SourceBug
Release Notes

SourceBug 1.1.1

This version of SourceBug is primarily a performance release. Most areas of SourceBug have been speeded up.

This version of SourceBug requires System 7 or later .

Improvements in v. 1.1.1

- The most noticeable improvement is targeting speed. You should observe a reduction in the time required to target an application by a factor ranging from 2 to 6.
- Display of variables and most other simple operations have been speeded up.
- SourceBug's default partition size has been reduced to 2 MB. SourceBug can now use MultiFinder memory for most of its needs. For smaller targets you can reduce the partition size ever further. For very large programs, you may have to manually increase the partition size, because there are a few areas in SourceBug which do not recover well after running out of memory.
- The initial memory requirement for targeting an application generated by the incremental linker has been reduced by about 2MB. About 1 MB of space is allocated later for computing scopes in MacApp.Lib if you try to evaluate a variable or set a break point in any of the MacApp functions. There is a delay of a few seconds while the scopes in the MacApp.Lib are parsed in. You can view `'this'` in MacApp functions without incurring this expense.

New Features in v. 1.1.1

- A much requested feature, “List Local Variables” has been added in the Inspect menu. There is key selection behavior in the list pane; you can type the name of the variable or use the up or down arrow keys. Help balloons have been added for this feature. The position of this window is remembered.
- A new menu item, “Set Break in Class…” has been added under the Control menu. Selecting this menu item sets a break point at the beginning of every member function of the class. Actually, the break points are set after the LINK instruction at the beginning of most function bodies, so that local variables and parameters are valid when the break point is hit. This menu is active only when the class list pane is the active pane and a class is selected in the pane.
- Key selection behavior has been added in the inspector panes.

Known Outstanding Bugs

- SourceBug sometimes crashes when stepping across a signal.
- SourceBug fails to execute properly when placing a breakpoint on a `WITH` statement in a Pascal program. When specifying `-opt on` (the default), this may result in a bus error.
- SourceBug cannot evaluate any variable in Common in a Fortran program.
- Patterns are not displayed correctly. SourceBug has trouble displaying byte-length values, par - ticularly chars or booleans defined in a packed record.
- There is a limit on the size of source code that SourceBug can deal with.
- Floating point variables are evaluated incorrectly when single-stepping over the statement that contains the assignment.
- SourceBug does not show arrays of records.
- SourceBug is unable to view `Str255` parameters passed to procedures.
- `Evaluate` works incorrectly on Booleans that are stored in data registers and sometimes incorrectly reports a value of `TRUE`.
- SourceBug ignores MPW’s tab settings in source code files.
- Hiding SourceBug does not work; the target program is run instead. This is a process manager bug and cannot be fixed in SourceBug. A fix to this bug may be included in a future release of the System.

  A workaround for this is option-clicking either the desktop or a window belonging to another application, or option selecting another application from the Application menu.

- If you compile parts of your application with symbols, others without, SourceBug's value display will try to find a definition of derived objects, but will not fall back to using static types if the dynamic type is not available. Actually, in SourceBug 1.1, instead of showing an error message, SourceBug tries a little harder, gets it right some of the time, and shows wrong values at other times.
- There are a few memory leaks in SourceBug This leads to a gradual performance degradation as programs are targeted, released, and retargeted.

The following bugs affect only targets produced by ILink.

- Consts generated by the Pascal compilers are not evaluated properly.
- SourceBug 1.1 modifies the statefile it uses for debugging.
- When using SourceBug with a statefile, all object files must have unique names, even if they are in different directories.

Workarounds

There is an intermittent problem that occurs with MacApp Applications built with debug on when `Step Into` steps into `FailNonObject` rather than the intended routine. A workaround for this is to patch `%_DISCIPLINEDISPATCH` in UPascalObject.a in the MacApp Sources.

Change

```
                  If qModelFarCode then
                        .
                        .
                        .
                        Move.W            4(A1),D1
                  Else
                        .
                        .
                        .
                        Move.W            2(A1),D1
                  Endif
```

to

```
                  If qModelFarCode then
                        .
                        .
                        .
                        Move.W            4(A1),D1
                        Ext.L       D1
                        Move.L            D1,-(sp)
                  Else
                        .
                        .
```

```
      .
      Move.W            2(A1),D1
      Ext.L       D1
      Move.L            D1,-(sp)
Endif
```

and change

```
      _GetHandleSize
      Cmp.W       D1,D0
```

to

```
      _GetHandleSize
      Cmp.L       (sp)+,D0
```

Bug Fixes in v. 1.1.1

- ~~If the user selects a portion of a variable name, the Evaluate menu displays the full variable name. However, if the user selectes the menu command, SourceBug tries to display a variable with the selected partial name~~.
- ~~SourceBug crashes while trying to display a filed in a record which contains an address in ROM. A common example of this is a record of the type TERec~~.
- ~~In SourceBug 1.1 there is no way to view class static variables~~ .

  Now you can pull down the “Evaluate…” menu, and type in a fully qualified name, e.g. `TClass::fgStaticField`. SourceBug will then display the variable.
- ~~In SourceBug 1.1, variables can be displayed even after they have gone out of scope, sometimes giving incorrect results~~.
- ~~SourceBug1.1 does not enable the “Set Breakpoint At Failure” menu item if there are no Pascal object based classes~~.

  For MacApp 3.1, this menu is now available.