

## 整理人：洋葱头

博客地址：

51testing: <http://www.51testing.com/index.php?uid-246015>

博客都刚刚起步，希望大家多给意见

如果您有时间意见和建议，[请发送邮件到邮箱 178656722@qq.com](mailto:178656722@qq.com)，非常乐意和广大测试者们交流。

您可以转载此文，但是请不要作为商业使用，本文所有内容仅代表作者个人意见。

## 1 安装卸载类：

- 1、在已经安装软件的情况下，再次进行安装，表现是否正常（比如提示是否升级、检测到已安装），需要考虑已安装和现安装版本差异问题
- 2、各种杀毒软件（卡巴、瑞星、360）对安装程序的影响
- 3、是否能在控制面板里面卸载
- 4、安装后快速启动、桌面、开始程序里面的快捷方式情况
- 5、卸载时是否退出客户端（退出和不退出都要考虑），卸载后的表现
- 6、安装的程序是否带有插件  
带有微软的 **framework**，而影响用户的安装和使用
- 7、安装目录的考虑（中英字符、长度、空目录、根目录、修改目录、默认目录）
- 8、**是否需要考虑在虚拟机中的安装使用？**
- 9、各个版本的安装包大小，客户端产品是需要下载的，所以包的大小对用户来说比较重要

## 2 字符（串）类（可输入编辑框或者文本框等也会涉及到）

- 1、需要考虑字符串长度、字符类型（中文、英文、数字等）、编码类型、如果是英文，还会涉及到大小写的区别。
- 2、全空格的考虑情况，字符中间含有空格，最前和最后包含空格情况考虑
- 3、涉及到编码的，要看各个编码下的显示是否正确，以及各个编码之间
- 4、当有限制长度类的输入时，需要考虑长度刚好达到限制和超过限制后仍然进行输入的情况，也就是需要考虑边界值。
- 5、对于只能输入字符的地方，尝试输入其他字符比如 汉字，看看操作表现是什么样子。

## 3 界面类

- 1、应用程序所有可点击地方是否可以进行操作，菜单、按钮、超链接（文字颜色以及是否能正常超链）、文字等。

- 2、各种操作对应的正确、错误类提示信息是否正确
- 3、窗口的缩放（双击的最大最小，点击按钮的最大最小，关闭）、拖动（开多个窗口拖动）、任务栏（左键单击和右键单击的操作）、托盘区、**任务管理器操作**  
一般客户端软件，开着窗口在桌面上移动的时候，cpu 占用都比较高，这个性能需要控制在某个合适的范围内。
- 4、需要考虑窗口的模态性问题，比如有模态窗口的时候，进行其他的操作，以及模态窗口的重绘等。
- 5、需要考虑软件对键盘上各个键的响应情况，最多用的是 enter、shift、ctrl、上下左右箭头，home，vendors，pgup，pgdn，del，对 tab 键的支持等。还要考虑各种热键（全局热键和软件自身的热键）是否能正确响应。
- 6、各种控件的表现和操作是否正常，下拉列表、日历控件等
- 7、如果有托盘图标，需要考虑托盘图标的显示状态，是否能显示，操作是否正常等
- 8、软件的 tooltip 是否正确合理齐全
- 9、如果有排序类功能，排序是否正确，如果不正确，和 windows 系统本身的排序进行对比，看是否一致（例如中文在英文之后，英文是否区分大小写）
- 10、操作界面的即使动态刷新
- 11、如果设计到焦点切换的，需要看鼠标的焦点切换是否正常，适合用户使用习惯。
- 12、涉及到列表类显示的，要看是否显示翻页，翻页是否正常
- 13、涉及到编辑框的，要看输入内容过多之后，是否有滚轮
- 14、窗口在屏幕上的位置是否需要具有记忆能力，比如某个窗口操作一次后，下次打开的位置定位在哪里？
- 15、有的客户端软件要求有飘窗类的提示，需要测试再不同情况下是否能出来，比如最小化到托盘、任务栏以及用 ctrl+D 显示桌面，是否能正常出来飘窗
- 16、需要考虑再不同显示器上的显示，各种比例和分辨率下的现实情况。
- 17、对换行符的处理，有的显示、输入区，如果有换行符的话可能会出现  
测试遇到过含有换行符的话，后面的内容无法显示出来。
- 18、一些操作状态的延续变化，很难发现啊。  
邮件列表中，在某个分组上点击右键，不放鼠标，将鼠标拖动到分组下的列表上，出现右键菜单不一致的 bug。
- 19、对任务栏的考虑，要考虑任务栏在下方以及在屏幕上下左右侧的情况

## 4 兼容性

- 1、在中英文系统上使用的区别，在控制面板的区域和语言选项里面进行设置，管理选项卡里更改系统区域设置。
- 2、在不同操作系统上使用的区别（XP,VISTA,WIN 7,2000,2003）
- 3、在远程操作电脑的时候使用情况，测试的时候遇到过远程操作的时候会可能崩溃的错误。
- 4、浏览器：不同 IE 浏览器、带标签页和没有标签页，同一个 IE 浏览器不同版本的
- 5、同一个系统的不同系统用户操作（管理员和非管理员）
- 6、需要考虑不同分辨率，屏幕大小下是否能合适的显示。
- 7、需要考虑各种浏览器的缓存情况，会不会因为缓存而对测试产生影响
- 8、对于需要输入文字的地方需要考虑多种输入法切换是否能正常输入。输入达到限制后，再继续输入，是否有问题

- 9、在 32 位和 64 位系统上都需要进行测试。特别是对新的 64 位系统的支持度。
- 10、需要操作系统，比如 sp1 sp2 sp3 等，其他很多操作，可能会有影响的地方都需要考虑一下。
- 11、需要考虑计算机休眠、待机后再启动软件的表现情况，（还有待机）  
各种杀毒软件对软件的影响。瑞星、卡巴、360 等  
杀毒软件对一些文件类型、端口等有监控，需要考虑。可能由于软件使用某些端口而被杀毒软件阻止而导致不能正常使用
- 12、jpeg 格式图片有灰度图和 RGB 格式图片，都需要测试到。
- 13、考虑文件系统格式 fat32 /ntfs 下区别，比如 fat32 下有单个文件 4G 大小的限制等

## 5 用户体验类

- 1、界面文字提示是否友好、易懂、简练（因为用户都是懒惰的，不愿意看复杂的东西）
- 2、操作流程是否清晰，用户知道自己每步都是在做什么
- 3、有错误类信息，不要使用代码类文字，考虑到用户群体的情况，还要区分中英文（用哪个更好）

## 6 上传下载传输类

- 1、上传是否超过最大容量、流量限制
- 2、上传格式
- 3、需要考虑不传输文件、传输文件内容为空（大小为 0KB，边界值考虑）、文件内容包含特殊字符、文件名字符
- 4、涉及到网络传输，和端口有关系的，要考虑模拟一下端口错误，封端口的操作（需要补充具体如何封端口）
- 5、和网络有关系的要考虑使用代理的情况下，软件的运行状况，  
在传输中设置错误的代理，本地传输并没有受影响（自动收信过程中，设置了代理，但是自动收信还能继续），不受影响应该是正确的。
- 6、上传下载文件，考虑本地文件，还要考虑 ftp，http 上的文件。

## 7 I/O 读取类

- 1、需要考虑磁盘空间不足的情况
- 2、考虑同不同目录下相同文件的操作情况（比如邮件附件，两次添加同目录下的一个文件和分别添加不同目录下的相同文件的表现）和同目录下同名文件的重复操作
- 3、正在使用的文件是否是独占状态
- 4、涉及到文件操作时要考虑文件的类型（例如：txt、doc、gif、png、jpg。。。。）、大小（0KB，正常、极大，其实也就是临界值考虑）
- 5、涉及到导入导出类操作的，需要查看导入导出过程中各种表现是否需要同步变化
- 6、涉及到文件保存时，需要考虑文件保存的类型、名称的默认给出。
- 7、文件拖动类的考虑

有的应用程序可以上传、下载、保存文件，那么拖动这些文件试试，看是否会有问

题。

例如:对于 foxmail 邮箱这个软件,可以携带附件,那么试图拖动文件到附件区,或者从附件区拖动附件到文件夹,任务栏,或者拖动到程序中其他地方。另外,发现附件可以直接拖动到正文区进行显示的(新发现的功能,应该是编辑区的控件本身就支持吧,呵呵,惊讶了一把,居然还有这个功能,似乎很方便)。

- 8、系统对单个文件夹大小做限制, ntfs 和 fat 格式的系统对单个文件大小有限制
- 9、图片文件原本为 jpg 格式的,但是修改后缀为 gif 后添加到表情 或者插入到其他地方。出现不能识别的问题。因为其他控件按照后缀先判断为 gif 格式,再走 gif 格式流程处理,但是实际上图片本身是 jpg 格式的
- 10、涉及到文件写入读取的,需要考虑移动设备,比如 U 盘、硬盘、ftp 等

## 8 性能类

- 1、单核、双核的区别
- 2、内存大小的区别
- 3、同一个操作涉及不同的文件大小的时候, PC 的反应(例如传输大文件和小文件)
- 4、涉及到网络操作时,超时是否及时、提示是否合理
- 5、是否有 GDI 泄漏(界面?)
- 6、使用过程中 cpu、内存的占用情况

## 9 检索、过滤、搜索类

- 1、对分词的检索是否准确,比如如果检索 ab,那么 a b 是否会被检索出来?要视要求而定。
- 2、搜索的时候,对不同格式的文件内容,是否能够正常搜索,比如 HTML 格式和 txt 格式之间的区别,因为 HTML 格式本身含有标签以及其他一些内容,但是这些内容并不显示出来,所以搜索的时候是否需要搜索这些内容,需要进行考虑
- 3、搜索匹配时,对中英文的支持度(比如输入英文能否匹配中文,输入中文,能否匹配英文等。)

## 10 其他

- 1、客户端类软件,需要注意到开启的各个窗口之间数据同步一致性问题,各个窗口之间事件触发是否会马上在其他窗口或者界面响应。
- 2、考虑界面上文字、各个窗口之间需要保持一致的文字说明。(诸如相同属性名称文字提示信息等)
- 3、同一个操作涉及到的不同状态变化是否正常。(例如,点击某个链接,文字颜色是否变化,点击某个按钮,按钮颜色或者属性是否变化等)
- 4、使用软件的过程中,多关注 cpu、内存、句柄占用等方面的情况。

- 5、要能多考虑各种异常情况（磁盘空间不足、文件占用、网络断掉、断电、手动切进程模拟异常退出）
- 6、涉及到对文件目录的操作，需要考虑是否能记住/清除原来使用过的文件目录。如果是新建，要考虑是否可以新建成功（windows 对新建文件的字符限制）
- 7、同一类的界面表现、操作应该尽量保持一致。（? 没有描述好）
- 8、要多考虑进行了一个操作/设置后，可能会影响的其他方面，同步表现是否正常，设置是否有效等。
- 9、和服务器有相关的一些操作，都要考虑一些操作是在客户端处理，还是在服务器端处理的。服务器和客户端之间的一些交互返回信息，比如错误码等。

## 11、 个人想法总结类

- 1、 写总结、bug 类语言描述一定要慎重，多读几遍，以便让其他人更能看明白，避免求快而写错别字，用错术语。总结类需要写的更专业一些，避免通俗的、磨蹭两可的语言描述。宁可多花时间少写内容，少报 bug，也不要报上去的 bug，给别人看的总结出现过多纰漏，没有发现的 bug 可能是工作失误，但是发现了，但是却有不描述好，或者自己描述的不确定后事后自己都解释不清楚的话，那就更糟糕了，更上级看的总结也是如此，及时发现 bug 再多，总结却是评价你这次测试的一个方面，如果总结写的很差，必然给领导留下很差的印象，或者总是在受到领导的批评。总之，三思而后行，是没错的，也许某些时候会降低工作效率，但是有时候，出现错误带来的负面影响比工作效率低下带来的负面影响更大。
- 2、 开发对于一个软件安装和使用中生成的各种文件，最好有一份比较好的说明文档，，当然开发可能没有时间去写，而且公司里面如果没有强行要求的话，他们也是不会写的，所以测试人员就只能自己多去钻研了，对于这些文件的了解对于测试也是很有必要的。遇到不懂的要及时跟开发沟通询问。有时候可能需要花费比较多的时间来了解开发的一些处理流程和文件具体含义（比如一些 XML 文件具体保存的是什么内容），这就需要协调和测试时间的冲突，因为要花时间了解，所以测试必然会耽误时间，但是了解之后却有利于进行某些功能的测试。慢慢改进吧。
- 3、 不属于自己的任务，还是不要多去做的好
- 4、 有时候自己提出来的产品问题，不一定会被领导、策划或者其他相关开发人员接受，除非等到产品发展部提出来。
- 5、 测试中，只要有一点问题，就应该及时提出来，如果自己用的不顺手，或者觉得不合理的，自己多记录和总结，虽然不一定会被公司采纳，但是可以作为自己的总结类内容，整理出来。

## 12、 可用性用户体验

- 1、 跟网络有关系的，对网络错误的提示，有的需要及时，有的不需要频繁提示网络错误，应该多提供几次重连，比如三次，如果重连三次都发现网络错误连接失败，就提示用户，否则太频繁会有骚扰和降低用户对产品的信赖
- 2、 给用户提供的操作，用户可以用，也可以选择不用，所以界面上需要提供取消类的入口，否则强制性的使用体验上不是很好，比如提供上一步类的入口也类似。
- 3、 需要判断重复性的操作（已经安装、已经导入、已经。。。）是否能提示用户
- 4、 涉及到告诉用户文件类型的操作，应该尽可能明确的给予显示类型，因为不是很多用户对文件类型有概念。比如如果某个功能需要导入 txt 格式文件，尽量做到能自动检测显示出来，而不是让用户自己去找
- 5、 像日历这种控件，不仅仅需要提供月更改入口，还要提供便利的年更改的入口
- 6、 对于一些快捷键，能给予 tip 或者附带在文字后面的，尽量让用户可见，否则让用户揣测，那太不人性化了点吧。
- 7、 发现\*\*\*和其他圆角的窗口有同样的情况，最大化时鼠标移到屏幕的最右上角点击，如果没点中按钮而是正好点在圆角的地方，则关掉的不是闪电邮而是它后面的窗口，比如浏览器……因为我经常把鼠标往右上角一推就按，不会去找按钮，所以好几次了。不过这倒也不太算是毛病..