCCESSFUL

```

۱۳- خروجی به صورت زیر خواهد بود:

- Building
- BuildingInstallation
- CeilingSurface
- CityModel
- Door
- RoofSurface
- WallSurface
- Window

هر کدام از فولدرهای فوق شامل یک فولدر به نام data و یک فایل tileset.json می‌باشد. داخل فولدر data، تایل‌های ساخته شده فرمت ۳۰۰۰۰۰۰۰ قرار دارند. حال نوبت به نمایش داده‌های تولید شده ۳۰۰۰۰۰۰۰ Tiles در CesiumJS می‌باشد که با انجام مراحل بعدی میسر است.

۱۴- فولدری که حاوی فولدرهای فوق‌الذکر است را در داخل نرم‌افزار CesiumJS و در مسیر زیر قرار می‌دهیم:
[...\Tomcat 9.0_Tomcat9_9090\webapps\My_CesiumJS\Specs\Data\Cesium3DTiles](#)

۱۵- سپس یک کپی از فایل HelloWorld.html که در مسیر زیر موجود است، می‌گیریم و در همین مسیر، paste می‌کنیم.

[...\Tomcat 9.0_Tomcat9_9090\webapps\My_CesiumJS\Apps](#)

۱۶- نام دلخواهی برای این فایل html کپی شده می‌گذاریم و آن را با نرم‌افزار ویرایشگر متنی باز می‌کنیم. کد زیر را به جای همه کدهای موجود در فایل فوق جایگزین می‌کنیم.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Use correct character set. -->
    <meta charset="utf-8" />

```

```

<!-- Tell IE to use the latest, best version. -->
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<!-- Make the application on mobile take up the full browser screen and disable user scaling.
-->
<meta
  name="viewport"
  content="width=device-width, initial-scale=1, maximum-scale=1, minimum-scale=1, user-
scalable=no"
/>
<title>Hello World!</title>
<script src="../../Build/CesiumUnminified/Cesium.js"></script>
<style>
@import url(../../Build/CesiumUnminified//Widgets/widgets.css);
html,
body,
#cesiumContainer {
  width: 100%;
  height: 100%;
  margin: 0;
  padding: 0;
  overflow: hidden;
}
</style>
</head>
<body>
<div id="cesiumContainer"></div>
<script>
  var viewer = new Cesium.Viewer('cesiumContainer');
  var tileset = viewer.scene.primitives.add(new Cesium.Cesium3DTileset({
    url
:
"../../Specs/Data/Cesium3DTiles/Building_3DTiles_LOD3_7_FeatureTypes/Building/tileset.json", // URL
from `Starting the Server` section.
  }));
  viewer.zoomTo(tileset);
</script>
<script>
  var viewer = new Cesium.Viewer('cesiumContainer');
  var tileset = viewer.scene.primitives.add(new Cesium.Cesium3DTileset({
    url
:
"../../Specs/Data/Cesium3DTiles/Building_3DTiles_LOD3_7_FeatureTypes/BuildingInstallation/tileset.js
on", // URL from `Starting the Server` section.
  }));
  viewer.zoomTo(tileset);
</script>
<script>
  var viewer = new Cesium.Viewer('cesiumContainer');
  var tileset = viewer.scene.primitives.add(new Cesium.Cesium3DTileset({
    url
:
"../../Specs/Data/Cesium3DTiles/Building_3DTiles_LOD3_7_FeatureTypes/CeilingSurface/tileset.json",
// URL from `Starting the Server` section.
  }));
  viewer.zoomTo(tileset);
</script>
<script>
  var viewer = new Cesium.Viewer('cesiumContainer');
  var tileset = viewer.scene.primitives.add(new Cesium.Cesium3DTileset({
    url
:
"../../Specs/Data/Cesium3DTiles/Building_3DTiles_LOD3_7_FeatureTypes/CityModel/tileset.json", // URL
from `Starting the Server` section.
  }));
  viewer.zoomTo(tileset);
</script>

```

```

<script>
var viewer = new Cesium.Viewer('cesiumContainer');
var tileset = viewer.scene.primitives.add(new Cesium.Cesium3DTileset({
    url
    :
"../Specs/Data/Cesium3DTiles/Building_3DTiles_LOD3_7_FeatureTypes/Door/tileset.json", // URL from
`Starting the Server` section.
    }));
viewer.zoomTo(tileset);
</script>

<script>
var viewer = new Cesium.Viewer('cesiumContainer');
var tileset = viewer.scene.primitives.add(new Cesium.Cesium3DTileset({
    url
    :
"../Specs/Data/Cesium3DTiles/Building_3DTiles_LOD3_7_FeatureTypes/RoofSurface/tileset.json", //
URL from `Starting the Server` section.
    }));
viewer.zoomTo(tileset);
</script>

<script>
var viewer = new Cesium.Viewer('cesiumContainer');
var tileset = viewer.scene.primitives.add(new Cesium.Cesium3DTileset({
    url
    :
"../Specs/Data/Cesium3DTiles/Building_3DTiles_LOD3_7_FeatureTypes/WallSurface/tileset.json", //
URL from `Starting the Server` section.
    }));
viewer.zoomTo(tileset);
</script>

<script>
var viewer = new Cesium.Viewer('cesiumContainer');
var tileset = viewer.scene.primitives.add(new Cesium.Cesium3DTileset({
    url
    :
"../Specs/Data/Cesium3DTiles/Building_3DTiles_LOD3_7_FeatureTypes/Window/tileset.json", // URL
from `Starting the Server` section.
    }));
viewer.zoomTo(tileset);
</script>
</body>
</html>

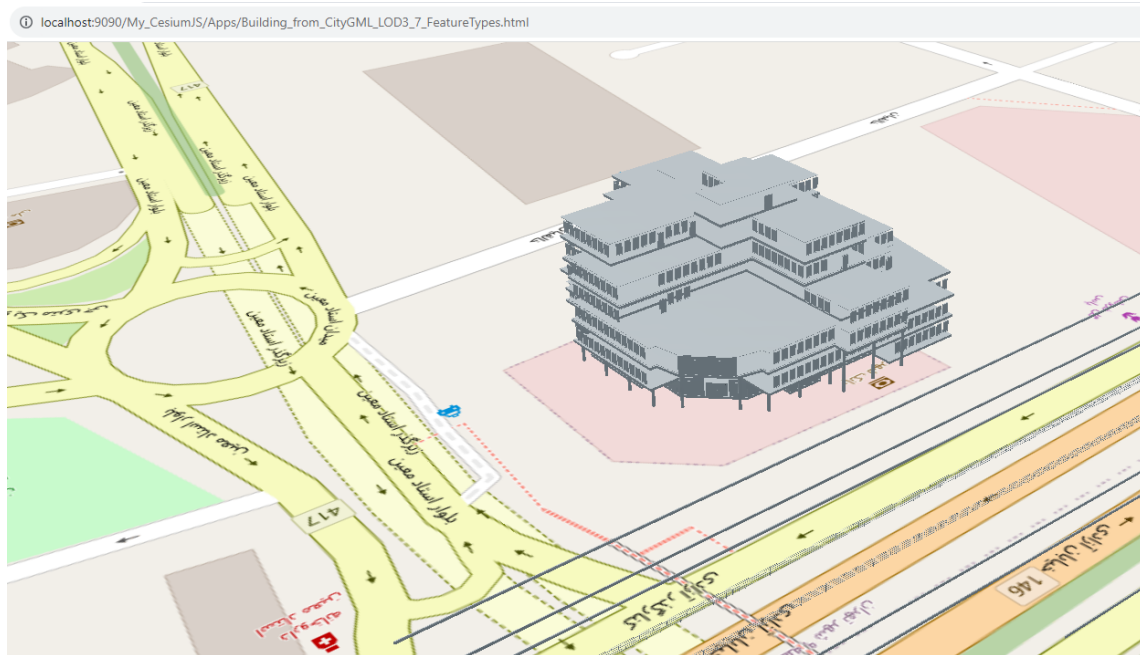
```

کد فوق برگرفته از کد موجود در فایل HelloWorld.html است، فقط کدهای بین دو تگ body در موجود در این فایل، با کدهای رنگی رنگ فوق، جایگزین شده‌اند. هر کدام از پاراگراف‌های رنگی فوق، مربوط به یکی از انواع عوارضی است که در فایل خروجی وجود دارند و در هر کدام از پاراگراف‌های مزبور، مسیر دسترسی به فایل tileset.json یک نوع عارضه درج شده است.

۱۷- حال می‌توان با تایپ مسیر URL ای زیر و فراخوانی فایل html، ۳۰۰۰۰۰های مربوط به ساختمان شهرداری را مشاهده نمود. Building_from_CityGML_LOD3_7_FeatureTypes.html نام فایل html مذکور است که کدهای فوق در آن قرار دارند.

http://localhost:9090/My_CesiumJS/Apps/Building_from_CityGML_LOD3_7_FeatureTypes.html

نتیجه کار در CesiumJS نیز به صورت زیر است:



پس‌زمینه ساختمان شهرداری، نقشه آنلاین OSM است که از طریق شکل زیر (کادر قرمز) در محیط CesiumJS قابل بارگذاری است.



مرجع:

<https://www.linkedin.com/pulse/visualizing-geospatial-data-web-browser-cesium-ion-fme-tiana-warner>
<https://community.safe.com/s/question/0D54Q000080hMeJSAU/citygml-to-cesium-3d-titles>