



VA for 2017!

VisualAnalysis Upgrade News

Our Rename

The schedule for VisualAnalysis slipped a couple months, so the target release is now landing solidly into 2017. We have been calling the next version 'VisualAnalysis 13' (logically, with some squeamishness). It is really an arbitrary number, as we used numerous major upgrades with versions like 2.5, 3.11, and 5.5. Whatever. The new name is now:

VA 17

As in 2017. As in not 2016 like we had hoped. But with the amount of effort poured into this version, it is perhaps appropriate to do the: *"One, Two, Skip-A-Few"* routine.

Our Time Frame

Development is nearing the day where all features and functionality are complete. And WE are excited to get it into your hands. Our intention is to do this as soon as possible so you have an opportunity to **direct our finishing touches** and tell us where it still needs fine tuning. The first goal, though, is to avoid wasting your precious time with an incomplete product, or something unusable! Here is the tentative schedule of events.

VisualAnalysis 17.0 Schedule

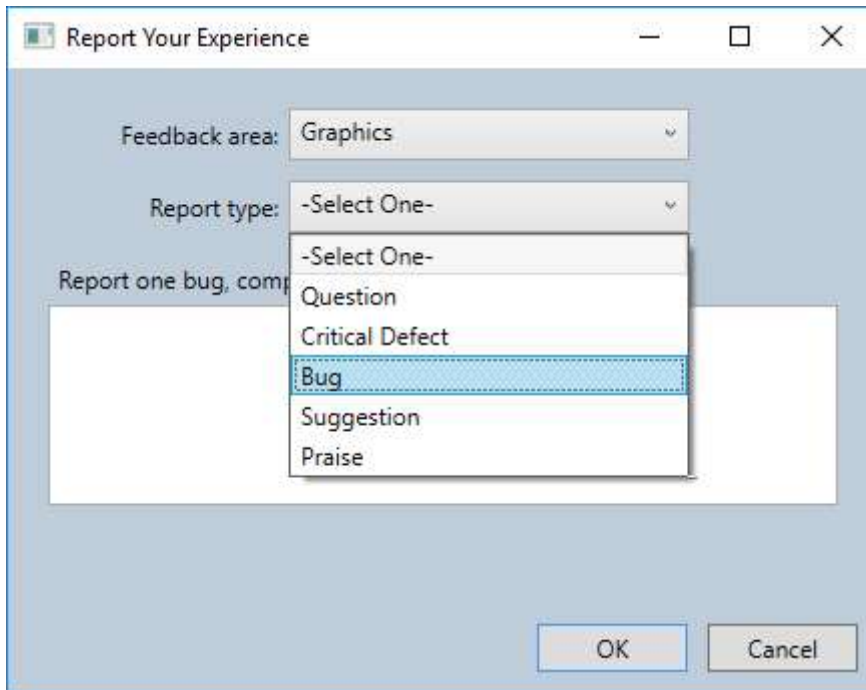
December	January	February	March
Alpha Test (10 users), Beta Test #1 (Soon)	Beta Test #2, Customer Training (Usability Lab)	Release Candidate	Official Launch

Beta Tests

We will be strongly encouraging all VA customers to participate in the beta testing process. We will send an announcement when its ready. You will need an active license for VisualAnalysis 12.0 to participate. The early test is the best opportunity for you to **have a huge impact** on the quality of the program for your very specific situation, projects, and features.

Not much time is required to put VA through its paces and give us your feedback. We know your time is valuable and you have REAL projects to work on--don't use the beta-test for those!

Your single hour (early in testing) might save many of hours in frustration, after the product is released. **The beta install will be fast**, painless. Your current VA will run side-by-side with the test VA. Simply keep your test projects separated from production projects because the file format is changing.



Feedback will also be made **trivially easy** with a built-in submission form. Just a few clicks to tell us what's wrong or right about the product!

Usability Lab - Customer Training

In January, IES is hosting a free customer training class in Bozeman. The primary purpose is to **help customers** understand and master VisualAnalysis. The secondary purpose is to observe customers one-on-one as they interact with the new version. We hope to learn as much as the attendees!

Your Claim to Fame

Our goal is to make VA 17 the best structural software on the planet--but not for everyone: just for you! We are listening to your concerns, needs, and ideas. This new version tries to incorporate the best of VA 12.0 with the ideas and goals customers like you have expressed. See the [Coming Soon](#) teaser-list in this issue.

[\[back to top\]](#)

Hockey in Bozeman

By Garrett Drake, PE Developer in R&D

Ice Time

Monday, June 20th was my favorite day of 2016. Why? The Summer solstice, when the longest day of the year begs to be spent outside, enjoying the Big Sky country we are lucky enough to call home. My second favorite day of the year? Monday September 26th, the first day of ice at the hockey rink and the beginning of another season playing

in the local adult hockey league.

```

00: @SuppressWarnings("PMD")
01: {
02:     public partial class DesignSelectionDialog : Window, IObjectFactoryDialog
03:     {
04:         #region public Interface
05:         [Browsable]
06:         public static GroupGetInitialDialogResult OptimizeSelected()
07:         {
08:             var selected = *InitialSelected;
09:             if (selected.Count != 1)
10:             {
11:                 MessageBox.Show("Please select a single design group for optimization.");
12:                 return null;
13:             }
14:             return Optimize(selected[0]);
15:         }
16:
17:         [Browsable]
18:         public static GroupGetInitialDialogResult Optimize(DesignParam group)
19:         {
20:             var search = GetInitialParameters(group);
21:             if (search.Item1 != null)
22:             {
23:                 MessageBox.Show("Search failed.");
24:                 return null;
25:             }
26:             return Optimize(group, search.Item2);
27:         }
28:
29:         [Browsable]
30:         private static GroupGetInitialDialogResult Optimize(DesignParam group, IObjectFactoryParameters search)
31:         {
32:             var dialog = new DesignSelectionDialog(group, search);
33:             if (Dialog.ShowDialog() == true & dialog.Result != null & dialog.Result.SearchParameters.GetResult() != null)
34:             {
35:                 return new GroupGetInitialDialogResult(dialog.Result, dialog.Result.SearchParameters.GetResult(), group);
36:             }
37:             return null;
38:         }
39:
40:         #region Private
41:         private DesignParam _designParam;
42:         private IObjectFactoryParameters _searchParam;
43:         private IObjectFactoryParameters _selectedParam;
44:
45:         private DesignSelectionDialog(DesignParam group, IObjectFactoryParameters search)
46:         {
47:             InitializeComponent();
48:             _designParam = group;
49:             _searchParam = search;
50:             _selectedParam = null;
51:         }
52:
53:         private void InitializeComponent()
54:         {
55:             // ...
56:         }
57:
58:         private void btnOptimize_Click(object sender, EventArgs e)
59:         {
60:             // ...
61:         }
62:
63:         private void btnCancel_Click(object sender, EventArgs e)
64:         {
65:             // ...
66:         }
67:
68:         private void btnOK_Click(object sender, EventArgs e)
69:         {
70:             // ...
71:         }
72:
73:         private void btnSearch_Click(object sender, EventArgs e)
74:         {
75:             // ...
76:         }
77:
78:         private void btnReset_Click(object sender, EventArgs e)
79:         {
80:             // ...
81:         }
82:
83:         private void btnHelp_Click(object sender, EventArgs e)
84:         {
85:             // ...
86:         }
87:
88:         private void btnClose_Click(object sender, EventArgs e)
89:         {
90:             // ...
91:         }
92:
93:         private void btnApply_Click(object sender, EventArgs e)
94:         {
95:             // ...
96:         }
97:
98:         private void btnDefault_Click(object sender, EventArgs e)
99:         {
100:             // ...
101:         }
102:
103:         private void btnLoad_Click(object sender, EventArgs e)
104:         {
105:             // ...
106:         }
107:
108:         private void btnSave_Click(object sender, EventArgs e)
109:         {
110:             // ...
111:         }
112:
113:         private void btnPrint_Click(object sender, EventArgs e)
114:         {
115:             // ...
116:         }
117:
118:         private void btnExit_Click(object sender, EventArgs e)
119:         {
120:             // ...
121:         }
122:
123:         private void btnHelp_Click(object sender, EventArgs e)
124:         {
125:             // ...
126:         }
127:
128:         private void btnClose_Click(object sender, EventArgs e)
129:         {
130:             // ...
131:         }
132:
133:         private void btnApply_Click(object sender, EventArgs e)
134:         {
135:             // ...
136:         }
137:
138:         private void btnDefault_Click(object sender, EventArgs e)
139:         {
140:             // ...
141:         }
142:
143:         private void btnLoad_Click(object sender, EventArgs e)
144:         {
145:             // ...
146:         }
147:
148:         private void btnSave_Click(object sender, EventArgs e)
149:         {
150:             // ...
151:         }
152:
153:         private void btnPrint_Click(object sender, EventArgs e)
154:         {
155:             // ...
156:         }
157:
158:         private void btnExit_Click(object sender, EventArgs e)
159:         {
160:             // ...
161:         }
162:
163:         private void btnHelp_Click(object sender, EventArgs e)
164:         {
165:             // ...
166:         }
167:
168:         private void btnClose_Click(object sender, EventArgs e)
169:         {
170:             // ...
171:         }
172:
173:         private void btnApply_Click(object sender, EventArgs e)
174:         {
175:             // ...
176:         }
177:
178:         private void btnDefault_Click(object sender, EventArgs e)
179:         {
180:             // ...
181:         }
182:
183:         private void btnLoad_Click(object sender, EventArgs e)
184:         {
185:             // ...
186:         }
187:
188:         private void btnSave_Click(object sender, EventArgs e)
189:         {
190:             // ...
191:         }
192:
193:         private void btnPrint_Click(object sender, EventArgs e)
194:         {
195:             // ...
196:         }
197:
198:         private void btnExit_Click(object sender, EventArgs e)
199:         {
200:             // ...
201:         }
202:
203:         private void btnHelp_Click(object sender, EventArgs e)
204:         {
205:             // ...
206:         }
207:
208:         private void btnClose_Click(object sender, EventArgs e)
209:         {
210:             // ...
211:         }
212:
213:         private void btnApply_Click(object sender, EventArgs e)
214:         {
215:             // ...
216:         }
217:
218:         private void btnDefault_Click(object sender, EventArgs e)
219:         {
220:             // ...
221:         }
222:
223:         private void btnLoad_Click(object sender, EventArgs e)
224:         {
225:             // ...
226:         }
227:
228:         private void btnSave_Click(object sender, EventArgs e)
229:         {
230:             // ...
231:         }
232:
233:         private void btnPrint_Click(object sender, EventArgs e)
234:         {
235:             // ...
236:         }
237:
238:         private void btnExit_Click(object sender, EventArgs e)
239:         {
240:             // ...
241:         }
242:
243:         private void btnHelp_Click(object sender, EventArgs e)
244:         {
245:             // ...
246:         }
247:
248:         private void btnClose_Click(object sender, EventArgs e)
249:         {
250:             // ...
251:         }
252:
253:         private void btnApply_Click(object sender, EventArgs e)
254:         {
255:             // ...
256:         }
257:
258:         private void btnDefault_Click(object sender, EventArgs e)
259:         {
260:             // ...
261:         }
262:
263:         private void btnLoad_Click(object sender, EventArgs e)
264:         {
265:             // ...
266:         }
267:
268:         private void btnSave_Click(object sender, EventArgs e)
269:         {
270:             // ...
271:         }
272:
273:         private void btnPrint_Click(object sender, EventArgs e)
274:         {
275:             // ...
276:         }
277:
278:         private void btnExit_Click(object sender, EventArgs e)
279:         {
280:             // ...
281:         }
282:
283:         private void btnHelp_Click(object sender, EventArgs e)
284:         {
285:             // ...
286:         }
287:
288:         private void btnClose_Click(object sender, EventArgs e)
289:         {
290:             // ...
291:         }
292:
293:         private void btnApply_Click(object sender, EventArgs e)
294:         {
295:             // ...
296:         }
297:
298:         private void btnDefault_Click(object sender, EventArgs e)
299:         {
300:             // ...
301:         }
302:
303:         private void btnLoad_Click(object sender, EventArgs e)
304:         {
305:             // ...
306:         }
307:
308:         private void btnSave_Click(object sender, EventArgs e)
309:         {
310:             // ...
311:         }
312:
313:         private void btnPrint_Click(object sender, EventArgs e)
314:         {
315:             // ...
316:         }
317:
318:         private void btnExit_Click(object sender, EventArgs e)
319:         {
320:             // ...
321:         }
322:
323:         private void btnHelp_Click(object sender, EventArgs e)
324:         {
325:             // ...
326:         }
327:
328:         private void btnClose_Click(object sender, EventArgs e)
329:         {
330:             // ...
331:         }
332:
333:         private void btnApply_Click(object sender, EventArgs e)
334:         {
335:             // ...
336:         }
337:
338:         private void btnDefault_Click(object sender, EventArgs e)
339:         {
340:             // ...
341:         }
342:
343:         private void btnLoad_Click(object sender, EventArgs e)
344:         {
345:             // ...
346:         }
347:
348:         private void btnSave_Click(object sender, EventArgs e)
349:         {
350:             // ...
351:         }
352:
353:         private void btnPrint_Click(object sender, EventArgs e)
354:         {
355:             // ...
356:         }
357:
358:         private void btnExit_Click(object sender, EventArgs e)
359:         {
360:             // ...
361:         }
362:
363:         private void btnHelp_Click(object sender, EventArgs e)
364:         {
365:             // ...
366:         }
367:
368:         private void btnClose_Click(object sender, EventArgs e)
369:         {
370:             // ...
371:         }
372:
373:         private void btnApply_Click(object sender, EventArgs e)
374:         {
375:             // ...
376:         }
377:
378:         private void btnDefault_Click(object sender, EventArgs e)
379:         {
380:             // ...
381:         }
382:
383:         private void btnLoad_Click(object sender, EventArgs e)
384:         {
385:             // ...
386:         }
387:
388:         private void btnSave_Click(object sender, EventArgs e)
389:         {
390:             // ...
391:         }
392:
393:         private void btnPrint_Click(object sender, EventArgs e)
394:         {
395:             // ...
396:         }
397:
398:         private void btnExit_Click(object sender, EventArgs e)
399:         {
400:             // ...
401:         }
402:
403:         private void btnHelp_Click(object sender, EventArgs e)
404:         {
405:             // ...
406:         }
407:
408:         private void btnClose_Click(object sender, EventArgs e)
409:         {
410:             // ...
411:         }
412:
413:         private void btnApply_Click(object sender, EventArgs e)
414:         {
415:             // ...
416:         }
417:
418:         private void btnDefault_Click(object sender, EventArgs e)
419:         {
420:             // ...
421:         }
422:
423:         private void btnLoad_Click(object sender, EventArgs e)
424:         {
425:             // ...
426:         }
427:
428:         private void btnSave_Click(object sender, EventArgs e)
429:         {
430:             // ...
431:         }
432:
433:         private void btnPrint_Click(object sender, EventArgs e)
434:         {
435:             // ...
436:         }
437:
438:         private void btnExit_Click(object sender, EventArgs e)
439:         {
440:             // ...
441:         }
442:
443:         private void btnHelp_Click(object sender, EventArgs e)
444:         {
445:             // ...
446:         }
447:
448:         private void btnClose_Click(object sender, EventArgs e)
449:         {
450:             // ...
451:         }
452:
453:         private void btnApply_Click(object sender, EventArgs e)
454:         {
455:             // ...
456:         }
457:
458:         private void btnDefault_Click(object sender, EventArgs e)
459:         {
460:             // ...
461:         }
462:
46
```

The new season brings with it a chance to reconnect with my winter family, the excitement of cheering on our youth and junior teams, and the extra coffee required the morning after a late night at the rink. This season is going to be especially exciting for the local ice community as we are expanding our facility from one to two sheets of ice. Currently our single sheet of ice is booked solid from 6:00 am to midnight, seven days a week, seven months a year.

Ice Building

Take a few minutes and **enjoy the video below**, surveying the local ice community and the amazing work that is being done by the Gallatin Ice Foundation to bring hockey and other ice related activities to more and more people.

As you look back on 2016 and forward to 2017, I strongly encourage you to head to your local rink, put on a pair of skates, and go skating. And hey, you never know, the first day of ice at the ice rink may just become your favorite day of the year, after the Summer solstice of course.

[\[back to top\]](#)

Custom Report Logo

Professional IES Reports with Your Company Logo

Show Your Colors

You may want to show reports from IES products to your clients, or building officials, and owners. You can give them corporate authority with your company logo.

This feature has been in some of our products for a long time, and others are just picking it up. Here is the scoop for current products:

ShapeBuilder 7.0-9.0

Drop a "ReportLogo.png" or "ReportLogo.jpg" image in the IES\Customer folder and it will get used automatically. You can access this normally hidden folder using the **Custom Data** button on the **HOME** tool bar. We may add a preference setting soon, to make it less mysterious.



VisualFoundation 7.0

Use the **Home | Preferences, Report** command to specify an image file.

Quick Products

Use the **Options | Preferences, Users** command to specify an image file. For best results use a gray background.

VisualAnalysis 12.0

Sorry, no direct support in the current version. Use **File | Save Report As...**, and then insert your logo in MS Word, Excel, or a similar tool. The next upgrade will offer direct custom logo support, just like in VisualFoundation.

Per Machine

Please note that these changes need to be done on each machine you use. We recommend putting your image file in a place where it won't get renamed or deleted...such as the ShapeBuilder location mentioned above.

Image Tips

For best results you want to use an image that is not TOO wide or TOO tall and skinny, and it should probably have a resolution (number of pixels) corresponding to about 1-2 inches, at 300 dots per inch. Small images will get fuzzy, larger images may hinder performance.

[\[back to top\]](#)

SE Diversity Project

IES Customers are Creative!

IES Wish List

You may feel like your projects are routing or boring: like a K-Mart building. Yet, most likely you have also done some very unusual or interesting work!

Around the world, engineers using IES tools are tasked with making whims and dreams stand tall, or zig-zag, or bend around obstacles! **You make them stand** reliably.

At IES, we are constantly suprised by customer projects. But we have likely **not seen a single one** of your projects! This makes us very, very sad.

Show & Tell!

We are putting together an online book or video showing the versatility, range, and achievements of IES customers. Please share your accomplishments with IES and your peers. Nothing fancy, just an art project.

Simply **send a project file or photograph** or both! Email it or FedEx it, we don't care how. Just get it to us.

Tell us as much or as little as you like about the project. Then tell us if you want to **remain anonymous or get credit** for your company (or *give credit* to others involved). Include photos of people involved and your corporate logo too. Let's make this happen!

Our Goal is 100 Projects!

Kudos & Prizes

To help us reach this goal, IES will offer **random prizes** of marginal value to participants (just like Cracker Jack). We will also engrave your name on a **bronze plaque** for the cornerstone of our amazing new headquarters building, if we ever build one.

[\[back to top\]](#)



Coming Soon

Some of the New Features in VA 17

Sneak Peak

Below is a list of some of the **new features** that will be arriving soon in VisualAnalysis 17. There is actually **MUCH** more on the way.

Extreme Speed

- Area Load Generation: 10x Faster
- Analysis in the Background
- Design Checks, In-Parallel & Background
- DirectX Graphics
- Report a 100 pages, in Seconds

Intelligent Engineering

- Automated Combined Members
- Area Sides are Optional
- Plate-Element Releases
- SJI Virtual Joists
- Import DXF Circles, Ellipses, and more
- More Model Checks
- Design Search offers more Control & Flexibility
- Optimize Concrete Slab Reinforcement
- Base Shear for Response Analysis

Powerful Capabilities

- Merge Project Files
- 64-bit (No Memory Limits)
- Drag & Drop Report Creation

Accurate and Precise

- Snap Points for Drawing Elements
- Drag a Line for Plate Moment & Shear
- Easily Scale all Loads in a Service Case
- Fewer Clicks for Common Actions
- Automatic Named Model Colors
- Preview Image for Saved Project Files

[\[back to top\]](#)

[Meet IES, Inc.](#) [Disclaimer](#) [Privacy Statement](#) [Trademarks](#)
Copyright © 1994-2018, IES, Inc. All rights reserved.