PB DeCompiler Question & Answer

How to use PB DeCompiler?

如何使用 PB 反编译工具？

Step1：open file

第一步：开启文件

Step2：Export PBL

第二步：导出 PBL

Step3: Export PBW & PBT

第三步：导出 PBW 和 PBT 文件

Step4：open pbw with PB-IDE(PB)

第四步：在 PB 中打开 PBW

Step5：Optimize every PBL

第五步：优化每个 PBL

Step6: Compile One of PBLs(Full Compile Mode)

第六步：编译其中一个 PBL(完全编译)

Step7: correct the syntax error unitl without error-prompt.

第七步：修正语法错误直到无错

Step8: Do other Work…

第八步：其他的修改和编辑

*详细的操作请看 PPT 文档的演示。

常规问题解答----------

如何购买正式版?

请按软件上方显示的邮件，QQ，MSN 联系作者购买。

软件的价格是多少？

软件的售价根据不同的版本以及一次购买的版本数而有所不同，具体请咨询作者。

测试版和正式版的差别

测试版仅仅提供部分代码以供评估产品。限制主要表现为限制部分事件不显示出来，而且过长的代码在其后半部分会截去部分代码。

正式版是否支持导出 PBL 和 PBW 以及 PBT

正式版支持全部的功能。

在 PB 中双击打开对象，代码看起来是错乱的，比如某个窗口的 open 中是 create 控件的代码。如果处理？

这是一个无法解决的 BUG，就是 PB 对手工生成的 PBL 的代码认为是没有变化的，所以未重新优化和编译，要解决这个问题。你可以在某个对象上单击右键，选择 EDIT SOURCE，进入源码直接编辑状态，在任意空白处，输入一个空格(不要插入到代码中间),然后保存即可消除这个问题。

另外导入 sr 也可以消除这个问题，因为导入时 PB 会优化和编译该对象，但是如果你的 sr 是被修正过的语法错误的，就不要采用此方法，请用前面的方式。

如何导出 PBD，DLL 和 EXE 中的 dw 的代码？

请使用本软件目录中的 RecoveryDW 中对应的版本，按提示选择文件并导出所有的 dw 源码。然后在 PB 中对应的 PBL 上点右键，选择 Import 菜单，并选择对应此 PBL 名字的文件夹内

的所有源码*.srd 进行导入。

如何导出图片？

用导出 Sr 的方式可以导出图片。

在编译一个 PBL 后，如何按照错误提示修正错误的语法？

以这个错误为例：

---------- Regenerate: Regenerating Object w_1

pb100.pbl(w_1).tab_1.create.3: Error C0031: Syntax error

---------- Finished Regenerating Object w_1

最前面是 PBL 的名字(pb100.pbl)，括号内是对象名称(w_1)，后面是控件名(tab_1)，然后是事件名(create),末尾是错误所在的行号(3).

如果是全局函数：提示如下

pbabc.pbl(f_get_group).338: Error C0031: Syntax error

最前面是 PBL 名称，括号内为全局函数名，括号后为错误行数

在错误的编译日志信息那行，右键单击，选择菜单：Edit Source，进入源码编辑窗口，搜索控件和事件名，在 PB DeCompiler 输出源码时，已经加了备注，你只需要输入“tab_1.create”就能准确地找到位置，一个例外是函数的重载，可能会有多个重名的函数，这个需要注意。

on tab_1.create //这里算代码的第一行

this.tabpage_1=create tabpage_1 //这是第二行

this.Control[]={this.tabpage_1 //这里就是错误信息指示的第三行.看出它缺少一个花括号，添加后保存即可正确。

end on

在这个编辑界面，当你鼠标点击到某行时，右下角的状态栏会提示行号和列号。如果错误的代码行数比较大，需要你做一个简单的加法，换言之，当你鼠标点击到 on tab_1.create 时显示行号为 98，然后加上 3 就是 101 行就是错误所在的行。

---------- Regenerate: Regenerating Object w_1

pb90.pbl(w_1).6: Error C0031: Syntax error

---------- Finished Regenerating Object w_1

这个错误是在整个 Source Edit 中的第六行,也就是整个对象的源码的第六行.

反编译的程序运行时还有一些运行结果的错误，怎么修正？

反编译软件仅仅能够保证足够少的语法错误和尽可能地恢复代码的原样，至于运行时的错误，需要你根据错误所在的位置设置断点，然后逐步调试去确定错误的原因，从而纠正错误。这个是程序员的必要工作，在此不再赘述。

换言之，错误并不一定是语法错误，也可能造成逻辑错误。

语法错误在编译时提示，逻辑错误在运行时才可以发现，并更为隐蔽。

提供怎样的技术支持？

技术支持的时间，一般在北京时间的 10-22 点可以提供 QQ，MSN 和邮件支持。也提供中文的电话支持。非中文技术支持请通过邮件或者 MSN 联系我，请注意只是用英文，因为我的电脑只能显示中文或者英文。中文包含繁体，这个不是问题。

是否提供离线的认证？

暂时只提供在线验证模式。服务器会 365*24H 开启。遇到无法链接服务器的情况请用最合适的方式通知我。服务器也设置有守护程序定时去检测保持稳定，以及出现故障后第一时间

通过手机短信通知作者维护。因为这个软件不是一个实时的运用，所以短暂的停顿应该不会影响到正常的工作。有时遇到机房切换服务器或者 IP 地址，域名解析生效在全球可能刷新时间会需要 24H，这是耽搁最久的维护事件。一般的网络阻塞，会在几分钟，或者一个钟后恢复的。

网络认证需要的端口？

认证需要客户端能访问 80 端口。

请保证你可以访问以下网址：

Ushost.mis2erp.com, cnhost.mis2erp.com, pbde.mis2erp.com

你在浏览器内输入网址应该可以访问到 PB DeCompiler 产品的页面。如果遇到无法访问的情况请 Ping 测试一下先，有遇到客户方 DNS 无法提供解析的情况(如果 DNS 解析正确 ping 域名应该跟 ping IP 一样显示响应)，请更换一个 DNS 试试。

另外，有些用户使用硬件宽带路由器，往往可以上 QQ 却无法认证，请重启路由器再测试。

如果客户出差在外面，各种网络环境非常复杂，有些内部网络对上网有严格地限制，比如 IP 地址，网卡 MAC，外部 IP 或者域名过滤等。一般请在能直接上网的情况下测试，如果遇到问题再请报故障。

客户是用 proxy 代理软件上网，怎样使用该软件？

因为现在使用代理软件上网的情况少之又少, 使用路由器代理上网的情况占绝大多数。

所以本软件未支持代理上网方式。具体可以使用 HA-Proxifier 这个软件来代理 PC 上所有的外部访问。这个软件通过 SOCKS5 代理可以让本软件通过在线认证。具体可来邮件索取软件以及连接配置图示。

哪些版本支持 PBW 和 PBT 直接开启方式，如果没有 PBW 怎么开启工程？

自 PB8 开始使用 PBW 和 PBT。

对于较低的版本：

PB7：工具栏上有两个按钮：Library 和 Library List。

前者可以列出目录，给你选择需要打开的 PBL 文件。

后者用于设置该工程的 PBL 搜索路径。其实就是要求反编译后，在这里把全部的 PBL 路径都选择进去,前提是要用第一个按钮找到 APL 对象所在的 PBL 并双击打开切换到当前工程。

PB6.5：用 Library 列出所有的 PBL，然后找出含有 APL 对象的 PBL，在 APL 上双击打开，点击工具栏按钮的 Properties，在 Libraries 标签页，设置该工程的 Library List。

PB5.0 同 PB6.5.

如何在工程中添加一个新的 PBL 或者附加一个已经存在的 PBL？

PB8 以上，在左边的 TreeView 中间中 Target 名字上单击右键选择 Properties 菜单，然后在 Library List 标签页进行添加(New)或者附加(Browse)。

PB7: 在工具栏打开 Library 按钮，然后在出现的新工具栏中选择 Create Library 按钮。
在工具栏 Library List 中添加存在的 PBL。

PB6.5：在工具栏打开 Library 按钮，然后在出现的新工具栏中选择 Create Library 按钮。
在 APL 右键，Properties 的标签页 Library List 中添加。

PB5.0：同 PB6.5 操作。

Call super 语句在 PB 中怎么看不到？

例子：

```
Call super::timer;
Int i
I = 100
```

在 PB 中，继承时，比如 w_1 有事件 Open，是祖先对象，w_2 继承自 w_1.

当在 w_2 的 open 中写代码时，点右键有个菜单选项：Extend ancestor script.如果打钩，就是扩展代码，Pb 会在你写的代码前面插入：call super::open; 这样的代码。并且这个代码是 PB 在后台悄悄添加的，在编辑状态无法见到，只有在 Edit Source 直接编辑源码时才能看到。在反编译中，是可以直接看到这句，因为它是完整代码的一部分。只是通过程序员的选择，可以让 PB 添加和删除掉这一句。注意，这句是在 w_2.open 事件代码之前(变量声明之前).因为它被保证在任何后继者代码执行前被执行。

Dw 的源码 srd 在导入 PBL 时出错怎么办？

用记事本打开 srd 文件，查看语法是否完整。

如果语法没问题。请在 PBL 随便建立一个 dw，grid 风格，external 数据源，保存，然后点 Edit source，然后 copy 反编译导出的 srd(全选),粘贴到源码编辑窗口，保存，如果有错误再按提示处理。

PBD 中的 bin 对象是什么？

Bin 对象是窗口或者用户对象中插入 Ole 对象时，形成的二进制文件。在导出 sr 文件时，有附加到源码的后面，而在导出 PBL 时，有写入到新的 PBL 中，用户不用管它。

为什么从 PB DeCompiler 中看，某些对象前面没加号，看起来好像是未处理？

在一般的开发中，会碰到在一个工程中重复写了两个同名对象的情况，比如 w_print.存在两个不同的 PBD 中，这一般是由于工程缺少检核造成的。也可能是对象复制到另外的 PBL 时造成的。PB 在运行时，会搜索对象名，所以，排在后面的对象不会被搜索和执行到。同理，反编译器只会找到第一个对象进行分析，第二个不会被分析。

在 PB 中导入 SRD 时，有时选择完文件后，无任何反映，或者 IDE 会崩溃，为什么？

经过测试，以 PB9 为例，我测试的情况是：当一次选择到最多 154 个 SRD 时，可以导入成功，而超过 154 个 SRD 时，PB9 无反应。可能是该工具设计所限，并不是什么大问题。因为 PB 的导入主要就是用于少量对象的移植之用。并不是做大量的恢复工作。

如果导入导致 IDE 崩溃后，在次开启，就算一次选择 60 个 SRD 再次导入也可能仍然使其崩溃。可能是损坏了文件，你可以删除该 PBL，新建一个 PBL 再重新导入。所以在此建议一次仅仅导入 50-100 个之间较为妥当。当然话说回来，这不是什么大问题。

对于 Full 编译时崩溃的情况如何处理？

在反编译出现某种的语法错误，我曾经遇到就是把 SQL 语句中两个单引号的字符折行到新的一行时，会出现崩溃的情况。其实 PB 在处理源码时，可能会遇到无法无法规避的情况，比如缓冲器溢出，或者访问到越界的地址。

解决的方法就是在编译时，仔细看是在编译到哪个文件崩溃的，比如 abc.pbl 时崩溃，然后将该 PBL 重新在反编译器中导出一次，导出时选择首选项：“将程序代码作为注释(注释掉所有代码)”，这样导出的代码段都被注释起来。然后复制到新工程的文件夹内，覆盖第一次导出的 pbl。重新编译。然后在逐个对象取消被注释的代码段。直到找出错误所在。有些麻烦，但是有时候是必由之路。

当然在上述的过程中，你也可以暂时用原来的 abc.pbd 附加进来进行编译以达到排除存在问题的文件的目的。

当用 ReCoveryDw 中的程序导出 dw 时，有时 dw 的 srd 内为空?

在导出时，当看到成功导出的提示后，不要马上关闭窗口，请稍等 20 秒，等文件成功关闭。(因为 Messasgebox 这种窗口，在很多处理过程之前弹出，阻止其他过程的完成)

还有种奇怪的错误，就是无法 optimize，也无法编译。

你可以按顺序双击对象，如果提示

Open of function(or window) xxx failed. Possible causes:

1- object does not exist
2- its ancestor not exist
3- ……
请点右键，source edit 进去，在任意空白处敲空格(不要改变程序)后保存。
原因，导出的 pbl 无法被 pb 识别。这样保存后，pb 重新读取和优化。就 Ok 了。
上述现象也可以导致双击对象时 IDE 崩溃。其原因也是读取源码是 PB 认不到。这样敲空格后可以正常。
导出 sr 再导入也可以修正此问题，但是有时需要导入很多次方可解决。（IDE 崩溃的情况下）
IDE 崩溃有种情况就是遇到严重的无法承受的语法错误。会崩溃。处理方法就是比如一个 function 双击崩溃，在 source edit 中注释掉代码。就可以双击打开了。
遇到此问题，可以在选项》源码输出选项》将代码作为注释。这样可以减少此问题。然后附加这种方式导出的 pbl 到新工程，并拖动此对象到出错的 PBL 中。这样。再双击打开对象，并手工取消掉注释符号，并保持，按提示错误位置修正。

遇到一个 exe 文件，没有 pbd。反编译后为一个 PBL，用 PB 打开出错，IDE 崩溃退出？

方法有两个：

一．随便新建一个 workspace，随便一个 target，比如叫 pbde001。
然后在 target 上右键，属性，附加导出的 PBL。然后 optimize 它。如果没错，删除导出的 PBL 中的 apl 对象。然后从反编译中复制 apl 的源码，到 pbde001 对象的 souredit 中，保存，注意修改源码中的 apl 对象的名字，如原来的叫 abc，改为你新建的 app 名称：pbde001，然后保存关闭 app 并编译。我处理过一个特例，这样能打开它并继续下一步，否则直接打开一直崩溃。
二．随便新建一个 workspace，随便一个 target，比如叫 pbde001。
然后在 target 上右键，属性，附加导出的 PBL。然后 optimize 它。
然后 File->Open workspace 重新选择导出的 pbw 名即可。

有用户报告称在 win2k3 上用 PB9 编译导出的 PBL 导致程序出错而换 win7 编译成功，具体原因不明。

PB 在不同操作系统上的差异早已有之。目前发现的情况是，早期的版本无法适应 WIN7 和 WIN8，或者兼容性有问题。有人尝试在虚拟机中安装 PB9 等，也出现无法 debug 的问题。
就目前的状态看，PB 的 5.0 到 12.5 应该在兼容性较好的 XP SP3 中作为反编译和编程的基本环境。不同于 XP 的系统，应该先做兼容性测试，以排除 IDE 不适应 OS 的问题。
目前来说，

用 Powershield 处理过的文件反编译时非常慢，为什么？

用 Powershield 处理过的文件,反编译时需要经过 60(截止 v20131001)步的特别逆向处理，代码会被逆向算法扫描多达 60 次。时间大概为未混淆过的代码的 6-10 倍。所以感觉会比较慢。请耐心等待，别无他法。

修正语法错误是否按编译时的 ERROR 日志的顺序进行处理？

不一定按这个顺序，原则上是按顺序，不过在错误比较多的情况下，出于心理宽慰，你可以先找容易修正的错误提示来处理。原则上我是从代码较短的程序段错误开始处理，例如：
pb100.pbl(w_1).tab_1.create.3: Error C0031: Syntax error
pb100.pbl(w_1).tab_1.create.103: Error C0031: Syntax error
如上，错误在 103 行，代码一定非常长。整个逻辑较为复杂，所以我先处理第一句。
其次从代码提示较为明显的错误处理。Syntax error 只是说语法错误，并不是明显提示。

另外一个惯例就是从一个提示行数很多的对象开始处理，因为处理好一个，会减少很多行(只是心理会感觉好，其实先处理哪一个效果都是一样的)
当然，原则上从上到下处理，因为考虑对象的依赖(引用)关系。

这种警告如何处理？

C0149: The identifier 'ls_no' conflicts with an existing property with this name in the parent class. The new definition of 'ls_no' will take precedence and the prior value will be ignored until this version of 'ls_no' goes out of scope

这种错误是因为 ls_no 这个变量名既是本地变量，也是窗口的实例变量。
提示的意思是本地变量在本代码段，将屏蔽实例变量，实例变量在这个代码段范围不起作用(不可见)。出现这个提示，不是反编译的问题，因为代码就是如此。你应该根据上下文来决定是否需要删除本地变量或者实例变量。本地变量只影响到本段代码的执行，而实例变量可能影响到任何引用它的地方。所以你需要酌情取舍才能让这种警告消失。
在这类系统定义好的事件中，如 pbm_mouseactive，它有声明一个参数，而且名字也为 message，如果你代码中使用到 message 这个变量，也会出现标题上类似的警告，它提示你 message 与全局的变量 message 发生冲突。可以不用理会它。

Optimize 时提示错误?

Optimize failed. Possible causes:

1-Library has no entries

2-Invalid library name

3-Insufficient disk space for optimize

你可以多执行一次，第二次或者第三次执行时无错就没问题(PB 尽是这样的提示)。
注意，如果是空的 PBL，会一直提示这个错误，不用管它。

Warning C0190: Instance variables of local structure type ('str_mousepos') will be implicitly private in the next release.

长期的，在多个版本中都这样提示，原因是后续的 pb 升级中可能会把对象内部声明的结构做私有化处理(技术上讲，它这样实现有悖原则，试想在外部直接操作该变量时，需要预先知道这个内部的结构声明!!!)：

```
type variables
private str_mousepos i_mousepos          //声明时，加上 private 即可不报这个警告。
end variables
```

出现两个错误提示，但是找不到具体位置

---------- Regenerate: Regenerating Object w_printx
print.pbl(w_printx).w_printx.1: Error C0031: Syntax error
print.pbl(w_printx).em_copy.20: Error C0031: Syntax error
---------- Finished Regenerating Object w_printx

一般情况不太可能错在控件的声明位置(上面两个错误没提示具体的函数或者事件)。有经验表明，SQL 语句被错误折行后会产生这种类似错误。处理的方法：在该对象的 source edit 中，逐个注释掉某个函数或者事件中的代码，直到提示消失,即可找到错误的具体位置

我没有某个版本的 PB，能否用高一个版本的 PB 去打开？

原则上是用同一个版本的 PB 去打开
但是有个例外，就是比如 PB10.5 的程序反出来，用 PB10.5 去 optimize 没问题，但是编译时总是报错退出 PB-IDE，你就可以尝试直接用 PB11 来打开，如果语法错误较少，运气好，可以直接升级通过。

某些版本的 PB，存在一些固有的缺陷，比如容错能力不足（因为反编译可能存在语法错误！）你换高一个版本的 PB 或者高一点的 Build 版本，就能顺序通过。
至于哪些版本比较稳健，这个靠自己经验去确定。

我的文件数超过 130 个之多，感觉很慢，很耗内存

自 v2012.11.15 开始，增加了“检视->分批分析文件“菜单。可以分四次来分析较大的工程。由于内存占用特别大，请分析一批后，重启反编译软件再反编译第二批。
在第一批时，导出 pbw 和 pbt。
每批都将 PBL 导出到同一个文件夹即可。最后完成四批导出，就是一个完整的工程。

代码无语法错误，调试运行时报错：Error accessing external object property xxxx at line nnn in xxx event of object ole_xxx of w_xxx

原因，本机未注册某个外部控件(***.dll 或者***.ocx)，比如微软的 com 通讯组件：MsComm.ocx
从程序安装目录拷贝这些需要自带发行的 dll 和 ocx 组件文件，放入开发目录或者 window 系统目录，运行命令行：regsrv32 path***.ocx。如果放入系统目录，可以不带 path
注册后需要重启 powerbuilder IDE。

一个简单程序反编译后(代码已经被注释掉)，用 PB9 8836 加载无论如何都要崩溃，用 PB10 4500 加载不崩溃？

不同的 IDE 对错误的耐受程度不一样，可以考虑这样直接升级到较高或者更高一个版本。

有一个工程，只有一个 PBL，另外附加两个 PBD(引用其中的函数和对象),但是编译时引起 IDE 的崩溃，而如果在 PBL 上 build runtime library 则可以顺序编译成功，如何处理？

这样处理，既然 PBL 可以成功编译成 PBD, 那我们只需要生成一个可以加载三个 PBD 的 exe 即可。在桌面新建任意 workspace，名为: abc，新建一个 PBL，同上述 PBL 的名字相同。然后再添加两个 PBL，名字同上述中的“附加的 PBD 名”一致。然后新建一个编译工程，编译得到一个 EXE。将你这个 EXE 拷贝后放入上述的功能目录下即可顺利运行。

Recoverydw 在导出一个 dw 时发现，一个 pbd 中的 dw，不同时间去操作，结果导出两个不同的 srd 来。

版本 12.5
现象：在不同时候打开 pbd 导出 srd 时，得到的 dw 的语法都正确，但是却是不同的。根据 dw 中用到的 table 和字段分析，可能是这个 dw 的历史版本编译成的二进制仍然存在。不过我想这是属于 PB 的 BUG。出现这个问题最终都会导致逻辑错误，或者字段不存在的提示等等。

如何去掉导出的代码中的行号，逻辑块的 ID 号？

在菜单》选项》源码导出选项 中，根据需要去掉勾选即可。

当强制中断一个项目的编译后，或者编译导致 PB 崩溃后再次打开该项目会导致 PB 崩溃，如何处理？

请将其中一个 PBL 改名(不含有 application 的)，比如 Abc.pbl 改为 abc.pbl.x，然后加载该工程即可顺序加载。加载后，再将 abc.pbl.x 改回 abc.pbl 即可正常操作。
猜想是因为改了其中一个 pbl 导致 PB 会跳过一些检测过程从而避免再次崩溃。
*反编译后首次加载崩溃也可以尝试此法。

Full 编译 PBL 时，或者创建和编译成 Exe 时，报错误：
*****.pbl(d_***.dwo) failed. Probable library file I/O error**
并且错误提示的数量很多，为什么？
环境 PB7.0 Build6012(7.0.1.6012)，被反编译的文件由 pb7.0.3.10077 创建
在 PB7.0 Build6012 中，dw 能双击被打开，界面元素，检索条件，预览等均正常。但是如

果对其进行 regenerate 的话，会报错误：datawindow syntax has incorrect release number。
我尝试如下方法：

1. 从 recoverydw 导出的语法中，重新 import 此 srd。问题解决
2. 我双击打开该 dw，不做任何编辑修改，直接按 save 图标，鼠标会呈现漏斗状，然后关闭并重新 regenerate，不会报错。
3. 将第 2 点完成的 dw 导出为 srd，然后和 recoverydw 导出的 dw 语法用 ultraedit 进行对比，发现一些差异：

   3.1

   保存后：

   datawindow(units=0 timer_interval=0 color=16777215 processing=1 **HTMLDW=no** print.documentname=""

   recoverydw 导出的：

   datawindow(units=0 timer_interval=0 color=16777215 processing=1 print.documentname=""

   3.2

   保存后：

   table(column=(type=char(12) update=yes updatewhereclause=yes **key=yes** name=

   recoverydw 导出的：

   table(column=(type=char(12) **key=yes** update=yes updatewhereclause=yes name=

   key = yes 位置有点不同

   3.3

   保存后：

   htmlgen(clientevents="1" clientvalidation="1" clientcomputedfields="1"

   recoverydw 导出的：

   htmlgen(clientComputedFields="1" clientEvents="1" clientFormatting="0" clientScrip

   保存后：

   text(band=header alignment="2" text="工号" border="4" color="16711680" x="

   recoverydw 导出的：

   text(name=pa_id_t band=header font.charset="134"

   综上差异，可见不同的 build 之前还是存在 dw 语法的差异。

   我是用较低版本的 pbvm 来导出 dw 语法的。如果用较高版本的 pbvm 来导出我想情况也是一样存在差异的。

有此可见，用重新 import 此 srd 的方法能有效的解决此问题。

某个函数 w_xxx..ue_xxx 中出现此错误：

Bad number of arguments for function: gf_xxx，实际打开 gf_xxx 为三个参数，而该函数中在调用 gf_xxx 时，只有两个参数？

增量编译 gf_xxx 所在的 PBL 造成的问题，或者是 gf_xxx 增加了参数，它所在的 pbd 得到了 update，而 w_xxx 所在的 pbd 并没有被 update。因为为了减少 update 的文件数量，我们通常只更新少量的 pbd 作为 patch。

错误：Incompatible property iuo_xxx for type interface_abc

错误同上 interface_abc 被 edit，有增删属性或者 instance 变量，但是引用它的代码并未被 full 编译或者未被 update。

*当然这个问题和上一个问题还有种可能是开发模式是多人或者多个开发组协作，他们采用将某些其他组开发好的模组用附加 PBD 到工程中的方式进行引用（调用）和编译。PBD 没

有做到实时的更新到最新状态。当然某些模组也可能出于废弃或者开发未完成的状态。所以不会被最终用户发觉。越是大的项目，出现此类问题越多。

Maximum script size exceeded？

代码容纳的字节数超出限制，你可以去掉行号和一些不必要的元素（不改变代码的前提下）。当然也可以分拆函数或者事件为两个。

原来的程序可能程序员写的行数较多，已经非常接近 Maximum script 的限制。当反编译后，由于增加了一些行号，逻辑块 ID 等辅助文字，就造成了超过限制。

*注意这不是行数过多造成的，是代码区的文字长度有限制。

Error scanning object source entry: w_xxx

由于某种原因，在某个代码解析时造成了代码段内出现不可见字符，从而造成源码最终输出时拼接出现异常（往往代码从那里中断，从而造成源码不可用）。

寻求技术支持。

西班牙语韩文等非英文系统，在中文系统上反编译出现不可显示的乱码？

在控制面板》区域与语言选项》高级》非 Unicode 程序的语言中选择“韩文”，如果未安装朝鲜语，将会提示插入 xp 的光盘。安装后重启电脑，并反编译导出 PBL。然后设置回去，并做 Optimize 和 Compile，以及后续的工作。

反编译后的代码有些 goto 语法，但是某些 goto 是正常的(逻辑上可读而且没有问题)少数有点问题需要调整，如何搜索和逐步排除？

Goto 语句采用的语法为 goto+空格+地址标签,如 goto 01BC

如果确认是正常的，可以改为 goto+tab+01BC, 然后保存，并再次收索关键字”goto”即可找到其他的 goto 位置，而不会找到你认为正常的 goto 语句。

反编译后我调试，出现奇怪的现象，某个 uo 的函数无法获取返回值，另外 dw 的 buttonclick 事件无法触发

作者遇到的案例是采用了 PB12.5 2511 做 debug 调试，遇到这样的问题。

如：int li_rtn

```
Li_rtn = of_xxx()       //of_xxx 返回  -1
Li_rtn 始终等于 0.
```

我更新 Patch 到 3072 遂解决。如此推演，其他的 PB 版本，其过渡的中间发行补丁，不一定非常准确无误，如果 debug 状态下出现此类怪异现象，请升级你的补丁。

PB12.5 2511 某段代码中写 parent.postevent(“ue_xxx”)可以触发，里面有代码：
this.setcolumn(“colname_xxx”)
但是 ue_xxx 中有代码：
ls_colname = dw_select.getcolumnname() 则取不到栏位名.

初步怀疑是 tab order 的问题。查看触发事件的图片 p_1 的 tab order=0. 这没问题！

后用 pb12.5 3072 重新导入该 dw，问题消失。怀疑仍然是 2511 补丁的问题，导致点击 p_1 时，p_1 会获得焦点从而导致 colname_xxx 这个栏位失去焦点。

导出 10 个 pbl，其他 pbd 做 optimize 都正常，唯独其中一个做 optimize 时会崩溃。而其中的多数对象当 source edit 时敲入空格再保存都正确，少数几个 source edit 时要崩溃。

通过反复的实践，总结一个经验：

将出错的 pbl，如 abc.pbd 复制一份名为 abc_2.pbl.然后从反编译中导出这个 pbd 的 sr.

将 library list 中添加上 abc_2.pbl.然后在 abc.pbl 上点右键，import 所有的 sr*文件，然后再对其做 optimzie.成功。

Private or protected function cannot be accessed: getcollectiontime
对于某些系统对象，它常用私有的函数来实现属性的值的设置和读取。
或者换种说法是属性 proterty 本质是对应两个函数 getxxx 和 setxxx。
例如这里的 TraceTreeRoutine routine
Routine.Getcollectiontime()改为：routine.collectiontime

个别 dw 导出后为 0 字节
我编译 recoverydw 中的工具时，使用的 PB 版本和你本机安装的 PB 的版本不一致(小版本差异，升级补丁差异)。
请用你本机的 PB 重新编译后再执行导出.

Optimize 正常，full 编译第一个 pbl 时，编译日志到某个 pbl 的某个对象时，IDE 崩溃
根据日志提示，确认出错的 pbl。将原始的 pbd 拷贝过来，然后 library list 中将 pbd 附加上来。而将 pbl 从 library list 中删除。再次编译。先处理其他的语法错误，然后再将出错的 pbl 附加上来，并用反编译导出的 sr 文件导入。并发现错误。

有函数 function_a(int a,int b), 有调用的地方写 function_a(1,2)却总是提示：
Bad number of arguments for function
全局搜索这个函数名发现在另外一个 PBL 存在一个新版的函数 function_a(int a,int b,int c) (同名, 修改了功能并作为补丁包，原函数并未删除之)。

在 PB 高版本 PBNI 可以采用 pbx 文件进行 import PB extension 来将 pbx 文件内部的功能映射成 uo 放入到某个 PBL 中。但是记住，任何改变 PBL 内容的动作（import，编译，增加对象，修改对象，import PB extension 等等）都需要在 Optimize PBL 这个动作之后。否则可能将 PBL 破坏掉，某些文件受损。实践过程中在 Optimize 之前导入 pbx 结果造成另外一个对象的源码只剩下一半。

PB6.5 编译时出现 Link Errors 窗口但是窗口中是空白的，没有任何错误
被编译工程的 exe 或者其他文件被打开时会这样。

某个 dw 导出语法时，能看到界面是一个 form 样式，但是报：内存不能为 read 错误，继而 recoveryw 崩溃。
首先修改 recoverydw 的源码，将 lb_1.selectionchanged 的代码中两行 insertrow 注释掉即可导出：

```
dw_1.setredraw(false)
dw_1.dataobject = this.text(index)
is_currentdw = dw_1.dataobject
//dw_1.insertrow(0)
//dw_1.insertrow(0)
//some form style dw,the foreground and background,color is white all.be seen nothing.
dw_1.setfocus()
```

dw_1.setrow(1)
dw_1.setcolumn(1)
dw_1.setredraw(true)

稍加研究把导出的 srd 文件导入一个新项目中，设置为测试窗体上某个 dw 的 object name。报错。仍然报：内存不能为 read 并引起 IDE 崩溃。

与一个正常的 form 样式的 dw 对比发现如下错误根源
错误的 DW 如此：
table(retrieve=" SELECT a_menu_resource.menuid
    FROM a_menu_resource
where 1=1
" update="a_menu_resource" updatewhere=1 updatekeyinplace=no)

正常的 DW 如此：
table(column=(type=char(6) update=yes updatewhereclause=yes key=yes name=menuid dbname="a_menu_resource.menuid"/*之所以出错，就是缺少这里*/)
 retrieve=" SELECT a_menu_resource.menuid
    FROM a_menu_resource
where 1=1
" update="a_menu_resource" updatewhere=1 updatekeyinplace=no)
进行修补或者打开错误的 dw，照着样式重新设计制作。

用 PB 打开对象时，反编译导出动作还未完成，或者完成瞬间就在 PB 中打开了，PB 打开后提示缺少对象或者函数
注意：recoverydw 工具导出 dw 时，pb 反编译导出 pbl 时，均需要在导出完成提示后约 10 秒后再行操作。对于文件较多的比如 50-100 个，其写 PBL 的时间也较长，要耐心等待文件写完并正常关闭才可以在 PB 中操作。也不要同时用两个 recoverydw 进程同时处理 dw，不要用两个反编译的进程加载相同的项目。不能急于这几秒钟。由此造成任何困扰完全没有必要。

提示[不能找到对象的祖先,在你选择的文件列表中]
CurrentObject: u_cst_tab
AncestorObject: u_canvas.udo
任何继承性的对象，都需要祖先对象才能分析和生成源码(对象的源头：所属类型，实例变量，函数，事件等来源于祖先，在祖先的基础之上增加自己的特性，才形成完整的继承性对象)。如果是多层继承也是同样需要各级祖先对象。同样对于 PBNI 接口创建的对象例如 u_canvas.udo，也是必须的。

一种 IDE 在 full 编译时崩溃的处置方法
Full 编译到某一个对象(注意看底部的编译日志进度!) 时弹出提示内存出错崩溃。
这是属于 IDE 没有进行捕获的错误(类似于我们遇到重大错误而没有 try 会导致崩溃一个道理)但是可以肯定是当前的最后一个对象出了问题。
打开反编译,将“选项》源码输出选项》将程序代码作为注释”打勾重新加载项目，并导出

当前出错的对象的 sr 源码。然后在 PB 中导入。然后打开 source edit 并逐个函数取消对它的注释(下图红色部分)点保存，注意并找出问题所在加以修正。

```
public    function boolean wf_xxx (int a);boolean CHEN_RTN_VAR
return CHEN_RTN_VAR
/*
//Code has been commented, for use,please remove the sentence and remove the comment symbol

/*0001*/  boolean lbl_return
/*     2*/  long ll_i
/*     2*/
/*     3*/  for ll_i = 1 to ll_count
                …
/*0030*/  next
/*     1*/
/*     2*/  return true
*/
end function
```

关于一套程序有多个 EXE，反编译出来后发觉他们有共用 PBL 的情况，该如何合并为一个 workspace，多个 target 的正常开发模式?

一般出现这个情况都是先写了一套主程序,然后需要做几个辅助程序时，用到了主项目中的一些对象或者函数。就会这样引用主项目的 pbl.

pb 编译时有个特点,如果不写 pbr 的情况下,每个 target 编译时,只会在 exe 中的对象清单中列出该 target 使用到的对象. 所以反编译辅助项目时,不能得到完整的被引用项目中的对象.

a.exe 反编译后的文件: a.pbl,a_1.pbl,a_2.pbl
b.exe 反编译后的文件: b.pbl,a_1.pbl(引用第一个项目的）
先把 a 项目完全反编译放入文件夹 a,因为它是主项目文件，pbl 中对象最完整.
再把 b 项目完整反编译放入文件夹 b.

然后把 a.pbl,a_1.pbl,a_2.pbl,b.pbl 复制放入同一个文件夹中 aaa 中。记得 b 项目反编译时的 pbl 除 b.pbl 外都不要（因为 b 项目反编译时出来的 a_1.pbl 对 b 项目是完整的对象，但是对 a 项目来说对象不完整)

然后 workspace 打开是 a 项目。再新添加一个项目(target)PBL 选择已经存在的 b.pbl，然后在 b 项目的 librarylist 添加 a_1.pbl 进去。

当在 Source edit 中或者 import sr 时出现这个错误

C0176: Badly ordered TYPE and VARIABLE declarations. Is this modified exported source?

原因：出现了对象的首尾相接的循环引用。

解决方法：import 时先排除掉这些特殊的对象不进行导入(下面例子中不导入 w_child)。在

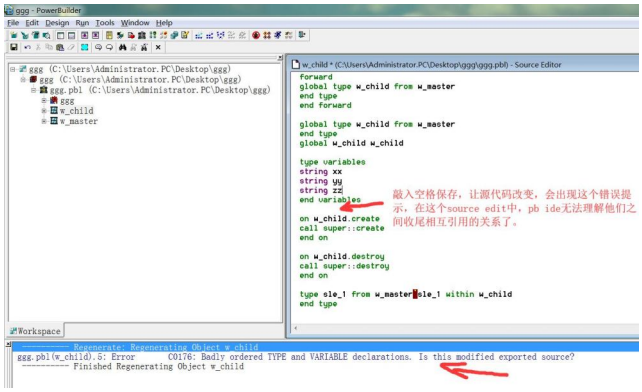
其他对象都全部导入完成后。最后再导入这些特殊的对象。如果造成对象损坏可以进行第一个 pbl 的 full 编译后，会提示缺少某个对象时，import 就能解决。而且这种对象不要去 source edit 它。
具体解释：
当出现继承关系，比如　祖先：w_master　　子孙：w_child　继承自 w_master
当 w_child 写好后（已经存在了）
在 w_master 中就可以首尾循环来引用 w_child, 比如写上 instance 变量：　w_child iw_child
问题来了，当你用 edit source 改变 w_child 时，或者用 PB Decompiler 反编译后再次 import 源码时就会报错：C0176: Badly ordered TYPE and VARIABLE declarations. Is this modified exported source?　而且你做 optimize 时，会提示你导出源码并会把这个对象从 pbl 中移除。这时如果关闭 pb 并重新打开 w_master，会崩溃，因为 w_master 缺少 w_child 这个对象的引用。此时只能在 w_master 的 source edit 中先屏蔽 w_child iw_child 的申明

我理解的：PB 提示的意思在于：你继承一个祖先对象时，必须干干净净的继承它，然后再申明变量（要在 IDE 中操作），而在 source edit 中修改或者 import sr 时，这些首尾循环引用的对象继承关系，pb 无法认同。它无法理解和处理。

具体说：在 ide 中，像 w_child 使用了 w_master，而 w_master 中又使用了 w_child。当你打开其中一个，再打开另外一个就提示你无法打开。只能关闭另外一个才能编辑。因为在编辑 w_master 时，pb 需要知道 w_child 的所有静止状态信息。同理，编辑 w_child 时，也需要完全知道 w_master 的静止状态信息。不允许同时打开修改。但是 pb 却可以允许这种循环的引用。

回到刚才的问题。pb 提示的错误位置（如 ggg.pbl(w_child).5.error），其实是在实例变量申明的位置。也就是
type variables
。 。 。 。
end variables　　《《错误行号提示的其实是这里。

尝试把子孙对象的变量申明剪切掉，可以正常保存。然后回到 ide 中，重新粘贴到 instance var 处是可以正常保存的。



某个 dw，当添加多个 PBD 和一个 PBD 时，单击 dw 名时，看到的右边预览图不一样？
这是因为添加一个 pbd 时，一定是显示这个 pbd 中的这个 dw 的预览图。
而一次添加多个 pbd 时，当其他 pbd 中存在同名 dw 时，会按 libraylist 顺序查找，有可能显

示的是另外 pbd 中的预览图。
所以：当你处理的 pbd 不是同一个项目时，请不要使用 Recoverydw_Vxxx.exe 界面的“打开其他文件”按钮。避免添加的 pbd 与前面添加的 pbd 混为一起。因为我们同一个项目中基本会在开发时避免同名的 dw。但是不同项目之间很容易发生这个问题。特别是相近似的项目，或者同一个项目的不同版本之间，同名率非常高。

***还有个奇特的现象，在某些版本中，可能是存在一些 bug。造成 pbd 中某个 dw 被删除了。但是好像二进制还存在。从而造成当 librarylist 顺序不同时，读取的 dw 预览却完全不同。令人费解。所以这个单独加载它存在的 pbd 来导出。

一个项目，在编译第一个 PBL 时用 FULL 编译模式会造成 IDE 退出，而用 Incremental 编译时可以通过，之后执行 FULL 编译时，也可以通过？
对，经过多次试验证明了这一点，当 IDE 崩溃时可以提前采用这个办法。这是一个解决问题的方法。

Chen
2008 开始
2019-10-29 最后更新