Support Procedures for Icon Program Monitors

Ralph E. Griswold

Department of Computer Science, The University of Arizona

As described in [1], the multi-thread implementation of Icon [2] is extensively instrumented to provide information needed by Icon program monitors.

Such monitors often need similar services. The procedures listed below provide such support. These procedures require the initialization provided by EvInit(), which must be called first.

Mappings

The procedure evnames() returns a two-way table [3] that associates event codes with strings that describe the events. For example, if

```
namemap := evnames()
```

then

```
namemap[E_List]
```

produces "list allocation" and

```
namemap["list allocation"]
```

produces the value of E_List.

The procedure evsyms() returns a two-way table that associates event codes with the names of the corresponding global variables. For example, if

```
symmap := evsyms()
```

then

```
symmap[E_List]
```

produces "E_List" and

```
symmap["E_List"]
```

produces the value of E_List.

The procedure typesyms() returns a table that associates both type codes and type code letters as given by typecode() [4] with the event codes associated with the allocation of these types. For example, if

```
typemap := typesyms()
```

then both

```
typemap[T_List]
```

and

```
typemap["L"]
```

produce the value of E_List.

The procedure opnames() returns a table that maps the integer operation codes for Icon's virtual-machine instructions to the string names for the instructions. For example, if

```
opermap := opnames()
```

and an E_Opcode event for procedure invocation occurs,

```
opermap[&eventvalue]
```

produces "Invoke".

Color Bindings

The procedure typebind(window, codes, opts) returns a table of graphic contexts bound to window in which there is a context for each allocation type event code in codes. These contexts have foreground colors corresponding to the type for that event. opts is a table in the style produced by options() [4]. opts["p"] specifies the palette to use and defaults to "standard" [1] if opt is null or if it does not contain an entry for the key "p".

For example,

```
typecolors := typebind(Visualization, TypeMask)
```

produces a table of contexts bound to the window Visualization with the standard color for each type.

The procedure addattrib(T, s1, s2, ...) sets the attributes given by s1, s2, ... to the table of contexts T. For example,

```
addattrib(typecolors, "bg=white")
```

sets the background for all the contexts in typecolors to white.

References

1. R. E. Griswold and C. L. Jeffery, *Writing Execution Monitors for Icon Programs*, The Univ. of Arizona Icon Project Document IPD192, 1994.
2. C. L. Jeffery, *The MT Icon Interpreter*, The Univ. of Arizona Icon Project Document IPD169, 1993.
3. R. E. Griswold and M. T. Griswold, *The Icon Programming Language*, Prentice-Hall, Inc., Englewood Cliffs, NJ, second edition, 1990.
4. R. E. Griswold, *The Icon Program Library; Version 9.0*, The Univ. of Arizona Icon Project Document IPD242, 1994.