# gbXML idf comparison tool overview

Summary

This is an extension of work completed in September of 2014 for Autodesk, built on top of a project to compare GBXML and IDF files to one another. The software deserializes the XML into class objects, and also parses the IDF into class objects using Regex functions. All GBXML and IDF files were generated using Autodesk Revit 2016, and downloaded from the Green Building Studio website, prior to running the tests. These files can be found in their respective files found in the deliverables package.

Software takes this collection of objects, and through parsing the objects and analyzing internal content, a series of tests are performed on various properties of the file, related primarily to information surrounding the geometric and thermodynamic definition of Spaces/Zones, where the two terms are used fairly interchangeably, and loosely. The tests follow a basic pattern of Passing or Failing. You should note that each test has an “overall” pass, which means that ***all*** tests on a given object (a Space, e.g.) have passed. To continue the example, a GBXML and IDF file may both have a Space with name “aim0033”. If that is the case, then the names of the two spaces match. This would indicate that one test for these two spaces passes. Next, area, volume, lighting power density, equipment power density are checked via more testing. If all of these tests also prove to show the GBXML and IDF space match, then this space would “Pass Overall”. If just one of the test fails, but the rest pass (for example, if the volumes in the GBXML and IDF did not match), then this space “Fails Overall”.

The results of these tests are stored as a .json file, which is subsequently stored on a local file folder. A separate .json file and local file folder is created for each test run in this series of tests (6 total Revit files). Data from the .json format is parsed using HTML and jQuery, to visualize the data more easily. JQuery is used to sort “PASS” and “FAIL” tests, and also to parse through the results that is converted to html text, and also to some simple visualization tools built with D3.js., as a way to simply summarize data. The hope is, Autodesk engineers may gain more uses for this .json format as it is easily accessible and easily-queried using tools like JQuery or pure javascript, and this makes the testing framework more easily-extensible.

All code libraries (bootstrap, jquery, d3, etc. are included in each of the file result folders. The file folder and subfolder structure should be self-explanatory from the naming convention used.

The HTML pages and JQuery was not tested for cross-browser compatibility. All development of the HTML, CSS, and JQuery was tested using Google Chrome, Version 40.0.2214.94 m, and it is recommended a modern version of Chrome is used to view these results. Instructions in this guide and in the HTML should make using the browser and HTML viewer of the results fairly straightforward. Note that load times can be long due to the size of the .json file.

While GBXML.org and its engineers have taken every opportunity to make the code base and subsequent analysis entirely bug free, there are likely some bugs still in the this software process, as time and resource have allowed only so much development time. That said, should there be any issues with the software, please contact the email address below. We believe the tool still gives fairly sound advice and supports analysis of the zones in an automated way that has proven useful, as a way to catch errors and quantify their impact.

Chien Si Harriman, Developer

Chien.harriman@gmail.com

## Space definition comparisons

Description: The comparison tool grabs all Space elements in the GBXML file, and all of the zones in an IDF file (Sizing:Zone)[[1]](#footnote-1). The purpose of this test is to compare the area, volume, Lighting Power Density definition (labeled LPD for short, includes tests of the input max LPD and the associated schedule assignment), Equipment Power Density (labeled EPD for short, includes tests of input max EPD and associated schedule assignment, people gain, and the number of surfaces associated with each (this surface test is somewhat involved and described in more detail below.

As is common in all of these comparison tests, the GBXML file is considered the primary root of energy model definition, meaning that the IDF file is derived from the root. As such, all tests are based on a simple premise, whereby a function gathers all Spaces declared in GBXML, and, as it loops through each one, tries to find a match for the GBXML Space ID in the collection of Zones found in the IDF.

If a match is found, then a series of tests are conducted on each space definition:

* Is the absolute value of the difference between floor area in GBXML and IDF < 0.001? (regardless of units in GBXML, conversions are in place to handle all units). Otherwise, this test will fail.
* Is the absolute value of the difference between room volume in GBSML and IDF < 0.01? (regardless of units in GBXML, conversions are in place to handle all units). Otherwise, this test will fail.
* Does the max LPD entry match (regardless of units defined in GBXML, conversions are in place) and is the schedule assignment the same string? Otherwise, this test will fail.
* Does the max EPD entry match (regardless of units defined in GBXML, conversions are in place) and is the schedule assignment the same string? Otherwise, this test will fail.
* Does the People Gain entry match (number of people and is the schedule assignment the same string)? [We had to be careful for this test because of the difference in the way GBXML and IDF files differ in description of Total, Sensible, and Latent Load, and this difference was not tested thoroughly. Only the number of people and people fraction schedule was checked.] Otherwise, this test will fail.
* Surfaces Associated with each space. We review the quantity of surfaces, and for each surface:
  + the names of each surface. If names do not match, the test fails
  + the number of vertices of each surface. If number of vertices do not match, the test fails
  + the absolute value of the difference in coordinate values of each surface must be less than 0.001, regardless of units, or else the test fails.
  + the azimuth is a bit complicated, because interior surfaces are represented twice. For this reason, the azimuth and tilt can sometimes be swapped by 180 degrees . The full logic for this test is available on request, but generally, the concept is the that if the absolute value of the difference between azimuth definitions differs by more than 0.01, the test fails
  + the tilt, is similar to the azimuth in this regard
  + planarity of the surface. If the surfaces is deemed non-planar, the test fails \*\*[[2]](#footnote-2)

Because of the large number of tests conducted in these trials, the HTML output document also has some summary statistics included in to show the failure rate of these tests, along with the cause of the failures. Below is a description of each one of the bar or pie charts, along with a rationale for why they were used:

* Spaces Test Pass Rate: gives the rate of “Overall Pass” for the space tests. You can also mouse over the pie chart and a popup will give you the quantity spaces that PASSED or FAILED.
* Invididual Test Failure Count : gives a count of which sub-test failed (the area test, the volume test, the LPD test, etc.. as described in the bullet points on the page above). This helps you understand which subtest caused the overall test to fail.
* Overwhelmingly, the reason the tests fail overall is due to the inability to find surfaces. The “Reasons for Surface Test Failure” bar chart, shows why the test of Surfaces failed:
  + A failed ID match
  + Failure of coordinate tests
  + Azimuths are not within tolerance
  + Tilts are not within tolerance
  + The surfaces is non-planar
  + The “Surface Match Rate” pie chart gives the percentage of surfaces that passed, for all spaces.
  + The “Planar Surface Rate” pie chart gives the percentage of surfaces that are non-planar. This seemed relevant because this indicates typically that Revit is having trouble translating a particular region of a file into planar surfaces.

common failure scenarios

* Surface IDs cannot always be found
* Volume tests occasionally fail, due to inability to meet the precision requirements of the test.

## Opaque surface comparisons

Description: Although surfaces have been vetted in some detail in the space test above (coordinates, tilt, azimuth, etc.), surfaces themselves have more attributes that have been saved for this test. Here, we gather all the surface descriptions in the GBXML and IDF file, and then, looping through each GBXML surface, attempt to find a match of ID in the IDF file. Once a match is identified by ID/Name, we check:

* the type of surface
* the construction name assigned
* whether the wind and sun exposure classification is appropriate for the surface type (E+ surfaces only)
  + if GBXML surface type is SlabOnGrade, SunExposure: NoSun, WindExposure:NoWind
  + if GBXML surface type is ExteriorWall, SunExposure: SunExposed, WindExposure: WindExposed
  + if GBXML surface type is InteriorFloor, SunExposure: NoSun, WindExposure:NoWind
  + if GBXML surface type is Ceiling, SunExposure: NoSun, WindExposure:NoWind
  + if GBXML surface type is InteriorWall, SunExposure: NoSun, WindExposure:NoWind
  + if GBXML surface type is Roof, SunExposure: SunExposed, WindExposure: WindExposed
* whether the adjacency conditions show the same names in each surface
  + For surfaces that are not interior surfaces, find the zone name associated with the EP surface. If it matches the one AdjacentSpaceId in the GBXML Surface, PASS, else FAIL.
  + For surfaces that are interior surfaces, find the zone name associated with the EP surface, then also find the zone name associated with the Copy of that EP Surface (denoted by the word “Copy” appended to the end of the surface name). If both zone names match the two AdjacentSpaceId names associated with the GBXML Surface, PASS, else FAIL.

Because of the large number of surfaces being tested, a summary statistics bar graph/pie chart section was added to the HTML output document. Here is a short description of these visualizations:

* Surfaces Test Pass Rate: gives the rate of Overall Pass or Fail for all surfaces found in the GBXML file, when compared to the IDF
* The Surface Match Failure Distribution bar chart shows the sub test that caused the overall test to fail.
* The Surface Type When Surface Name Match Fails, is there because, overwhelmingly, the reason these tests fail is due to the inability for the test to find a corresponding Surface Name that matches a given GBXML Surface id. This bar chart gives a breakdown of the type of Surface in GBXML that could not be found.

COmmon Failure Scenarios

* A Surface Name in IDF could not be found to match the GBXML Surface ID
  + Overwhelmingly, these surface types are Interior Floor or Interior Wall
* For a surface with GBXML surfaceType=RaisedFloor, we find that the IDF representation of the same surface has OutsideBoundary is set as “Outdoors” which seems inconsistent with the type. (SingleFamilyResidence (20 surface occurrences), Big Box Retail (2 surface occurrences), e.g.)

## Opening comparisons

Description: These tests are specifically for items in the GBXML and IDF that have been labeled as Opening (GBXML) or Window (IDF). After collections of these items have been found, like in other tests, the tests loop through the GBXML objects, and for each ID, tries to find a Name match in the IDF collection. If a name is matched, then these sub tests are subsequently performed:

* Do the parent surface name/ID match? If not, this test FAILS.
* Does the construction string match? If not, this test FAILS.
* Is the absolute value of the difference between the GBXML width and IDF length < 0.01? (conversions are done to ensure any unit of measure in GBXML is allowed)? If not, this test FAILS.
* Is the absolute value of the difference between the GBXML height and IDF height < 0.01? (conversions are done to ensure any unit of measure in GBXML is allowed)? If not, this test FAILS.

There can be many openings to be tested, and so the HTML output report has a summary section to elaborate on why a test may have failed. Here is a description:

* Opening Match Rate gives the percentage of all GBXML openings that have passed all of the sub tests, and for which a name match was found in the IDF.
* The Opening Match Failure Distribution bar chart identifies the Sub Test failure frequency distribution.
* The Failed Opening Name Match Parent Surface Type bar chart is to elaborate on the parent surface type associated with the GBXML opening, when an ID match could not be found. This is here, because it answers the question: “If an opening ID match cannot be found in the IDF file, is the opening an exterior opening or an interior opening?”

One item to be noted. The opening tests currently only test for windows, and not for doors that may be sliding glass doors. We have noted, in the following tests, where this was an issue:

* Krishnan’s Load Calculation : 3 sliding glass doors (on manual inspection, these items came through successfully in the IDF and are correct.

common failure scenarios

* Names are not matched often for openings in the GBXML that are associated with interior surfaces.

## Opaque Construction comparisons

Description: For each opaque surface, the constructions associated with them are gathered for comparison. As in other tests, we first take the GBXML construction definitions, grab its ID, and search for an identical ID in the IDF file. If one is found, a battery of subtests are performed to ensure that the descriptions of the files is identical. If no ID can be found in the IDF to match, the test will fail.

If an ID is found, the test attempts to ascertain how the constructions have been defined. These subtests are:

* The Layer Object is skipped in the GBXML, because of the manner in which it differs from IDF. However, each layer is analyzed and the materials extracted. The order and quantity of resulting materials is compared to the material definitions in the IDF File. If the order and number of materials matches, this test PASSES.
* For each material found:
  + Is the conductivity the same? If so, (regardless of units, which are handled), this test PASSES
  + Is the material thickness the same? If so, (regardless of units, which are handled), this test PASSES
  + Is the specific heat the same? If so, (regardless of units, which are handled), this test PASSES
  + Is the density the same? If so, (regardless of units, which are handled), this test PASSES
  + Is the R-value (if reported) the same? If so, (regardless of units, which are handled), this test PASSES

common failure scenarios

* None

## Opening Construction Comparisons

Description: For each opening, the constructions associated with them are gathered for comparison. As in other tests, we first take the GBXML opening construction definitions, grab its ID, and search for an identical ID in the IDF file. If one is found, a battery of subtests are performed to ensure that the descriptions of the files is identical. If no ID can be found in the IDF to match, the test will fail.

If an ID is found, the test attempts to ascertain how the Opening constructions have been defined. Because of the wide variety of options for an opening construction (simple input of SHGC, VLT, and U-value), more complex matrix input of angular dependence of SHGC, complete glazing and glass layer descriptions, frame descriptions, etc…this test attempts to cover all possibilities. However, this test, as a result, is one of the least tested in this regard. So consider the results of this test to be a WIP.

That said, the general concept for the tests is to first identify how the constructions have been defined. Once a match is found by ID, the tests then see if the constructions have been identically defined. For example, if in the GBXML file the opening description is entered as a single U-value, VLT, and an SHGC matrix of angular dependent values, but the IDF file has the construction defined as glazing and gas layers, the test will fail.

However, if the construction definitions are identical, the tests will resume. For simple entries, where a simple SHGC, VLT, and U-value have been entered, in both files the units are checked for each entry, and the value is checked. The values cannot differ by more than 0.001. The same is true if an angular-dependence matrix is used in lieu of a single value.

If all glazing and gas descriptions have been entered, the tests check each layer, seeing of the physical descriptions of thickness, conductivity, emittance, transmittance, and reflectance match. For gases, thickness, gas type, and units of measure.

common failure scenarios

* The definitions did not always input data in the same way in GBXML and IDF. As in the example above, sometimes the opening construction definitions were entered more simply in GBXML and full glazing and gas descriptions in IDF, although the constructions did share the same name.

## schedule comparisons

Description: In each the GBXML and IDF files, the software grabs all entities from both files. As in other tests, the GBXML file is primary, meaning that GBXML schedule ID values are used to search through the IDF file. Matches are initially based on finding a match of ID’s.

Once a year schedule ID is matched, we then loop through each of the descriptions, looking high level features:

* Year Schedule Type (Fraction, etc.)
* Intervals and the Weekly Schedules in each interval

Once the weekly schedule ids have been gathered, a second set of tests then compare the weekly schedules with the same id in GBXML and IDF, respectively. Here, the tests check, for each day of the week[[3]](#footnote-3):

* The name of the daily schedule associated with that day of the week. If the names are not the same for the day in question, the test fails.

The collection of day schedules are then found as well associated with all weekly schedules previously identified in the Year Schedule intervals. Again, starting by assuming the id’s in the GBXML and IDF files describe the same day schedule, each hour of the day is subsequently checked, along with the Day Schedule Type. If any value for an hour deviate by more than 0.001, the software assumes the day schedules do not match, even if Day Schedule types do match. To summarize the day schedule tests:

* Day Schedules are matched based on ID
* Day Schedule types are compared. If the types differ, the test fails
* Each Hour Value is compared. If the values in GBXML and IDF Day Schedule for the hour in question deviates by more than 0.001, the test fails.

common failure scenarios

* Weekly Schedule tests often fail because the CDD and HDD definitions are null for the IDF, although a value appears in GBXML
* The values for each hour are not always identical. The abs(difference) between them is sometimes greater than the 0.001 threshold

1. This routine is capable of understanding the meaning of zone multipliers, but this feature was not needed for any of these most recent tests. [↑](#footnote-ref-1)
2. We have used an test that appears to, on occasion, find false negatives (surfaces that are planar that are flagged as non-planar). This seems to happen when three consecutive vertices form a straight line and the surface is triangular in shape [↑](#footnote-ref-2)
3. Days labeled Custom1 and Custom2 are not currently GBXML enumerations, and these are ignored and not tested. [↑](#footnote-ref-3)