hsext

Claude Heiland-Allen

2007–2010

Contents

1 api/PureData.hs

```
{-# OPTIONS_GHC -Onot #-}
-- hsext -- Haskell objects for Pd
-- Copyright (C) 2007 Claude Heiland-Allen
--
--
-- This program is free software; you can redistribute it and/or
-- modify it under the terms of the GNU General Public License
-- as published by the Free Software Foundation; either version 2
-- of the License, or (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, ↙
    ↳ USA.


module PureData (
  Outlet,
  Method,
  Instance(Instance,iInlets,iOutlets,iMethod),
  Creator,
  Symbol,
  Atom(AFloat,ASymbol),
  peekSymbol,
  gensym,
) where

import Pd (
  Symbol,
  Atom(AFloat,ASymbol),
  peekSymbol,
  gensym)

import Foreign.C.Types (CInt)
import Foreign.Ptr (Ptr)

type Outlet = CInt -> Ptr Symbol -> [Atom] -> IO ()
type Method = Outlet -> CInt -> Ptr Symbol -> [Atom] -> IO ()
data Instance = Instance {
  iInlets  :: CInt,
  iOutlets :: CInt,
  iMethod  :: Method}
type Creator = [Atom] -> IO (Maybe Instance)
```

2 COPYING

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you

distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.

However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding

those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively

```
290 convey the exclusion of warranty; and each file should have at least
    the "copyright" line and a pointer to where the full notice is found.

        <one line to give the program's name and a brief idea of what it does.>
        Copyright (C) <year>  <name of author>
295
        This program is free software; you can redistribute it and/or modify
        it under the terms of the GNU General Public License as published by
        the Free Software Foundation; either version 2 of the License, or
        (at your option) any later version.
300
        This program is distributed in the hope that it will be useful,
        but WITHOUT ANY WARRANTY; without even the implied warranty of
        MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
        GNU General Public License for more details.
305
        You should have received a copy of the GNU General Public License
        along with this program; if not, write to the Free Software
        Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA

310
    Also add information on how to contact you by electronic and paper mail.

    If the program is interactive, make it output a short notice like this
    when it starts in an interactive mode:
315
        Gnomovision version 69, Copyright (C) year name of author
        Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
        This is free software, and you are welcome to redistribute it
        under certain conditions; type `show c' for details.
320
    The hypothetical commands `show w' and `show c' should show the appropriate
    parts of the General Public License.  Of course, the commands you use may
    be called something other than `show w' and `show c'; they could even be
    mouse-clicks or menu items--whatever suits your program.
325
    You should also get your employer (if you work as a programmer) or your
    school, if any, to sign a "copyright disclaimer" for the program, if
    necessary.  Here is a sample; alter the names:

330   Yoyodyne, Inc., hereby disclaims all copyright interest in the program
      `Gnomovision' (which makes passes at compilers) written by James Hacker.

      <signature of Ty Coon>, 1 April 1989
      Ty Coon, President of Vice
335
    This General Public License does not permit incorporating your program into
    proprietary programs.  If your program is a subroutine library, you may
    consider it more useful to permit linking proprietary applications with the
    library.  If this is what you want to do, use the GNU Library General
340 Public License instead of this License.
```

3 doc/create.dot

```
digraph G {
        graph [bgcolor=white,rankdir=TB];
        edge [color=black];
```

```
        subgraph pdMain {
                node [shape=box];
                pdCreate [label="pd"];
                pdNew [label="pd"];
        }
        subgraph hsextHsExt {
                node [shape="ellipse"];
                hsCreate [label="HsExt.hs"];
        }
        subgraph hsextlibhsext {
                node [shape="parallelogram"];
                hslibCreate [label="libhsext.c"];
        }
        subgraph hsPlugin {
                node [shape="ellipse"];
                hsCreator [label="UserCode.hs"];
        }
        subgraph create {
                pdCreate -> hsCreate [label="newMethod (wrapped)"];
                hsCreate -> hsCreator [label="(1) creator"];
                hsCreate -> hslibCreate [label="(2) libhsext_new"];
                hslibCreate -> pdNew [label="pd_new"];
        }
}
```

4 doc/dot4latex

```
#!/bin/sh
# @(#) creates scalable eps and pdf from dot-files
# @(#) by Sebastian Dietzold
# @(#) see http://sebastian.dietzold.de/blog/2005/02/16/eps4latex
# @(#) usage: dot4latex file1.dot file2.dot ...

if [ "$1" = "" ]
then
        what `which $0` 2>/dev/null
        exit 1
fi

for i in $*
do
        file=$i
        name=`echo $file | cut -d "." -f 1`
        dot -Tps $file -o $name.ps && cat $name.ps | sed "s/PS-Adobe-2.0/PS-↙
            ↳ Adobe-2.0 EPSF-1.2/1" >$name.eps && rm $name.ps && epstopdf $name.↙
            ↳ eps -o=$name.pdf && echo $i
done
```

5 doc/hsext-abstract.tex

```
\documentclass{NIME-alternate}

\begin{document}

\title{hsext}
\subtitle{[Extended Abstract]}
```

```
\date{15 April 2007}

\numberofauthors{1}

\author{
\alignauthor Claude Heiland-Allen\\
\affaddr{GOTO10}\\
\email{claudiusmaximus@goto10.org}
}

\maketitle

\begin{abstract}

hsext is a PureData external library for loading and using within PureData code
written in the Haskell\footnote{http://www.haskell.org} programming language.
Haskell is a non-strict, purely-functional programming language, with
polymorphic typing, lazy evaluation and higher-order functions. This paper aims
to introduce and discuss the motivation, implementation and use of hsext, and
also present a roadmap for future development.

Development of hsext started in January 2007, and hsext-0.1 was tagged on 14th
March 2007. hsext-0.1 is sufficient to write simple externals (for example, a
generic ``Swap'' object that can handle more than single floats), and is stable
and efficient enough to be used in live performance (for example, the author's
``$d_0 \rightarrow d_1 \rightarrow d_2 \rightarrow d_3 \rightarrow d_4$'').

Motivation: C, Python, Java, and so on, are all imperative languages, wherein a
program consists of a sequence of commands describing how to compute something.
Haskell is a pure functional language, wherein a program consists of a single
expression describing what needs to be computed. This focus on the high-level
``what'' rather than the low-level ``how'' is a distinguishing characteristic of
functional programming languages, and allows a good compiler to make seemingly
astonishing optimizations.

Implementation: Haskell has a Run Time System (RTS) which must be started before
any Haskell code can be used. This requires a small amount of C code in
hsext\_setup() to start the RTS and call into the Haskell code. Haskell has a
Foreign Function Interface (FFI) which allows Haskell to call C functions, and
also allows Haskell functions to be ``wrapped'' into C function pointers that
are able to be called from C code. Haskell has a package for dynamically
compiling and loading Haskell code at runtime (Plugins), which hsext uses to
instantiate Pd objects defined by user-created Haskell code.

\begin{figure}[t]
\begin{verbatim}
module Swap where
import PureData
import Data.IORef

creator :: Creator
creator args = do
  bang <- gensym "bang"
  state <- newIORef (bang, [])
  return (Just Instance{
    iInlets  = 2,
    iOutlets = 2,
```

```
    iMethod  = inlet state})

inlet state outlet 0 s m = do
  (rs,rm) <- readIORef state
  outlet 1 s m
  outlet 0 rs rm
  return ()

inlet state outlet 1 s m = do
  writeIORef state (s,m)
  return ()
\end{verbatim}
\caption{Example: a generic ``Swap'' object in hsext-0.1.}
\end{figure}

The next level: the PureData interface defined by hsext-0.1 is deficient, for
several reasons, chief among them being the distance between Pd's way of doing
things and how hsext presents that to the user-created Haskell code. These
deficiencies will be rectified in hsext-1.0.

Pipe dreams: while hsext-0.1 is sufficient for simple tasks, and hsext-1.0 will
make the experience more user-friendly and complete, there remains the issue of
having to marshall all data sent between objects into Pd's messages - lists of
symbols and floats. hsext-2.0 will lift this marshalling responsibility from
the user-created Haskell code, allowing arbitrary Haskell values to be passed
between hsext objects. This can be implemented using the Haskell libraries
Data.Typeable and Data.Dynamic.

\end{abstract}

\keywords{PureData, external, Haskell}

\end{document}
```

6 doc/hsext.tex

```
\documentclass{NIME-alternate}

\usepackage{graphicx}

\begin{document}

\title{hsext}
\subtitle{[Draft Version]}
\date{27 March 2007}

\numberofauthors{1}

\author{
\alignauthor Claude Heiland-Allen\\
\affaddr{GOTO10}\\
\email{claudiusmaximus@goto10.org}
}

\maketitle

\begin{abstract}
```

```
hsext is a PureData external library for loading and using within PureData code
written in the Haskell\footnote{http://www.haskell.org} programming language.

Haskell is a non-strict, purely-functional programming language, with
polymorphic typing, lazy evaluation and higher-order functions.

This paper aims to introduce and discuss the motivation, implementation and use
of hsext, and also present a roadmap for future development.

\end{abstract}

\keywords{PureData, external, Haskell}


\section{Introduction}

The birth of hsext can be traced to late 2006, when the author returned to
Haskell after several years' abscence. Development of hsext started in January
2007, and hsext-0.1 was tagged on 14th March 2007.

hsext-0.1 is sufficient to write simple externals, such as a generic ``Swap''
external that can swap arbitrary messages (Pd's internal ``swap'' is specific to
``float'' messages).


\section{Motivation: Why Haskell?}

C, Python, Java, and so on, are all imperative languages, wherein a program
consists of a sequence of commands describing how to compute something. Haskell
is a pure functional language, wherein a program consists of a single expression
describing what needs to be computed. This focus on the high-level ``what''
rather than the low-level ``how'' is a distinguishing characteristic of ↙
    ↳ functional
programming languages.

The ``purity'' of Haskell -- the explicit separation of
execution of ``actions'' (which may have side effects) and evaluation of
expressions (which may not) -- allows a good compiler to make seemingly
astonishing optimizations\footnote{ClaudiusMaximus: Compiling with -O2 reduced
the time taken by my program's execution from 28 mins to 17 seconds. -- Haskell
Weekly News: January 31, 2007: Quotes of the Week
http://sequence.complete.org/hwn/20070131}.

Haskell is non-strict, using lazy evaluation by default. This allows infinite
data structures to be defined, and evaluated just as much as is necessary (but
no more). The classic example is the simple definition of the Fibonacci
sequence\footnote{0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... where each successive
number is the sum of the previous two numbers
http://www.research.att.com/~njas/sequences/A000045} by:

\begin{verbatim}
fibs = 0 : 1 : zipWith (+) fibs (tail fibs)
\end{verbatim}


\section{Implementation: Weaving from C to Haskell and back again}
```

```
Haskell has a Run Time System (RTS) which must be started before any Haskell
code can be used. When compiling standalone programs this is handled
automatically by the Haskell compiler, but for shared libraries such as Pd
externals there is not such compiler support (as of ghc-6.6, the current stable
release of the most prevalent Haskell compiler). This prevents hsext being
written solely in Haskell, and so there is a small amount of C code that starts
the RTS and calls the main hsext entry point (written in Haskell) (Figure
\ref{F:setup}).

\begin{figure}
  \centering
  \includegraphics{setup}
  \caption{Control flow when loading hsext.}
  \label{F:setup}
\end{figure}

Haskell has a Foreign Function Interface (FFI) which allows Haskell to call C
functions, and also allows Haskell functions to be ``wrapped'' into C function
pointers that are able to be called from C code. The tool hsc2hs helps write
Haskell to manipulate C values (including defines and structs), but the author
found it more comfortable to write small helper functions (libhsext) in C to
abstract some of the more idiosyncratic aspects of Pd's API from the Haskell
part of hsext.

Haskell has a package for dynamically compiling and loading Haskell code at
runtime (Plugins). When an hsext object is instantiated in Pd, the first
creation argument is used to form the name of a Haskell source code file to
compile and load. This Haskell source code defines a ``resource'', which must ↙
    ↳ be
of type ``Creator'' (which is defined by the ``PureData'' API provided by hsext)↙
    ↳ ,
and the remaining creation arguments are passed to it. The creator resource in
hsext-0.1 can reject the arguments, or it may return an ``Instance'' specifying
the number of inlets and outlets, and a single function, which will handle all
messages arriving at the inlets (Figure \ref{F:create}).

\begin{figure}
  \centering
  \includegraphics{create}
  \caption{Control flow when creating a hsext object.}
  \label{F:create}
\end{figure}

Pd's API has several functions to create inlets and add methods, but in
hsext-0.1 an object has but one method (a design decision that later turned out
to be a mistake). If the Creator returns an Instance successfully, hsext calls
out to libhsext to create a new Pd object with the required number of proxy
inlets (a technique used in Pd's ``list'' family of objects) and the required
number of outlets, then links the Pd object with the Instance and returns the
Pd object to the caller (the main Pd program).

When the Pd object receives a message at one of its proxy inlets, Pd calls a
function in libhsext. This function looks into the inlet structure to find the
owning Instance ID, and passes all this information into the Haskell core of
hsext. The core calls the one method defined in the Instance with these
arguments, and in addition passes in a structure exposing the PureData API. In
```

```
hsext-0.1 this last is a function that can be used to send messages to the Pd
object's outlets (and thus back into the C environment of the Pd main program)
(Figure \ref{F:inlet}).

\begin{figure}
  \centering
  \includegraphics{inlet}
  \caption{Control flow when data arrives at an hsext inlet.}
  \label{F:inlet}
\end{figure}

\section{Using hsext: a generic Swap object}

The simplest not-quite-trivial example showing most of the features of hsext-0.1
follows. We want to build a Pd object that behaves like Pd's internal swap, but
that can operate on arbitrary messages, not just float messages. We are going
to call it Swap, store it in a file called Swap.hs, and instantiate it in Pd
like [hsext Swap].

First, we need a module declaration:

\begin{verbatim}
module Swap where
\end{verbatim}
Next, we need to import the PureData API:
\begin{verbatim}
import PureData
\end{verbatim}
We need to preserve state between calls from Pd, and a simple way to do this is
using IORefs:
\begin{verbatim}
import Data.IORef
\end{verbatim}

Now we can define our creator resource, which creates a new initial state and
returns an Instance. Notice how the state variable reference is embedded in the
returned method:
\begin{verbatim}
creator :: Creator
creator args = do
  bang <- gensym "bang"
  state <- newIORef (bang, [])
  return (Just Instance{
    iInlets  = 2,
    iOutlets = 2,
    iMethod  = inlet state})
\end{verbatim}

The left inlet recalls the stored state (a message), sends the incoming message
to the right outlet, and sends the recalled message to the left outlet:
\begin{verbatim}
inlet state outlet 0 s m = do
  (rs,rm) <- readIORef state
  outlet 1 s m
  outlet 0 rs rm
  return ()
\end{verbatim}
```

```
190 The right inlet simply stores the incoming message:
    \begin{verbatim}
    inlet state outlet 1 s m = do
      writeIORef state (s,m)
      return ()
195 \end{verbatim}


    \section{The next level: towards hsext-1.0}

200 The PureData interface defined by hsext-0.1 is deficient, for several
    reasons:

    \begin{itemize}

205 \item The ``creator resource'' implies that a Pd object written in Haskell
    cannot have a state shared between its instances \footnote{Haskell provides
    functions such as unsafePerformIO which might be used to work around this, but
    these are fraught with danger}.

210 \item The ``creator resource'' implies that each Haskell source file can define
    only one Pd object class.

    \item The ``one method for all inlets'' paradigm is mismatched to Pd's way of
    doing things, making it awkward to have different behaviour for different
215 selectors.

    \item Passing the ``outlet function'' to the inlet method is non-extensible.

    \item An insufficient amount of Pd's functionality is accessible from Haskell.
220
    \end{itemize}

    These deficiencies could be rectified in the following ways:

225 \begin{itemize}

    \item Replace the ``creator resource'' with a ``setup resource'', which hsext
    will guarantee is executed exactly once (thus allowing state unique to each
    source file).
230
    \item Map to Pd's API more directly from hsext's PureData interface. In
    particular, wrap access to Pd object class creation, and introduce a ``one
    method for each (inlet, selector) pair'' paradigm (while still allowing
    ``anything'' methods for all inlets).
235
    \item Pass a Haskell record containing a collection of possible API calls to
    each method, ensuring future extensibility without requiring hsext users to
    modify their code.

240 \item Wrap more of Pd's API, including:
    \begin{itemize}
      \item Clock callbacks.
      \item Receive names.
      \item Table access.
245   \item Pointer atoms.
    \end{itemize}
```

```
\end{itemize}

\section{Pipe dreams: sending Haskell data through Pd patch-cords}

TODO FIXME

\end{document}
```

7 doc/inlet.dot

```
digraph G {
        graph [bgcolor=white,rankdir=TB];       /* set background */
        edge [color=black];
        subgraph pdMain {
                node [shape=box];
                pdInlet [label="pd"];
                pdOutlet [label="pd"];
        }
        subgraph hsextHsExt {
                node [shape="ellipse"];
                hsInlet [label="HsExt.hs"];
                hsOutlet [label="HsExt.hs"];
        }
        subgraph hsextlibhsext {
                node [shape="parallelogram"];
                hslibInletWrapper [label="libhsext.c"];
                hslibOutletWrapper [label="libhsext.c"];
        }
        subgraph hsPlugin {
                node [shape="ellipse"];
                hsInletMethod [label="UserCode.hs"];
        }
        subgraph inlet {
                pdInlet -> hslibInletWrapper [label="libhsext_inlet"];
                hslibInletWrapper -> hsInlet [label="inletMethod (wrapped)"];
                hsInlet -> hsInletMethod [label="Instance{iMethod}"];
        }
        subgraph outlet {
                hsInletMethod -> hsOutlet [label="outlet (passed)"];
                hsOutlet -> hslibOutletWrapper [label="libhsext_outlet"];
                hslibOutletWrapper -> pdOutlet [label="outlet_anything"];
        }
}
```

8 doc/layers.dot

```
digraph G {
        graph [bgcolor=white,rankdir=TB];       /* set background */
        edge [color=black];
        subgraph pdMain {
                node [shape=box];
                pdLoad [label="pd"];
                pdCreate [label="pd"];
                pdInlet [label="pd"];
                pdOutlet [label="pd"];
```

```
                pdNew [label="pd"];
        }
        subgraph hsextbootstrap {
                node [shape="diamond"];
                hsBootstrap ["bootstrap.c"];
        }
        subgraph hsextHsExt {
                node [shape="ellipse"];
                hsSetup [label="HsExt.hs"];
                hsCreate [label="HsExt.hs"];
                hsInlet [label="HsExt.hs"];
                hsOutlet [label="HsExt.hs"];
        }
        subgraph hsextlibhsext {
                node [shape="parallelogram"];
                hslibSetup [label="libhsext.c"];
                hslibCreate [label="libhsext.c"];
                hslibInletWrapper [label="libhsext.c"];
                hslibOutletWrapper [label="libhsext.c"];
        }
        subgraph hsPlugin {
                node [shape="ellipse"];
                hsCreator [label="UserCode.hs"];
                hsInletMethod [label="UserCode.hs"];
        }
        subgraph load {
                pdLoad -> hsBootstrap [label="hsext_setup"];
                hsBootstrap -> hsSetup [label="hsext_init"];
                hsSetup -> hslibSetup [label="libhsext_setup"];
        }
        subgraph create {
                pdCreate -> hsCreate [label="newMethod (wrapped)"];
                hsCreate -> hsCreator [label="(1) creator"];
                hsCreate -> hslibCreate [label="(2) libhsext_new"];
                hslibCreate -> pdNew [label="pd_new"];
        }
        subgraph inlet {
                pdInlet -> hslibInletWrapper [label="libhsext_inlet"];
                hslibInletWrapper -> hsInlet [label="inletMethod (wrapped)"];
                hsInlet -> hsInletMethod [label="Instance{iMethod}"];
        }
        subgraph outlet {
                hsInletMethod -> hsOutlet [label="outlet (passed)"];
                hsOutlet -> hslibOutletWrapper [label="libhsext_outlet"];
                hslibOutletWrapper -> pdOutlet [label="outlet_anything"];
        }
}
```

9 doc/make-abstract.sh

```
#!/bin/bash
pdflatex hsext-abstract.tex
pdflatex hsext-abstract.tex
```

10 doc/make.sh

```
#!/bin/bash
./dot4latex setup.dot
./dot4latex create.dot
./dot4latex inlet.dot
pdflatex hsext.tex
pdflatex hsext.tex
```

11 doc/NIME-alternate.cls

```
% SIG-ALTERNATE.CLS - VERSION 1.7
% "COMPATIBLE" WITH THE "ACM_PROC_ARTICLE-SP.CLS" V2.7SP
% Gerald Murray October 15th. 2004
%
% ---- Start of 'updates' ----
%
% Allowance made to switch default fonts between those systems using
% METAFONT and those using 'Type 1' or 'Truetype' fonts.
% See LINE NUMBER 252 for details.
% Also provided for enumerated/annotated Corollaries 'surrounded' by
% enumerated Theorems (line 841).
% Gerry November 11th. 1999
%
% Made the Permission Statement / Conference Info / Copyright Info
% 'user definable' in the source .tex file OR automatic if
% not specified.
%
% Georgia fixed bug in sub-sub-section numbering in paragraphs (July 29th. 2002)
% JS/GM fix to vertical spacing before Proofs (July 30th. 2002)
%
% Footnotes inside table cells using \minipage (Oct. 2002)
%
% ---- End of 'updates' ----
%
\def\fileversion{v1.7}          % for ACM's tracking purposes
\def\filedate{October 15, 2004}    % Gerry Murray's tracking data
\def\docdate {Friday 15th. October 2004} % Gerry Murray (with deltas to doc}
\usepackage{epsfig}
\usepackage{amssymb}
\usepackage{amsmath}
\usepackage{amsfonts}
%
% SIG-ALTERNATE DOCUMENT STYLE
% G.K.M. Tobin August-October 1999
%    adapted from ARTICLE document style by Ken Traub, Olin Shivers
%    also using elements of esub2acm.cls
% HEAVILY MODIFIED, SUBSEQUENTLY, BY GERRY MURRAY 2000
% ARTICLE DOCUMENT STYLE -- Released 16 March 1988
%    for LaTeX version 2.09
% Copyright (C) 1988 by Leslie Lamport
%
%
%%% sig-alternate.cls is an 'ALTERNATE' document style for producing
%%% two-column camera-ready pages for ACM conferences.
%%% THIS FILE DOES NOT STRICTLY ADHERE TO THE SIGS (BOARD-ENDORSED)
%%% PROCEEDINGS STYLE. It has been designed to produce a 'tighter'
%%% paper in response to concerns over page budgets.
%%% The main features of this style are:
```

```
%%%
%%% 1)  Two columns.
%%% 2)  Side and top margins of 4.5pc, bottom margin of 6pc, column gutter of
%%%     2pc, hence columns are 20pc wide and 55.5pc tall.  (6pc =3D 1in, approx)
%%% 3)  First page has title information, and an extra 6pc of space at the
%%%     bottom of the first column for the ACM copyright notice.
%%% 4)  Text is 9pt on 10pt baselines; titles (except main) are 9pt bold.
%%%
%%%
%%% There are a few restrictions you must observe:
%%%
%%% 1)  You cannot change the font size; ACM wants you to use 9pt.
%%% 3)  You must start your paper with the \maketitle command.  Prior to the
%%%     \maketitle you must have \title and \author commands.  If you have a
%%%     \date command it will be ignored; no date appears on the paper, since
%%%     the proceedings will have a date on the front cover.
%%% 4)  Marginal paragraphs, tables of contents, lists of figures and tables,
%%%     and page headings are all forbidden.
%%% 5)  The 'figure' environment will produce a figure one column wide; if you
%%%     want one that is two columns wide, use 'figure*'.
%%%
%
%%% Copyright Space:
%%% This style automatically reserves 1" blank space at the bottom of page 1/
%%% column 1.  This space can optionally be filled with some text using the
%%% \toappear{...} command.  If used, this command must be BEFORE the \maketitle
%%% command.  If this command is defined AND [preprint] is on, then the
%%% space is filled with the {...} text (at the bottom); otherwise, it is
%%% blank.  If you use \toappearbox{...} instead of \toappear{...} then a
%%% box will be drawn around the text (if [preprint] is on).
%%%
%%% A typical usage looks like this:
%%%     \toappear{To appear in the Ninth AES Conference on Medievil Lithuanian
%%%               Embalming Technique, June 1991, Alfaretta, Georgia.}
%%% This will be included in the preprint, and left out of the conference
%%% version.
%%%
%%% WARNING:
%%% Some dvi-ps converters heuristically allow chars to drift from their
%%% true positions a few pixels. This may be noticeable with the 9pt sans-serif
%%% bold font used for section headers.
%%% You may turn this hackery off via the -e option:
%%%     dvips -e 0 foo.dvi >foo.ps
%%%
\typeout{Document Class 'sig-alternate' <15th. October '04>.  Modified by G.K.M.↙
    ↪  Tobin/Gerry Murray}
\typeout{Based in part upon document Style 'acmconf' <22 May 89>. Hacked 4/91 by↙
    ↪ }
\typeout{shivers@cs.cmu.edu, 4/93 by theobald@cs.mcgill.ca}
\typeout{Excerpts were taken from (Journal Style) 'esub2acm.cls'.}
\typeout{****** Bugs/comments/suggestions/technicalities to Gerry Murray -- ↙
    ↪ murray@hq.acm.org ******}
\typeout{Questions on the style, SIGS policies, etc. to Adrienne Griscti ↙
    ↪ griscti@acm.org}
\oddsidemargin 4.5pc
\evensidemargin 4.5pc
\advance\oddsidemargin by -1in  % Correct for LaTeX gratuitousness
```

```
    \advance\evensidemargin by -1in % Correct for LaTeX gratuitousness
    \marginparwidth 0pt             % Margin pars are not allowed.
    \marginparsep 11pt              % Horizontal space between outer margin and
105                                 % marginal note

                                    % Top of page:
    \topmargin 4.5pc                % Nominal distance from top of page to top of
                                    % box containing running head.
110 \advance\topmargin by -1in      % Correct for LaTeX gratuitousness
    \headheight 0pt                 % Height of box containing running head.
    \headsep 0pt                    % Space between running head and text.
                                    % Bottom of page:
    \footskip 30pt                  % Distance from baseline of box containing foot
115                                 % to baseline of last line of text.
    \@ifundefined{footheight}{\newdimen\footheight}{}% this is for LaTeX2e
    \footheight 12pt                % Height of box containing running foot.

    %% Must redefine the top margin so there's room for headers and
120 %% page numbers if you are using the preprint option. Footers
    %% are OK as is. Olin.
    \advance\topmargin by -37pt     % Leave 37pt above text for headers
    \headheight 12pt                % Height of box containing running head.
    \headsep 25pt                   % Space between running head and text.
125
    \textheight 666pt       % 9 1/4 column height
    \textwidth 42pc         % Width of text line.
                            % For two-column mode:
    \columnsep 2pc          %    Space between columns
130 \columnseprule 0pt      %    Width of rule between columns.
    \hfuzz 1pt              % Allow some variation in column width, otherwise it's
                            % too hard to typeset in narrow columns.

    \footnotesep 5.6pt      % Height of strut placed at the beginning of every
135                         % footnote =3D height of normal \footnotesize strut,
                            % so no extra space between footnotes.

    \skip\footins 8.1pt plus 4pt minus 2pt  % Space between last line of text and
                                            % top of first footnote.
140 \floatsep 11pt plus 2pt minus 2pt       % Space between adjacent floats moved
                                            % to top or bottom of text page.
    \textfloatsep 18pt plus 2pt minus 4pt   % Space between main text and floats
                                            % at top or bottom of page.
    \intextsep 11pt plus 2pt minus 2pt      % Space between in-text figures and
145                                         % text.
    \@ifundefined{@maxsep}{\newdimen\@maxsep}{}% this is for LaTeX2e
    \@maxsep 18pt                           % The maximum of \floatsep,
                                            % \textfloatsep and \intextsep (minus
                                            % the stretch and shrink).
150 \dblfloatsep 11pt plus 2pt minus 2pt    % Same as \floatsep for double-column
                                            % figures in two-column mode.
    \dbltextfloatsep 18pt plus 2pt minus 4pt% \textfloatsep for double-column
                                            % floats.
    \@ifundefined{@dblmaxsep}{\newdimen\@dblmaxsep}{}% this is for LaTeX2e
155 \@dblmaxsep 18pt                        % The maximum of \dblfloatsep and
                                            % \dbltexfloatsep.
    \@fptop 0pt plus 1fil    % Stretch at top of float page/column. (Must be
                             % 0pt plus ...)
```

```
    \@fpsep 8pt plus 2fil    % Space between floats on float page/column.
160 \@fpbot 0pt plus 1fil    % Stretch at bottom of float page/column. (Must be
                             % 0pt plus ... )
    \@dblfptop 0pt plus 1fil % Stretch at top of float page. (Must be 0pt plus ...)
    \@dblfpsep 8pt plus 2fil % Space between floats on float page.
    \@dblfpbot 0pt plus 1fil % Stretch at bottom of float page. (Must be
165                          % 0pt plus ... )
    \marginparpush 5pt       % Minimum vertical separation between two marginal
                             % notes.

    \parskip 0pt plus 1pt            % Extra vertical space between paragraphs.
170 \parindent 9pt  % GM July 2000 / was 0pt - width of paragraph indentation.
    \partopsep 2pt plus 1pt minus 1pt% Extra vertical space, in addition to
                                     % \parskip and \topsep, added when user
                                     % leaves blank line before environment.

175 \@lowpenalty   51       % Produced by \nopagebreak[1] or \nolinebreak[1]
    \@medpenalty  151       % Produced by \nopagebreak[2] or \nolinebreak[2]
    \@highpenalty 301       % Produced by \nopagebreak[3] or \nolinebreak[3]

    \@beginparpenalty -\@lowpenalty % Before a list or paragraph environment.
180 \@endparpenalty   -\@lowpenalty % After a list or paragraph environment.
    \@itempenalty     -\@lowpenalty % Between list items.

    \@namedef{ds@10pt}{\@latexerr{The `10pt' option is not allowed in the `acmconf'
      document style.}\@eha}
185 \@namedef{ds@11pt}{\@latexerr{The `11pt' option is not allowed in the `acmconf'
      document style.}\@eha}
    \@namedef{ds@12pt}{\@latexerr{The `12pt' option is not allowed in the `acmconf'
      document style.}\@eha}

190 \@options

    \lineskip 2pt           % \lineskip is 1pt for all font sizes.
    \normallineskip 2pt
    \def\baselinestretch{1}
195
    \abovedisplayskip 9pt plus2pt minus4.5pt%
    \belowdisplayskip \abovedisplayskip
    \abovedisplayshortskip  \z@ plus3pt%
    \belowdisplayshortskip  5.4pt plus3pt minus3pt%
200 \let\@listi\@listI     % Setting of \@listi added 9 Jun 87

    \def\small{\@setsize\small{9pt}\viiipt\@viiipt
    \abovedisplayskip 7.6pt plus 3pt minus 4pt%
    \belowdisplayskip \abovedisplayskip
205 \abovedisplayshortskip \z@ plus2pt%
    \belowdisplayshortskip 3.6pt plus2pt minus 2pt
    \def\@listi{\leftmargin\leftmargini %% Added 22 Dec 87
    \topsep 4pt plus 2pt minus 2pt\parsep 2pt plus 1pt minus 1pt
    \itemsep \parsep}}
210
    \def\footnotesize{\@setsize\footnotesize{9pt}\ixpt\@ixpt
    \abovedisplayskip 6.4pt plus 2pt minus 4pt%
    \belowdisplayskip \abovedisplayskip
    \abovedisplayshortskip \z@ plus 1pt%
215 \belowdisplayshortskip 2.7pt plus 1pt minus 2pt
```

```
\def\@listi{\leftmargin\leftmargini %% Added 22 Dec 87
\topsep 3pt plus 1pt minus 1pt\parsep 2pt plus 1pt minus 1pt
\itemsep \parsep}}

\newcount\aucount
\newcount\originalaucount
\newdimen\auwidth
\auwidth=\textwidth
\newdimen\auskip
\newcount\auskipcount
\newdimen\auskip
\global\auskip=1pc
\newdimen\allauboxes
\allauboxes=\auwidth
\newtoks\addauthors
\newcount\addauflag
\global\addauflag=0 %Haven't shown additional authors yet

\newtoks\subtitletext
\gdef\subtitle#1{\subtitletext={#1}}

\gdef\additionalauthors#1{\addauthors={#1}}

\gdef\numberofauthors#1{\global\aucount=#1
\ifnum\aucount>3\global\originalaucount=\aucount \global\aucount=3\fi %g}
\global\auskipcount=\aucount\global\advance\auskipcount by 1
\global\multiply\auskipcount by 2
\global\multiply\auskip by \auskipcount
\global\advance\auwidth by -\auskip
\global\divide\auwidth by \aucount}

% \and was modified to count the number of authors. GKMT 12 Aug 1999
\def\alignauthor{%                    % \begin{tabular}
\end{tabular}%
  \begin{tabular}[t]{p{\auwidth}}\centering}%

%  *** NOTE *** NOTE *** NOTE *** NOTE ***
%  If you have 'font problems' then you may need
%  to change these, e.g. 'arialb' instead of "arialbd".
%  Gerry Murray 11/11/1999
%  *** OR ** comment out block A and activate block B or vice versa.
% ******************************************
%
%  -- Start of block A -- (Type 1 or Truetype fonts)
%\newfont{\secfnt}{timesbd at 12pt} % was timenrb originally - now is timesbd
%\newfont{\secit}{timesbi at 12pt}   %13 Jan 00 gkmt
%\newfont{\subsecfnt}{timesi at 11pt} % was timenrri originally - now is timesi
%\newfont{\subsecit}{timesbi at 11pt} % 13 Jan 00 gkmt -- was times changed to ↙
    ↪ timesbi gm 2/4/2000
%                          % because "normal" is italic, "italic" is Roman
%\newfont{\ttlfnt}{arialbd at 18pt} % was arialb originally - now is arialbd
%\newfont{\ttlit}{arialbi at 18pt}    % 13 Jan 00 gkmt
%\newfont{\subttlfnt}{arial at 14pt} % was arialr originally - now is arial
%\newfont{\subttlit}{ariali at 14pt} % 13 Jan 00 gkmt
%\newfont{\subttlbf}{arialbd at 14pt}  % 13 Jan 00 gkmt
%\newfont{\aufnt}{arial at 12pt} % was arialr originally - now is arial
%\newfont{\auit}{ariali at 12pt} % 13 Jan 00 gkmt
```

```
%\newfont{\affaddr}{arial at 10pt} % was arialr originally - now is arial
%\newfont{\affaddrit}{ariali at 10pt} %13 Jan 00 gkmt
%\newfont{\eaddfnt}{arial at 12pt} % was arialr originally - now is arial
%\newfont{\ixpt}{times at 9pt} % was timenrr originally - now is times
%\newfont{\confname}{timesi at 8pt} % was timenrri - now is timesi
%\newfont{\crnotice}{times at 8pt} % was timenrr originally - now is times
%\newfont{\ninept}{times at 9pt} % was timenrr originally - now is times

% ****************************************
%  -- End of block A --
%
%
% -- Start of block B -- METAFONT
% ++++++++++++++++++++++++++++++++++++++++
% Next (default) block for those using Metafont
% Gerry Murray 11/11/1999
% *** THIS BLOCK FOR THOSE USING METAFONT *****
% ****************************************
\newfont{\secfnt}{ptmb at 12pt}
\newfont{\secit}{ptmbi at 12pt}    %13 Jan 00 gkmt
\newfont{\subsecfnt}{ptmri at 11pt}
\newfont{\subsecit}{ptmbi at 11pt}  % 13 Jan 00 gkmt -- was ptmr changed to ptmbi gm 2/4/2000
                        % because "normal" is italic, "italic" is Roman
\newfont{\ttlfnt}{phvb at 18pt}
\newfont{\ttlit}{phvbo at 18pt}    % GM 2/4/2000
\newfont{\subttlfnt}{phvr at 14pt}
\newfont{\subttlit}{phvro at 14pt} % GM 2/4/2000
\newfont{\subttlbf}{phvb at 14pt}  % 13 Jan 00 gkmt
\newfont{\aufnt}{phvr at 12pt}
\newfont{\auit}{phvro at 12pt}     % GM 2/4/2000
\newfont{\affaddr}{phvr at 10pt}
\newfont{\affaddrit}{phvro at 10pt} % GM 2/4/2000
\newfont{\eaddfnt}{phvr at 12pt}
\newfont{\ixpt}{ptmr at 9pt}
\newfont{\confname}{ptmri at 8pt}
\newfont{\crnotice}{ptmr at 8pt}
\newfont{\ninept}{ptmr at 9pt}
% ++++++++++++++++++++++++++++++++++++++++
% -- End of block B --


\def\email#1{{{\eaddfnt{\vskip 4pt#1}}}}

\def\addauthorsection{\ifnum\originalaucount>3
    \section{Additional Authors}\the\addauthors
  \fi}

\newcount\savesection
\newcount\sectioncntr
\global\sectioncntr=1

\setcounter{secnumdepth}{3}

\def\appendix{\par
\section*{APPENDIX}
\setcounter{section}{0}
```

```
  \setcounter{subsection}{0}
  \def\thesection{\Alph{section}} }

\leftmargini 22.5pt
\leftmarginii 19.8pt    % > \labelsep + width of '(m)'
\leftmarginiii 16.8pt   % > \labelsep + width of 'vii.'
\leftmarginiv 15.3pt    % > \labelsep + width of 'M.'
\leftmarginv 9pt
\leftmarginvi 9pt

\leftmargin\leftmargini
\labelsep 4.5pt
\labelwidth\leftmargini\advance\labelwidth-\labelsep

\def\@listI{\leftmargin\leftmargini \parsep 3.6pt plus 2pt minus 1pt%
\topsep 7.2pt plus 2pt minus 4pt%
\itemsep 3.6pt plus 2pt minus 1pt}

\let\@listi\@listI
\@listi

\def\@listii{\leftmargin\leftmarginii
   \labelwidth\leftmarginii\advance\labelwidth-\labelsep
   \topsep 3.6pt plus 2pt minus 1pt
   \parsep 1.8pt plus 0.9pt minus 0.9pt
   \itemsep \parsep}

\def\@listiii{\leftmargin\leftmarginiii
    \labelwidth\leftmarginiii\advance\labelwidth-\labelsep
    \topsep 1.8pt plus 0.9pt minus 0.9pt
    \parsep \z@ \partopsep 1pt plus 0pt minus 1pt
    \itemsep \topsep}

\def\@listiv{\leftmargin\leftmarginiv
     \labelwidth\leftmarginiv\advance\labelwidth-\labelsep}

\def\@listv{\leftmargin\leftmarginv
     \labelwidth\leftmarginv\advance\labelwidth-\labelsep}

\def\@listvi{\leftmargin\leftmarginvi
     \labelwidth\leftmarginvi\advance\labelwidth-\labelsep}

\def\labelenumi{\theenumi.}
\def\theenumi{\arabic{enumi}}

\def\labelenumii{(\theenumii)}
\def\theenumii{\alph{enumii}}
\def\p@enumii{\theenumi}

\def\labelenumiii{\theenumiii.}
\def\theenumiii{\roman{enumiii}}
\def\p@enumiii{\theenumi(\theenumii)}

\def\labelenumiv{\theenumiv.}
\def\theenumiv{\Alph{enumiv}}
\def\p@enumiv{\p@enumiii\theenumiii}
```

```
\def\labelitemi{$\bullet$}
\def\labelitemii{\bf --}
\def\labelitemiii{$\ast$}
\def\labelitemiv{$\cdot$}

\def\verse{\let\\=\@centercr
  \list{}{\itemsep\z@ \itemindent -1.5em\listparindent \itemindent
          \rightmargin\leftmargin\advance\leftmargin 1.5em}\item[]}
\let\endverse\endlist

\def\quotation{\list{}{\listparindent 1.5em
    \itemindent\listparindent
    \rightmargin\leftmargin \parsep 0pt plus 1pt}\item[]}
\let\endquotation=\endlist

\def\quote{\list{}{\rightmargin\leftmargin}\item[]}
\let\endquote=\endlist

\def\descriptionlabel#1{\hspace\labelsep \bf #1}
\def\description{\list{}{\labelwidth\z@ \itemindent-\leftmargin
       \let\makelabel\descriptionlabel}}

\let\enddescription\endlist

\def\theequation{\arabic{equation}}

\arraycolsep 4.5pt   % Half the space between columns in an array environment.
\tabcolsep 5.4pt    % Half the space between columns in a tabular environment.
\arrayrulewidth .4pt % Width of rules in array and tabular environment.
\doublerulesep 1.8pt % Space between adjacent rules in array or tabular env.

\tabbingsep \labelsep   % Space used by the \' command.  (See LaTeX manual.)

\skip\@mpfootins =\skip\footins

\fboxsep =2.7pt       % Space left between box and text by \fbox and \framebox.
\fboxrule =.4pt       % Width of rules in box made by \fbox and \framebox.

\def\thepart{\Roman{part}} % Roman numeral part numbers.
\def\thesection        {\arabic{section}}
\def\thesubsection     {\thesection.\arabic{subsection}}
%\def\thesubsubsection {\thesubsection.\arabic{subsubsection}} % GM 7/30/2002
%\def\theparagraph      {\thesubsubsection.\arabic{paragraph}}  % GM 7/30/2002
\def\thesubparagraph  {\theparagraph.\arabic{subparagraph}}

\def\@pnumwidth{1.55em}
\def\@tocrmarg {2.55em}
\def\@dotsep{4.5}
\setcounter{tocdepth}{3}

\def\tableofcontents{\@latexerr{\tableofcontents: Tables of contents are not
  allowed in the 'acmconf' document style.}\@eha}

\def\l@part#1#2{\addpenalty{\@secpenalty}
   \addvspace{2.25em plus 1pt}  % space above part line
   \begingroup
   \@tempdima 3em       % width of box holding part number, used by
```

```
      \parindent \z@ \rightskip \@pnumwidth      %% \numberline
      \parfillskip -\@pnumwidth
      {\large \bf        % set line in \large boldface
      \leavevmode        % TeX command to enter horizontal mode.
      #1\hfil \hbox to\@pnumwidth{\hss #2}}\par
      \nobreak           % Never break after part entry
    \endgroup}

\def\l@section#1#2{\addpenalty{\@secpenalty} % good place for page break
    \addvspace{1.0em plus 1pt}   % space above toc entry
    \@tempdima 1.5em             % width of box holding section number
    \begingroup
      \parindent  \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      \bf                        % Boldface.
      \leavevmode                % TeX command to enter horizontal mode.
       \advance\leftskip\@tempdima %% added 5 Feb 88 to conform to
       \hskip -\leftskip          %% 25 Jan 88 change to \numberline
      #1\nobreak\hfil \nobreak\hbox to\@pnumwidth{\hss #2}\par
    \endgroup}


\def\l@subsection{\@dottedtocline{2}{1.5em}{2.3em}}
\def\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
\def\l@paragraph{\@dottedtocline{4}{7.0em}{4.1em}}
\def\l@subparagraph{\@dottedtocline{5}{10em}{5em}}

\def\listoffigures{\@latexerr{\listoffigures: Lists of figures are not
  allowed in the 'acmconf' document style.}\@eha}

\def\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}

\def\listoftables{\@latexerr{\listoftables: Lists of tables are not
  allowed in the 'acmconf' document style.}\@eha}
\let\l@table\l@figure

\def\footnoterule{\kern-3\p@
  \hrule width .4\columnwidth
  \kern 2.6\p@}                % The \hrule has default height of .4pt .
% ------
\long\def\@makefntext#1{\noindent
%\hbox to .5em{\hss$^{\@thefnmark}$}#1}   % original
\hbox to .5em{\hss\textsuperscript{\@thefnmark}}#1}  % C. Clifton / GM Oct. 2nd.↙
    ↳ 2002
% -------

\long\def\@maketntext#1{\noindent
#1}

\long\def\@maketitlenotetext#1#2{\noindent
            \hbox to 1.8em{\hss$^{#1}$}#2}

\setcounter{topnumber}{2}
\def\topfraction{.7}
\setcounter{bottomnumber}{1}
\def\bottomfraction{.3}
\setcounter{totalnumber}{3}
```

```
\def\textfraction{.2}
\def\floatpagefraction{.5}
\setcounter{dbltopnumber}{2}
\def\dbltopfraction{.7}
\def\dblfloatpagefraction{.5}

%
\long\def\@makecaption#1#2{
   \vskip \baselineskip
   \setbox\@tempboxa\hbox{\textbf{#1: #2}}
   \ifdim \wd\@tempboxa >\hsize % IF longer than one line:
       \textbf{#1: #2}\par               %   THEN set as ordinary paragraph.
     \else                         %   ELSE  center.
       \hbox to\hsize{\hfil\box\@tempboxa\hfil}\par
   \fi}

%

\long\def\@makecaption#1#2{
   \vskip 10pt
   \setbox\@tempboxa\hbox{\textbf{#1: #2}}
   \ifdim \wd\@tempboxa >\hsize % IF longer than one line:
       \textbf{#1: #2}\par               %   THEN set as ordinary paragraph.
     \else                         %   ELSE  center.
       \hbox to\hsize{\hfil\box\@tempboxa\hfil}
   \fi}

\@ifundefined{figure}{\newcounter {figure}} % this is for LaTeX2e

\def\fps@figure{tbp}
\def\ftype@figure{1}
\def\ext@figure{lof}
\def\fnum@figure{Figure \thefigure}
\def\figure{\@float{figure}}
\let\endfigure\end@float
\@namedef{figure*}{\@dblfloat{figure}}
\@namedef{endfigure*}{\end@dblfloat}

\@ifundefined{table}{\newcounter {table}} % this is for LaTeX2e

\def\fps@table{tbp}
\def\ftype@table{2}
\def\ext@table{lot}
\def\fnum@table{Table \thetable}
\def\table{\@float{table}}
\let\endtable\end@float
\@namedef{table*}{\@dblfloat{table}}
\@namedef{endtable*}{\end@dblfloat}

\newtoks\titleboxnotes
\newcount\titleboxnoteflag

\def\maketitle{\par
 \begingroup
   \def\thefootnote{\fnsymbol{footnote}}
   \def\@makefnmark{\hbox
       to 0pt{$^{\@thefnmark}$\hss}}
```

```
555       \twocolumn[\@maketitle]
    \@thanks
     \endgroup
     \setcounter{footnote}{0}
     \let\maketitle\relax
560  \let\@maketitle\relax
     \gdef\@thanks{}\gdef\@author{}\gdef\@title{}\gdef\@subtitle{}\let\thanks\relax
     \@copyrightspace}

    %% CHANGES ON NEXT LINES
565 \newif\if@ll % to record which version of LaTeX is in use

    \expandafter\ifx\csname LaTeXe\endcsname\relax % LaTeX2.09 is used
    \else% LaTeX2e is used, so set ll to true
    \global\@lltrue
570 \fi

    \if@ll
      \NeedsTeXFormat{LaTeX2e}
      \ProvidesClass{NIME-alternate} [2004/15/10 - V1.7 - based on acmproc.cls V1.3 ↙
          ↳ <Nov. 30 '99>]
575   \RequirePackage{latexsym}% QUERY: are these two really needed?
      \let\dooptions\ProcessOptions
    \else
      \let\dooptions\@options
    \fi
580 %% END CHANGES

    \def\@height{height}
    \def\@width{width}
    \def\@minus{minus}
585 \def\@plus{plus}
    \def\hb@xt@{\hbox to}
    \newif\if@faircopy
    \@faircopyfalse
    \def\ds@faircopy{\@faircopytrue}
590
    \def\ds@preprint{\@faircopyfalse}

    \@twosidetrue
    \@mparswitchtrue
595 \def\ds@draft{\overfullrule 5\p@}
    %% CHANGE ON NEXT LINE
    \dooptions

    \lineskip \p@
600 \normallineskip \p@
    \def\baselinestretch{1}
    \def\@ptsize{0} %needed for amssymbols.sty

    %% CHANGES ON NEXT LINES
605 \if@ll% allow use of old-style font change commands in LaTeX2e
    \@maxdepth\maxdepth
    %
    \DeclareOldFontCommand{\rm}{\ninept\rmfamily}{\mathrm}
    \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
610 \DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
```

```
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
\DeclareRobustCommand*{\cal}{\@fontswitch{\relax}{\mathcal}}
\DeclareRobustCommand*{\mit}{\@fontswitch{\relax}{\mathnormal}}
\fi
%
\if@ll
  \renewcommand{\rmdefault}{cmr}  % was 'ttm'
% Note! I have also found 'mvr' to work ESPECIALLY well.
% Gerry - October 1999
% You may need to change your LV1times.fd file so that sc is
% mapped to cmcsc - -for smallcaps -- that is if you decide
% to change {cmr} to {times} above. (Not recommended)
  \renewcommand{\@ptsize}{}
  \renewcommand{\normalsize}{%
    \@setfontsize\normalsize\@ixpt{10.5\p@}%\ninept%
    \abovedisplayskip 6\p@ \@plus2\p@ \@minus\p@
    \belowdisplayskip \abovedisplayskip
    \abovedisplayshortskip 6\p@ \@minus 3\p@
    \belowdisplayshortskip 6\p@ \@minus 3\p@
    \let\@listi\@listI
  }
\else
  \def\@normalsize{%changed next to 9 from 10
    \@setsize\normalsize{9\p@}\ixpt\@ixpt
   \abovedisplayskip 6\p@ \@plus2\p@ \@minus\p@
    \belowdisplayskip \abovedisplayskip
    \abovedisplayshortskip 6\p@ \@minus 3\p@
    \belowdisplayshortskip 6\p@ \@minus 3\p@
    \let\@listi\@listI
  }%
\fi
\if@ll
  \newcommand\scriptsize{\@setfontsize\scriptsize\@viipt{8\p@}}
  \newcommand\tiny{\@setfontsize\tiny\@vpt{6\p@}}
  \newcommand\large{\@setfontsize\large\@xiipt{14\p@}}
  \newcommand\Large{\@setfontsize\Large\@xivpt{18\p@}}
  \newcommand\LARGE{\@setfontsize\LARGE\@xviipt{20\p@}}
  \newcommand\huge{\@setfontsize\huge\@xxpt{25\p@}}
  \newcommand\Huge{\@setfontsize\Huge\@xxvpt{30\p@}}
\else
  \def\scriptsize{\@setsize\scriptsize{8\p@}\viipt\@viipt}
  \def\tiny{\@setsize\tiny{6\p@}\vpt\@vpt}
  \def\large{\@setsize\large{14\p@}\xiipt\@xiipt}
  \def\Large{\@setsize\Large{18\p@}\xivpt\@xivpt}
  \def\LARGE{\@setsize\LARGE{20\p@}\xviipt\@xviipt}
  \def\huge{\@setsize\huge{25\p@}\xxpt\@xxpt}
  \def\Huge{\@setsize\Huge{30\p@}\xxvpt\@xxvpt}
\fi
\normalsize

% make aubox hsize/number of authors up to 3, less gutter
% then showbox gutter showbox gutter showbox -- GKMT Aug 99
\newbox\@acmtitlebox
\def\@maketitle{\newpage
```

```
 \null
 \setbox\@acmtitlebox\vbox{%
\baselineskip 20pt
\vskip 2em                  % Vertical space above title.
   \begin{center}
    {\ttlfnt \@title\par}       % Title set in 18pt Helvetica (Arial) bold size.
    \vskip 1.5em                % Vertical space after title.
%This should be the subtitle.
{\subttlfnt \the\subtitletext\par}\vskip 1.25em%\fi
    {\baselineskip 16pt\aufnt   % each author set in \12 pt Arial, in a
     \lineskip .5em             % tabular environment
     \begin{tabular}[t]{c}\@author
     \end{tabular}\par}
    \vskip 1.5em                % Vertical space after author.
   \end{center}}
 \dimen0=\ht\@acmtitlebox
 \advance\dimen0 by -12.75pc\relax % Increased space for title box -- KBT
 \unvbox\@acmtitlebox
 \ifdim\dimen0<0.0pt\relax\vskip-\dimen0\fi}


\newcount\titlenotecount
\global\titlenotecount=0
\newtoks\tntoks
\newtoks\tntokstwo
\newtoks\tntoksthree
\newtoks\tntoksfour
\newtoks\tntoksfive

\def\abstract{
\ifnum\titlenotecount>0 % was =1
    \insert\footins{%
    \reset@font\footnotesize
        \interlinepenalty\interfootnotelinepenalty
        \splittopskip\footnotesep
        \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
        \hsize\columnwidth \@parboxrestore
        \protected@edef\@currentlabel{%
        }%
        \color@begingroup
\ifnum\titlenotecount=1
      \@maketntext{%
        \raisebox{4pt}{$\ast$}\rule\z@\footnotesep\ignorespaces\the\tntoks\↙
            ↳ @finalstrut\strutbox}%
\fi
\ifnum\titlenotecount=2
      \@maketntext{%
      \raisebox{4pt}{$\ast$}\rule\z@\footnotesep\ignorespaces\the\tntoks\par\↙
          ↳ @finalstrut\strutbox}%
\@maketntext{%
        \raisebox{4pt}{$\dagger$}\rule\z@\footnotesep\ignorespaces\the\↙
            ↳ tntokstwo\@finalstrut\strutbox}%
\fi
\ifnum\titlenotecount=3
      \@maketntext{%
        \raisebox{4pt}{$\ast$}\rule\z@\footnotesep\ignorespaces\the\tntoks\par\↙
            ↳ @finalstrut\strutbox}%
```

```
\@maketntext{%
          \raisebox{4pt}{$\dagger$}\rule\z@\footnotesep\ignorespaces\the\↙
              ↳ tntokstwo\par\@finalstrut\strutbox}%
\@maketntext{%
          \raisebox{4pt}{$\ddagger$}\rule\z@\footnotesep\ignorespaces\the\↙
              ↳ tntoksthree\@finalstrut\strutbox}%
\fi
\ifnum\titlenotecount=4
      \@maketntext{%
          \raisebox{4pt}{$\ast$}\rule\z@\footnotesep\ignorespaces\the\tntoks\par\↙
              ↳ @finalstrut\strutbox}%
\@maketntext{%
          \raisebox{4pt}{$\dagger$}\rule\z@\footnotesep\ignorespaces\the\↙
              ↳ tntokstwo\par\@finalstrut\strutbox}%
\@maketntext{%
          \raisebox{4pt}{$\ddagger$}\rule\z@\footnotesep\ignorespaces\the\↙
              ↳ tntoksthree\par\@finalstrut\strutbox}%
\@maketntext{%
          \raisebox{4pt}{$\S$}\rule\z@\footnotesep\ignorespaces\the\tntoksfour\↙
              ↳ @finalstrut\strutbox}%
\fi
\ifnum\titlenotecount=5
      \@maketntext{%
          \raisebox{4pt}{$\ast$}\rule\z@\footnotesep\ignorespaces\the\tntoks\par\↙
              ↳ @finalstrut\strutbox}%
\@maketntext{%
          \raisebox{4pt}{$\dagger$}\rule\z@\footnotesep\ignorespaces\the\↙
              ↳ tntokstwo\par\@finalstrut\strutbox}%
\@maketntext{%
          \raisebox{4pt}{$\ddagger$}\rule\z@\footnotesep\ignorespaces\the\↙
              ↳ tntoksthree\par\@finalstrut\strutbox}%
\@maketntext{%
          \raisebox{4pt}{$\S$}\rule\z@\footnotesep\ignorespaces\the\tntoksfour\↙
              ↳ par\@finalstrut\strutbox}%
\@maketntext{%
          \raisebox{4pt}{$\P$}\rule\z@\footnotesep\ignorespaces\the\tntoksfive\↙
              ↳ @finalstrut\strutbox}%
\fi
   \color@endgroup} %g}
\fi
\setcounter{footnote}{0}
\section*{ABSTRACT}\normalsize%\ninept
}

\def\endabstract{\if@twocolumn\else\endquotation\fi}

\def\keywords{\if@twocolumn
\section*{Keywords}
\else \small
\quotation
\fi}

\def\terms{\if@twocolumn
\section*{General Terms}
\else \small
\quotation
\fi}
```

```
% -- Classification needs to be a bit smart due to optionals - Gerry/Georgia ↙
    ↳ November 2nd. 1999
\newcount\catcount
\global\catcount=1

\def\category#1#2#3{%
\ifnum\catcount=1
\section*{Categories and Subject Descriptors}
\advance\catcount by 1\else{\unskip; }\fi
    \@ifnextchar [{\@category{#1}{#2}{#3}}{\@category{#1}{#2}{#3}[]}%
}

\def\@category#1#2#3[#4]{%
    \begingroup
        \let\and\relax
            #1 [\textbf{#2}]%
            \if!#4!%
                \if!#3!\else : #3\fi
            \else
                :\space
                \if!#3!\else #3\kern\z@---\hskip\z@\fi
                \textit{#4}%
            \fi
    \endgroup
}
%

%%% This section (written by KBT) handles the 1" box in the lower left
%%% corner of the left column of the first page by creating a picture,
%%% and inserting the predefined string at the bottom (with a negative
%%% displacement to offset the space allocated for a non-existent
%%% caption).
%%%
\newtoks\copyrightnotice
\def\ftype@copyrightbox{8}
\def\@copyrightspace{
\@float{copyrightbox}[b]
\begin{center}
\setlength{\unitlength}{1pc}
\begin{picture}(20,6) %Space for copyright notice
\put(0,-0.95){\crnotice{\@toappear}}
\end{picture}
\end{center}
\end@float}

\def\@toappear{} % Default setting blank - commands below change this.
\long\def\toappear#1{\def\@toappear{\parbox[b]{20pc}{\baselineskip 9pt#1}}}
\def\toappearbox#1{\def\@toappear{\raisebox{5pt}{\framebox[20pc]{\parbox[b]{19pc↙
    ↳ }{#1}}}}}

\newtoks\conf
\newtoks\confinfo
\def\conferenceinfo#1#2{\global\conf={#1}\global\confinfo{#2}}


\def\marginpar{\@latexerr{The \marginpar command is not allowed in the
```

```
  'acmconf' document style.}\@eha}

\mark{{}{}}     % Initializes TeX's marks

\def\today{\ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or December\fi
  \space\number\day, \number\year}

\def\@begintheorem#1#2{%
    \parskip 0pt % GM July 2000 (for tighter spacing)
    \trivlist
    \item[%
        \hskip 10\p@
        \hskip \labelsep
        {\sc #1}\hskip 5\p@\relax#2.}%
    ]
    \it
}
\def\@opargbegintheorem#1#2#3{%
    \parskip 0pt % GM July 2000 (for tighter spacing)
    \trivlist
    \item[%
        \hskip 10\p@
        \hskip \labelsep
        {\sc #1\ #2\             % This mod by Gerry to enumerate corollaries
   \setbox\@tempboxa\hbox{(#3)}  % and bracket the 'corollary title'
        \ifdim \wd\@tempboxa>\z@ % and retain the correct numbering of e.g. ↙
            ↳ theorems
            \hskip 5\p@\relax    % if they occur 'around' said corollaries.
            \box\@tempboxa       % Gerry - Nov. 1999.
        \fi.}%
    ]
    \it
}
\newif\if@qeded
\global\@qededfalse

% -- original
%\def\proof{%
%  \vspace{-\parskip} % GM July 2000 (for tighter spacing)
%    \global\@qededfalse
%    \@ifnextchar[{\@xproof}{\@proof}%
%}
% -- end of original

% (JSS) Fix for vertical spacing bug - Gerry Murray July 30th. 2002
\def\proof{%
\vspace{-\lastskip}\vspace{-\parsep}\penalty-51%
\global\@qededfalse
\@ifnextchar[{\@xproof}{\@proof}%
}

\def\endproof{%
    \if@qeded\else\qed\fi
    \endtrivlist
}
```

```
\def\@proof{%
    \trivlist
    \item[%
        \hskip 10\p@
        \hskip \labelsep
        {\sc Proof.}%
    ]
    \ignorespaces
}
\def\@xproof[#1]{%
    \trivlist
    \item[\hskip 10\p@\hskip \labelsep{\sc Proof #1.}]%
    \ignorespaces
}
\def\qed{%
    \unskip
    \kern 10\p@
    \begingroup
        \unitlength\p@
        \linethickness{.4\p@}%
        \framebox(6,6){}%
    \endgroup
    \global\@qededtrue
}

\def\newdef#1#2{%
    \expandafter\@ifdefinable\csname #1\endcsname
        {\@definecounter{#1}%
         \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
         \global\@namedef{#1}{\@defthm{#1}{#2}}%
         \global\@namedef{end#1}{\@endtheorem}%
    }%
}
\def\@defthm#1#2{%
    \refstepcounter{#1}%
    \@ifnextchar[{\@ydefthm{#1}{#2}}{\@xdefthm{#1}{#2}}%
}
\def\@xdefthm#1#2{%
    \@begindef{#2}{\csname the#1\endcsname}%
    \ignorespaces
}
\def\@ydefthm#1#2[#3]{%
    \trivlist
    \item[%
        \hskip 10\p@
        \hskip \labelsep
        {\it #2%
         \savebox\@tempboxa{#3}%
         \ifdim \wd\@tempboxa>\z@
            \ \box\@tempboxa
         \fi.%
        }]%
    \ignorespaces
}
\def\@begindef#1#2{%
    \trivlist
    \item[%
```

```
935             \hskip 10\p@
                \hskip \labelsep
                {\it #1\ \rm #2.}%
        ]%
    }
940 \def\theequation{\arabic{equation}}

    \newcounter{part}
    \newcounter{section}
    \newcounter{subsection}[section]
945 \newcounter{subsubsection}[subsection]
    \newcounter{paragraph}[subsubsection]
    \def\thepart{\Roman{part}}
    \def\thesection{\arabic{section}}
    \def\thesubsection{\thesection.\arabic{subsection}}
950 \def\thesubsubsection{\thesubsection.\arabic{subsubsection}} %removed \subsecfnt↙
        ↳ 29 July 2002 gkmt
    \def\theparagraph{\thesubsubsection.\arabic{paragraph}} %removed \subsecfnt 29 ↙
        ↳ July 2002 gkmt
    \newif\if@uchead
    \@ucheadfalse

955 %% CHANGES: NEW NOTE
    %% NOTE: OK to use old-style font commands below, since they were
    %% suitably redefined for LaTeX2e
    %% END CHANGES
    \setcounter{secnumdepth}{3}
960 \def\part{%
        \@startsection{part}{9}{\z@}{-10\p@ \@plus -4\p@ \@minus -2\p@}
            {4\p@}{\normalsize\@ucheadtrue}%
    }
    \def\section{%
965     \@startsection{section}{1}{\z@}{-10\p@ \@plus -4\p@ \@minus -2\p@}% GM
        {4\p@}{\baselineskip 14pt\secfnt\@ucheadtrue}%
    }

    \def\subsection{%
970     \@startsection{subsection}{2}{\z@}{-8\p@ \@plus -2\p@ \@minus -\p@}
        {4\p@}{\secfnt}%
    }
    \def\subsubsection{%
        \@startsection{subsubsection}{3}{\z@}{-8\p@ \@plus -2\p@ \@minus -\p@}%
975     {4\p@}{\subsecfnt}%
    }
    \def\paragraph{%
        \vskip 12pt\@startsection{paragraph}{3}{\z@}{6\p@ \@plus \p@}%
        {-5\p@}{\subsecfnt}%
980 }
    \let\@period=.
    \def\@startsection#1#2#3#4#5#6{%
            \if@noskipsec  %gkmt, 11 aug 99
            \global\let\@period\@empty
985         \leavevmode
            \global\let\@period.%
        \fi
          \par %
        \@tempskipa #4\relax
```

```
990         \@afterindenttrue
            \ifdim \@tempskipa <\z@
                \@tempskipa -\@tempskipa
                \@afterindentfalse
            \fi
995         \if@nobreak
            \everypar{}%
            \else
                \addpenalty\@secpenalty
                \addvspace\@tempskipa
1000        \fi
    \parskip=0pt % GM July 2000 (non numbered) section heads
            \@ifstar
                {\@ssect{#3}{#4}{#5}{#6}}
                {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%
1005    }
    \def\@sect#1#2#3#4#5#6[#7]#8{%
            \ifnum #2>\c@secnumdepth
                \let\@svsec\@empty
            \else
1010            \refstepcounter{#1}%
                \edef\@svsec{%
                    \begingroup
                        %\ifnum#2>2 \noexpand\rm \fi % changed to next 29 July 2002 gkmt
                                \ifnum#2>2 \noexpand#6 \fi
1015                    \csname the#1\endcsname
                    \endgroup
                    \ifnum #2=1\relax .\fi
                    \hskip 1em
                }%
1020        \fi
            \@tempskipa #5\relax
            \ifdim \@tempskipa>\z@
                \begingroup
                    #6\relax
1025                \@hangfrom{\hskip #3\relax\@svsec}%
                    \begingroup
                        \interlinepenalty \@M
                        \if@uchead
                            \uppercase{#8}%
1030                    \else
                            #8%
                        \fi
                        \par
                    \endgroup
1035            \endgroup
                \csname #1mark\endcsname{#7}%
                \vskip -12pt  %gkmt, 11 aug 99 and GM July 2000 (was -14) - numbered ↙
                    ↳ section head spacing
    \addcontentsline{toc}{#1}{%
                    \ifnum #2>\c@secnumdepth \else
1040                    \protect\numberline{\csname the#1\endcsname}%
                    \fi
                    #7%
                }%
            \else
1045            \def\@svsechd{%
```

```
                #6%
                \hskip #3\relax
                \@svsec
                \if@uchead
                    \uppercase{#8}%
                \else
                    #8%
                \fi
                \csname #1mark\endcsname{#7}%
                \addcontentsline{toc}{#1}{%
                    \ifnum #2>\c@secnumdepth \else
                        \protect\numberline{\csname the#1\endcsname}%
                    \fi
                    #7%
                }%
            }%
        \fi
        \@xsect{#5}\hskip 1pt
        \par
    }
    \def\@xsect#1{%
        \@tempskipa #1\relax
        \ifdim \@tempskipa>\z@
            \par
            \nobreak
            \vskip \@tempskipa
            \@afterheading
        \else
            \global\@nobreakfalse
            \global\@noskipsectrue
            \everypar{%
                \if@noskipsec
                    \global\@noskipsecfalse
                    \clubpenalty\@M
                    \hskip -\parindent
                    \begingroup
                        \@svsechd
                        \@period
                    \endgroup
                    \unskip
                    \@tempskipa #1\relax
                    \hskip -\@tempskipa
                \else
                    \clubpenalty \@clubpenalty
                    \everypar{}%
                \fi
            }%
        \fi
        \ignorespaces
    }
    \def\@trivlist{%
        \@topsepadd\topsep
        \if@noskipsec
            \global\let\@period\@empty
            \leavevmode
            \global\let\@period.%
        \fi
```

```
    \ifvmode
        \advance\@topsepadd\partopsep
    \else
        \unskip
        \par
    \fi
    \if@inlabel
        \@noparitemtrue
        \@noparlisttrue
    \else
        \@noparlistfalse
        \@topsep\@topsepadd
    \fi
    \advance\@topsep \parskip
    \leftskip\z@skip
    \rightskip\@rightskip
    \parfillskip\@flushglue
    \@setpar{\if@newlist\else{\@@par}\fi}
    \global\@newlisttrue
    \@outerparskip\parskip
}

%%% Actually, 'abbrev' works just fine as the default
%%% Bibliography style.

\typeout{Using 'Abbrev' bibliography style}
\newcommand\bibyear[2]{%
    \unskip\quad\ignorespaces#1\unskip
    \if#2..\quad \else \quad#2 \fi
}
\newcommand{\bibemph}[1]{{\em#1}}
\newcommand{\bibemphic}[1]{{\em#1\/}}
\newcommand{\bibsc}[1]{{\sc#1}}
\def\@normalcite{%
    \def\@cite##1##2{[##1\if@tempswa , ##2\fi]}%
}
\def\@citeNB{%
    \def\@cite##1##2{##1\if@tempswa , ##2\fi}%
}
\def\@citeRB{%
    \def\@cite##1##2{##1\if@tempswa , ##2\fi]}%
}
\def\start@cite#1#2{%
    \edef\citeauthoryear##1##2##3{%
        ###1%
        \ifnum#2=\z@ \else\ ###2\fi
    }%
    \ifnum#1=\thr@@
        \let\@@cite\@citeyear
    \else
        \let\@@cite\@citenormal
    \fi
    \@ifstar{\@citeNB\@@cite}{\@normalcite\@@cite}%
}
\def\cite{\start@cite23}
\def\citeNP{\cite*}
\def\citeA{\start@cite10}
```

```
\def\citeANP{\citeA*}
\def\shortcite{\start@cite23}
\def\shortciteNP{\shortcite*}
\def\shortciteA{\start@cite20}
\def\shortciteANP{\shortciteA*}
\def\citeyear{\start@cite30}
\def\citeyearNP{\citeyear*}
\def\citeN{%
    \@citeRB
    \def\citeauthoryear##1##2##3{##1\ [##3%
        \def\reserved@a{##1}%
        \def\citeauthoryear####1####2####3{%
            \def\reserved@b{####1}%
            \ifx\reserved@a\reserved@b
                ####3%
            \else
                \errmessage{Package acmart Error: author mismatch
                        in \string\citeN^^J^^J%
                    See the acmart package documentation for explanation}%
            \fi
        }%
    }%
    \@ifstar\@citeyear\@citeyear
}
\def\shortciteN{%
    \@citeRB
    \def\citeauthoryear##1##2##3{##2\ [##3%
        \def\reserved@a{##2}%
        \def\citeauthoryear####1####2####3{%
            \def\reserved@b{####2}%
            \ifx\reserved@a\reserved@b
                ####3%
            \else
                \errmessage{Package acmart Error: author mismatch
                        in \string\shortciteN^^J^^J%
                    See the acmart package documentation for explanation}%
            \fi
        }%
    }%
    \@ifstar\@citeyear\@citeyear  %GM July 2000
}
\def\@citenormal{%
    \@ifnextchar [{\@tempswatrue\@citex;}
                  {\@tempswafalse\@citex,[]}% %GM July 2000
}
\def\@citeyear{%
    \@ifnextchar [{\@tempswatrue\@citex,}%
                  {\@tempswafalse\@citex,[]}%
}
\def\@citex#1[#2]#3{%
    \let\@citea\@empty
    \@cite{%
        \@for\@citeb:=#3\do{%
            \@citea
            \def\@citea{#1 }%
            \edef\@citeb{\expandafter\@iden\@citeb}%
            \if@filesw
```

```
                \immediate\write\@auxout{\string\citation{\@citeb}}%
            \fi
            \@ifundefined{b@\@citeb}{%
                {\bf ?}%
                \@warning{%
                    Citation `\@citeb' on page \thepage\space undefined%
                }%
            }%
            {\csname b@\@citeb\endcsname}%
        }%
    }{#2}%
}
\let\@biblabel\@gobble
\newdimen\bibindent
\setcounter{enumi}{1}
\bibindent=0em
\def\thebibliography#1{% \ifnum\addauflag=0\addauthorsection\global\addauflag=1\fi
    \section[References]{%     <=== OPTIONAL ARGUMENT ADDED HERE
        {References} % was uppercased but this affects pdf bookmarks (SP/GM October 2004)
         \vskip -9pt  % GM July 2000 (for tighter spacing)
        \@mkboth{{\refname}}{{\refname}}%
    }%
    \list{[\arabic{enumi}]}{%
        \settowidth\labelwidth{[#1]}%
        \leftmargin\labelwidth
        \advance\leftmargin\labelsep
        \advance\leftmargin\bibindent
        \parsep=0pt\itemsep=1pt % GM July 2000
        \itemindent -\bibindent
        \listparindent \itemindent
        \usecounter{enumi}
    }%
    \let\newblock\@empty
    \raggedright % GM July 2000
    \sloppy
    \sfcode`\.=1000\relax
}


\gdef\balancecolumns
{\vfill\eject
\global\@colht=\textheight
\global\ht\@cclv=\textheight
}

\newcount\colcntr
\global\colcntr=0
\newbox\savebox

\gdef \@makecol {%
\global\advance\colcntr by 1
\ifnum\colcntr>2 \global\colcntr=1\fi
   \ifvoid\footins
     \setbox\@outputbox \box\@cclv
   \else
```

```
      \setbox\@outputbox \vbox{%
\boxmaxdepth \@maxdepth
        \@tempdima\dp\@cclv
1275    \unvbox \@cclv
        \vskip -\@tempdima
        \vskip \skip\footins
        \color@begingroup
          \normalcolor
1280      \footnoterule
          \unvbox \footins
        \color@endgroup
        }%
    \fi
1285    \xdef\@freelist{\@freelist\@midlist}%
    \global \let \@midlist \@empty
    \@combinefloats
    \ifvbox\@kludgeins
      \@makespecialcolbox
1290    \else
      \setbox\@outputbox \vbox to\@colht {%
\@texttop
        \dimen@ \dp\@outputbox
        \unvbox \@outputbox
1295    \vskip -\dimen@
        \@textbottom
        }%
    \fi
    \global \maxdepth \@maxdepth
1300 }
\def\titlenote{\@ifnextchar[\@xtitlenote{\stepcounter\@mpfn
\global\advance\titlenotecount by 1
\ifnum\titlenotecount=1
    \raisebox{9pt}{$\ast$}
1305 \fi
\ifnum\titlenotecount=2
    \raisebox{9pt}{$\dagger$}
\fi
\ifnum\titlenotecount=3
1310    \raisebox{9pt}{$\ddagger$}
\fi
\ifnum\titlenotecount=4
\raisebox{9pt}{$\S$}
\fi
1315 \ifnum\titlenotecount=5
\raisebox{9pt}{$\P$}
\fi
          \@titlenotetext
}}
1320
\long\def\@titlenotetext#1{\insert\footins{%
\ifnum\titlenotecount=1\global\tntoks={#1}\fi
\ifnum\titlenotecount=2\global\tntokstwo={#1}\fi
\ifnum\titlenotecount=3\global\tntoksthree={#1}\fi
1325 \ifnum\titlenotecount=4\global\tntoksfour={#1}\fi
\ifnum\titlenotecount=5\global\tntoksfive={#1}\fi
    \reset@font\footnotesize
    \interlinepenalty\interfootnotelinepenalty
```

```
    \splittopskip\footnotesep
    \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
    \hsize\columnwidth \@parboxrestore
    \protected@edef\@currentlabel{%
    }%
    \color@begingroup
  \color@endgroup}}

%%%%%%%%%%%%%%%%%%%%%%%%%
\ps@plain
\baselineskip=11pt
\let\thepage\relax % For NO page numbers - GM Nov. 30th. 1999 and July 2000
\def\setpagenumber#1{\global\setcounter{page}{#1}}
%\pagenumbering{arabic}  % Arabic page numbers GM July 2000
\twocolumn              % Double column.
\flushbottom            % Even bottom -- alas, does not balance columns at end of document
\pagestyle{plain}

% Need Copyright Year and Copyright Data to be user definable (in .tex file).
% Gerry Nov. 30th. 1999
\newtoks\copyrtyr
\newtoks\acmcopyr
\newtoks\boilerplate
\global\acmcopyr={X-XXXXX-XX-X/XX/XX}  % Default - 5/11/2001 *** Gerry
\global\copyrtyr={200X}                % Default - 3/3/2003 *** Gerry
\def\CopyrightYear#1{\global\copyrtyr{#1}}
\def\crdata#1{\global\acmcopyr{#1}}
\def\permission#1{\global\boilerplate{#1}}
%
\global\boilerplate={Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.}
\newtoks\copyrightetc
\global\copyrightetc{Copyright \the\copyrtyr\  \the\acmcopyr}

\toappear{\the\boilerplate\par
{\confname{\the\conf}} \the\confinfo\par \the\copyrightetc}
%
%% End of sig-alternate.cls -- V1.7 - 10/15/2004 --
%% Gerry Murray -- Wednesday October 15th. 2004
```

12 doc/setup.dot

```
digraph G {
        graph [bgcolor=white,rankdir=TB];
        edge [color=black];
        subgraph pdMain {
                node [shape=box];
                pdLoad [label="pd"];
        }
        subgraph hsextbootstrap {
                node [shape="diamond"];
                hsBootstrap [label="bootstrap.c"];
```

```
        }
        subgraph hsextHsExt {
                node [shape="ellipse"];
                hsSetup [label="HsExt.hs"];
        }
        subgraph hsextlibhsext {
                node [shape="parallelogram"];
                hslibSetup [label="libhsext.c"];
        }
        subgraph load {
                pdLoad -> hsBootstrap [label="hsext_setup"];
                hsBootstrap -> hsSetup [label="hsext_init"];
                hsSetup -> hslibSetup [label="libhsext_setup"];
        }
}
```

13 examples/Print-help.pd

```
#N canvas 8 107 343 303 10;
#X obj 73 200 hsext Print 1 2 a b;
#X msg 216 146 a b c;
#X msg 137 60 list a b c;
#X msg 169 96 symbol a;
#X msg 201 118 1 a 2 b;
#X msg 18 131 1;
#X obj 53 130 bng 15 250 50 0 empty empty empty 0 -6 0 8 -262144 -1
-1;
#X msg 73 105 list;
#X msg 83 133 a;
#X msg 160 141 1 2 3;
#X msg 22 84 float;
#X msg 91 166 symbol;
#X msg 120 15 float a;
#X msg 105 40 symbol 1;
#X msg 183 37 symbol a b;
#X msg 179 15 float 1 a;
#X msg 225 59 symbol a 1 2;
#X connect 1 0 0 4;
#X connect 2 0 0 2;
#X connect 3 0 0 3;
#X connect 4 0 0 4;
#X connect 5 0 0 0;
#X connect 6 0 0 0;
#X connect 7 0 0 0;
#X connect 8 0 0 0;
#X connect 9 0 0 2;
#X connect 10 0 0 0;
#X connect 11 0 0 0;
#X connect 12 0 0 1;
#X connect 13 0 0 1;
#X connect 14 0 0 1;
#X connect 15 0 0 1;
#X connect 16 0 0 4;
```

14 examples/Print.hs

```
-- hsext -- Haskell objects for Pd
-- Copyright (C) 2007 Claude Heiland-Allen
--
--
-- This program is free software; you can redistribute it and/or
-- modify it under the terms of the GNU General Public License
-- as published by the Free Software Foundation; either version 2
-- of the License, or (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, ↙
    ↳ USA.

-- print everything

module Print where

import Data.List (intersperse)
import PureData

creator :: Creator
creator args = do
  putStr   ("Print.creator: args<")
  sargs <- mapM showAtom args
  putStr   (concat (intersperse "," sargs))
  putStrLn (">")
  return (Just Instance{iInlets = 5, iOutlets = 0, iMethod = inlet})

inlet _ n selector message = do
  putStr    ("Print.method: inlet<" ++ (show n) ++ "> selector<")
  sselector <- peekSymbol selector
  putStr    (sselector)
  putStr    ("> message<")
  smsg <- mapM showAtom message
  putStr    (concat (intersperse "," smsg))
  putStrLn  (">")

showAtom (AFloat f) = do
  return ("float<" ++ (show f) ++ ">")

showAtom (ASymbol s) = do
  ss <- peekSymbol s
  return ("symbol<" ++ ss ++ ">")
```

15 examples/Swap-help.pd

```
#N canvas 0 0 450 300 10;
#X obj 48 136 hsext Swap;
#X floatatom 113 96 5 0 0 0 - - -;
#X floatatom 113 184 5 0 0 0 - - -;
#X floatatom 48 95 5 0 0 0 - - -;
```

```
#X floatatom 48 183 5 0 0 0 - - -;
#X obj 194 136 hsext Swap;
#X floatatom 259 96 5 0 0 0 - - -;
#X floatatom 259 184 5 0 0 0 - - -;
#X floatatom 194 95 5 0 0 0 - - -;
#X floatatom 194 183 5 0 0 0 - - -;
#X connect 0 0 4 0;
#X connect 0 1 2 0;
#X connect 1 0 0 1;
#X connect 3 0 0 0;
#X connect 5 0 9 0;
#X connect 5 1 7 0;
#X connect 6 0 5 1;
#X connect 8 0 5 0;
```

16 examples/Swap.hs

```
-- hsext -- Haskell objects for Pd
-- Copyright (C) 2007 Claude Heiland-Allen
--
--
-- This program is free software; you can redistribute it and/or
-- modify it under the terms of the GNU General Public License
-- as published by the Free Software Foundation; either version 2
-- of the License, or (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, ↙
    ↳ USA.

-- swap two messages

module Swap where

import Data.IORef
import PureData

creator :: Creator
creator args = do
  bang <- gensym "bang"
  state <- newIORef (bang, [])
  return (Just Instance{iInlets = 2, iOutlets = 2, iMethod = inlet state})

inlet state outlet 0 s m = do
  (rs,rm) <- readIORef state
  outlet 1 s m
  outlet 0 rs rm
  return ()

inlet state outlet 1 s m = do
  writeIORef state (s,m)
```

```
40    return ()
```

17 hsext/bootstrap.c

```
/*
hsext -- Haskell objects for Pd
Copyright (C) 2007 Claude Heiland-Allen

5
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.
10
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.
15
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
20
#include "HsExt_stub.h"

extern void __stginit_HsExt ( void );

25 extern void hsext_init(void);

extern void hsext_setup(void) { /* bootstrap */
    hs_init(0, 0);
    hs_add_root(__stginit_HsExt);
30  hsext_init();
}
```

18 hsext/Dispatcher.hs

```
module Dispatcher (
  Dispatcher,  -- abstract data type
  empty,       -- an empty dispatcher
  fallback,    -- a method to call when type unrecognized
5 insert,      -- associate a method to a type
  delete,      -- disassociate a method from a type
  dispatch     -- dispatch methods according to type
) where

10 import Data.Dynamic
import Data.Typeable
import Data.Maybe
import Data.IntMap(IntMap)
import qualified Data.IntMap as IntMap
15
data Dispatcher = D (IntMap Dynamic) (Maybe (Dynamic -> IO ()))

empty :: Dispatcher
```

```
empty = D IntMap.empty Nothing

fallback :: Maybe (Dynamic -> IO ()) -> Dispatcher -> IO Dispatcher
fallback f (D d _) = return (D d f)

insert :: (Typeable a) => a -> (a -> IO ()) -> Dispatcher -> IO Dispatcher
insert t m (D d f) = typeRepKey (typeOf t)
  >>= \k -> return (D (IntMap.insert k (toDyn m) d) f)

delete :: (Typeable a) => a -> Dispatcher -> IO Dispatcher
delete t (D d f) = typeRepKey (typeOf t)
  >>= \k -> return (D (IntMap.delete k d) f)

dispatch :: Dynamic -> Dispatcher -> IO ()
dispatch t (D d f) = do
  key <- typeRepKey (dynTypeRep t)
  case (IntMap.lookup key d) of
    Nothing -> case f of
      Nothing -> putStrLn "Dispatcher.dispatch: no fallback"
      Just f' -> f' t
    Just method -> case (dynApply method t) of
      Nothing -> putStrLn "Dispatcher.dispatch: dynApply -> Nothing"
      Just r -> case (fromDynamic r) of
        Nothing -> putStrLn "Dispatcher.dispatch: fromDynamic -> Nothing"
        Just r' -> r'

test = return empty
  >>= fallback (Just (\x -> putStrLn ("WTF I has a " ++ show x ++ "!")))
  >>= insert (undefined::Integer) (putStrLn . ("I has a Integer! "++) . show)
  >>= insert (undefined::String) (putStrLn . ("I has a String!  "++) . show)
  >>= insert (undefined::[Int]) (putStrLn . ("I has a [Int]!   "++) . show)
  >>= \d -> do
    dispatch (toDyn (1::Int)) d
    dispatch (toDyn ("hello"::String)) d
    dispatch (toDyn (1::Integer)) d
    dispatch (toDyn ([0..9]::[Int])) d
    delete (undefined::String) d
      >>= fallback Nothing
      >>= dispatch (toDyn ['H','i'])
```

19 hsext/Forward.c

```
/*
hsext -- Haskell objects for Pd
Copyright (C) 2007 Claude Heiland-Allen

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.
```

```
    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
    */

    #include <m_pd.h>

    typedef (struct hsobject *)(t_hsobject_constructor)(
      t_symbol *, int, t_atom *
    );

    typedef void (t_hsobject_destructor)(
      struct hsobject *
    );

    typedef void (t_hsinlet_dispatcher)(
      struct hsobject *, struct hsinlet *, t_symbol *, int, t_atom *
    );

    static t_class *hsinlet_class;
    static t_class *hsobject_class;

    extern void hsobject_setup(
      t_hsobject_constructor constructor,
      t_hsobject_destructor destructor,
      t_hsinlet_dispatcher dispatcher
    ) {
      /* create proxy inlet class */
      hsinlet_class = class_new(
        gensym("hsext inlet"), 0, 0, sizeof(struct hsinlet), 0, 0
      );
      class_addanything(hsinlet_class, dispatcher);
      /* create hsext class */
      hsobject_class = class_new(
        gensym("hsext"),
        (t_newmethod) constructor,
        (t_method) destructor,
        sizeof(t_libhsext),
        CLASS_NOINLET,
        A_GIMME,
        0
      );
    }


      post("lua: (GPL) 2007 Claude Heiland-Allen <claudiusmaximus@goto10.org>");
      L = lua_open();
      luaL_openlibs(L);
      Pd_Init(L);
      char buf[MAXPDSTRING];
      char *ptr;
      t_pdlua_readerdata reader;
      /* file "pd.lua" is some glue to ease the pain of SWIG */
      int fd = canvas_open(0, "pd", ".lua", buf, &ptr, MAXPDSTRING, 1);
      if (fd >= 0) {
        reader.fd = fd;
        if (lua_load(L, pdlua_reader, &reader, "pd.lua") ||
```

```
        lua_pcall(L, 0, 0, 0)) {
      post(
        "lua: error loading `pd.lua': %s",
        lua_tostring(L, -1)
      );
      post("lua: loader will not be registered!");
      lua_pop(L, 1);
      close(fd);
    } else {
      close(fd);
      sys_register_loader(pdlua_loader);
    }
  } else {
    post("lua: error loading `pd.lua': canvas_open() failed");
    post("lua: loader will not be registered!");
  }
```

20 hsext/HsExt.hs

```
-- hsext -- Haskell objects for Pd
-- Copyright (C) 2007 Claude Heiland-Allen
--
--
-- This program is free software; you can redistribute it and/or
-- modify it under the terms of the GNU General Public License
-- as published by the Free Software Foundation; either version 2
-- of the License, or (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, ↙
    ↳ USA.


module HsExt where

import Data.IORef
import Data.Map(Map)
import qualified Data.Map as Map
import Foreign.C.Types
import Foreign.C.String
import Foreign.Ptr
import Foreign.Marshal.Array
import Foreign.Marshal.Alloc (allocaBytes)

import System.Plugins.Make (
  make,
  MakeStatus(MakeFailure,MakeSuccess))

import System.Plugins.Load (
  pdynload,
  LoadStatus(LoadFailure,LoadSuccess))
```

```
import Pd

import PureData -- API

type InstanceID  = CInt

type InletNumber = CInt

type CallStack   = [InstanceID]

data State       = State {
  sCreator  :: Map (Ptr Symbol) Creator,
  sInstance :: Map InstanceID Instance,
  sObject   :: Map InstanceID (Ptr Object),
  sNextId   :: InstanceID}

type NewMethod =
  Ptr Symbol -> CInt -> Ptr Atom -> IO (Ptr Object)
foreign import ccall "wrapper"
  wrapNew :: NewMethod -> IO (FunPtr NewMethod)

type FreeMethod =
  Ptr Object -> IO ()
foreign import ccall "wrapper"
  wrapFree :: HsExt.FreeMethod -> IO (FunPtr HsExt.FreeMethod)

type InletMethod =
  InstanceID -> InletNumber -> Ptr Symbol -> CInt -> Ptr Atom -> IO ()
foreign import ccall "wrapper"
  wrapInlet :: InletMethod -> IO (FunPtr InletMethod)


foreign import ccall "libhsext.h libhsext_setup"
  libhsext_setup :: (FunPtr NewMethod) -> (FunPtr HsExt.FreeMethod)
                 -> (FunPtr InletMethod) -> IO ()

foreign import ccall "libhsext.h libhsext_new"
  libhsext_new :: CInt -> CInt -> CInt -> IO (Ptr Object)

foreign import ccall "libhsext.h libhsext_free"
  libhsext_free :: Ptr Object -> IO CInt

foreign import ccall "libhsext.h libhsext_outlet"
  libhsext_outlet :: Ptr Object -> CInt -> Ptr Symbol -> CInt -> Ptr Atom
                  -> IO ()

foreign import ccall "libhsext.h libhsext_find"
  libhsext_find :: CString -> CUInt -> Ptr CChar -> IO ()


foreign export ccall hsext_init :: Setup
hsext_init :: Setup
hsext_init = do
  sref <- newIORef s0
  newMethod <- wrapNew (new sref)
  freeMethod <- wrapFree (free sref)
```

```
95      inletMethod <- wrapInlet (inlet sref)
        libhsext_setup newMethod freeMethod inletMethod
        return ()
        where
          s0 = State{
100         sCreator = Map.empty,
            sInstance = Map.empty,
            sObject = Map.empty,
            sNextId = 1}

105       new :: IORef State -> NewMethod
          new sref _ argc argv
            | argc > 0 = do
              msg <- peekArray (fromIntegral argc) argv
              case (head msg) of
110             ASymbol sel -> do
                  mbc <- loader sref sel
                  s <- readIORef sref
                  case mbc of
                    Nothing -> do
115                   return nullPtr
                    Just c -> do
                      mbins <- c (tail msg)
                      case mbins of
                        Nothing -> do
120                       return nullPtr
                        Just ins -> do
                          pd <- libhsext_new (sNextId s) (iInlets ins) (iOutlets ins)
                          writeIORef sref s{
                            sInstance = Map.insert (sNextId s) ins (sInstance s),
125                         sObject   = Map.insert (sNextId s) pd  (sObject s),
                            sNextId   = (sNextId s) + 1}
                          return pd
              _ -> do return nullPtr
            | otherwise = do return nullPtr
130

          free :: IORef State -> HsExt.FreeMethod
          free sref pd = do
            s <- readIORef sref
            insId <- libhsext_free pd
135         writeIORef sref s{
              sObject   = Map.delete insId (sObject   s),
              sInstance = Map.delete insId (sInstance s)}

          inlet :: IORef State -> InletMethod
140       inlet sref insId n sel argc argv = do
            s <- readIORef sref
            case (Map.lookup insId (sInstance s)) of
              Nothing -> do
                putStrLn "hsext.inlet: can't find instance"
145           Just ins -> do
                msg <- peekArray (fromIntegral argc) argv
                (iMethod ins) (outlet sref insId) n sel msg

          outlet :: IORef State -> InstanceID -> Outlet
150       outlet sref insId n sel msg = do
            s <- readIORef sref
```

```
      case (Map.lookup insId (sObject s)) of
        Nothing -> do
          putStrLn "hsext.outlet: can't find object"
        Just pd -> do
          withArrayLen msg (\argc argv ->
            libhsext_outlet pd n sel (fromIntegral argc) argv)

-- TODO: use Pd's path
    loader :: IORef State -> Ptr Symbol -> IO (Maybe Creator)
    loader sref sel = do
      s <- readIORef sref
      apipath <- getApiPath
      case (Map.lookup sel (sCreator s)) of
        Just c  -> do
          return (Just c)
        Nothing -> do
          filename <- getSource sel    -- "-v" in make options breaks things
          makestatus <- make filename ("-O2":(map ("-i" ++) apipath))
          case makestatus of
            MakeFailure e -> do
              putStrLn "hsext: MakeFailure"
              mapM_ putStrLn e
              return Nothing
            MakeSuccess _ o -> do
              loadstatus <- pdynload o apipath [] "PureData.Creator" "creator"
              case loadstatus of
                LoadFailure e -> do
                  putStrLn "hsext: LoadFailure"
                  mapM_ putStrLn e
                  return Nothing
                LoadSuccess _ c -> do
                  writeIORef sref s{sCreator = Map.insert sel c (sCreator s)}
                  return (Just c)
      where
        getApiPath = do
          ps <- peekPath
          return (ps ++ (map (++ "/api") ps))
        getSource s = do
          name <- peekSymbol s
          dir <- allocaBytes 1024 (\p -> do
            withCString name (\cname -> libhsext_find cname 1024 p)
            d <- peekCString p
            return d)
          return (dir ++ "/" ++ name ++ ".hs") -- TODO: platform dependent
```

21 hsext/libhsext.c

```
/*
hsext -- Haskell objects for Pd
Copyright (C) 2007 Claude Heiland-Allen

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.
```

```
    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
    */

    #include "libhsext.h"

    #include <unistd.h>
    #include <string.h>

    /* ~~~~ global variables ~~~~ */
    static t_class    *libhsext_inlet_class;   /* proxy inlet class */
    static t_class    *libhsext_receive_class; /* proxy receive class */
    static t_class    *libhsext_class;         /* real class */
    static t_inletf   *libhsext_inletf;        /* inlet dispatcher */
    static t_receivef *libhsext_receivef;      /* receive dispatcher */

    /* ~~~~ anything method for our proxy inlet ~~~~ */
    static void libhsext_inlet(
        t_libhsext_inlet *hsi, t_symbol *s, int argc, t_atom *argv
    ) {
        /* dispatch based on proxy inlet contents */
        (libhsext_inletf)(hsi->hsi_id, hsi->hsi_number, s, argc, argv);
    }

    /* ~~~~ anything method for our proxy receive ~~~~ */
    static void libhsext_receive(
        t_libhsext_receive *hsr, t_symbol *s, int argc, t_atom *argv
    ) {
        /* dispatch based on proxy receive contents */
        (libhsext_receivef)(hsr->hsr_id, hsr->hsr_number, s, argc, argv);
    }

    /* ~~~~ set up C stuff ~~~~ */
    extern void libhsext_setup(
        t_newf newf, t_freef freef, t_inletf inletf, t_receivef receivef
    ) {
        /* create proxy inlet class */
        libhsext_inletf = inletf;
        libhsext_inlet_class = class_new(
            gensym("hsext inlet"), 0, 0, sizeof(t_libhsext_inlet), 0, 0
        );
        class_addanything(libhsext_inlet_class, libhsext_inlet);

        /* create proxy receive class */
        libhsext_receivef = receivef;
        libhsext_receive_class = class_new(
            gensym("hsext receive"), 0, 0, sizeof(t_libhsext_receive),
            CLASS_NOINLET, 0
        );
        class_addanything(libhsext_receive_class, libhsext_receive);
```

```
    libhsext_class = class_new(  /* create hsext class */
        gensym("hsext"),
        (t_newmethod) newf,
        (t_method) freef,
        sizeof(t_libhsext),
        CLASS_NOINLET,      /* inlets defined on a per-instance basis */
        A_GIMME,            /* the raw unadulterated truth */
        0
    );
}

/* ~~~~ constructor ~~~~ */
extern t_libhsext *libhsext_new(
    int id, int inlets, int outlets, int receives, t_symbol **receivenames
) {
    t_libhsext *hs = NULL;
    int i, o;
    hs = (t_libhsext *) pd_new(libhsext_class);
    if (!hs) goto bork;
    hs->hs_id           = id;
    hs->hs_inletcount   = inlets;
    hs->hs_receivecount = receives;
    hs->hs_outletcount  = outlets;
    hs->hs_inlets       = NULL; /* lest we bork */
    hs->hs_receives     = NULL; /* lest we bork */
    hs->hs_outlets      = NULL; /* lest we bork */

    /* create proxy inlets */
    if (hs->hs_inletcount > 0) {
        if (!(hs->hs_inlets = (t_libhsext_inlet *)
            getbytes(hs->hs_inletcount * sizeof(t_libhsext_inlet)))) goto bork;
        for (i = 0; i < hs->hs_inletcount; i++) {
            hs->hs_inlets[i].hsi_owner  = libhsext_inlet_class;
            hs->hs_inlets[i].hsi_id     = hs->hs_id;
            hs->hs_inlets[i].hsi_number = i;
            if (!(inlet_new(&hs->x_obj, &hs->hs_inlets[i].hsi_owner, 0, 0)))
                goto bork;
        }
    }

    /* create proxy receives */
/*    if (hs->hs_receivecount > 0) {
        if (!(hs->hs_receives = (t_libhsext_receive *)
            getbytes(hs->hs_receivecount * sizeof(t_libhsext_receive)))) goto ↙
                ↳ bork;
        for (i = 0; i < hs->hs_receivecount; i++) {
            hs->hs_receives[i].hsr_owner  = libhsext_receive_class;
            hs->hs_receives[i].hsr_id     = hs->hs_id;
            hs->hs_receives[i].hsr_number = i;
            pd_bind(&hs->x_obj.ob_pd, s);
void pd_bind(t_pd *x, t_symbol *s)

        }
    }
*/

    /* create outlets */
```

```
    if (hs->hs_outletcount > 0) {
        if (!(hs->hs_outlets = (t_outlet **)
            getbytes(hs->hs_outletcount * sizeof(t_outlet *)))) goto bork;
        for (o = 0; o < hs->hs_outletcount; o++) {
            if (!(hs->hs_outlets[o] = outlet_new(&hs->x_obj, 0))) goto bork;
        }
    }

    return (hs);
bork: /* something went wrong */
    libhsext_free(hs);
    hs = NULL;
    return (NULL);
}

/* ~~~~ destructor ~~~~ */
extern void libhsext_free(t_libhsext *hs) {
    int o;
    if (hs) {
        if (hs->hs_inlets) { /* don't know if the following is safe */
            freebytes(hs->hs_inlets, hs->hs_inletcount * sizeof(t_libhsext_inlet
                ↳ ));
            hs->hs_inlets = NULL;
        }
        if (hs->hs_outlets) { /* free outlets */
            for (o = 0; o < hs->hs_outletcount; o++) {
                if (hs->hs_outlets[o]) {
                    outlet_free(hs->hs_outlets[o]);
                    hs->hs_outlets[o] = NULL;
                } else {
                    break; /* outlets allocated in order */
                }
            }
            freebytes(hs->hs_outlets, hs->hs_outletcount * sizeof(t_outlet *));
            hs->hs_outlets = NULL;
        }
    }
}

/* ~~~~ send anything to one of our outlets ~~~~ */
extern void libhsext_outlet(
    t_libhsext *hs, int o, t_symbol *s, int argc, t_atom *argv
) {
    if (hs) {
        if (0 <= o && o < hs->hs_outletcount) {
            if (hs->hs_outlets) {
                if (hs->hs_outlets[o]) {
                    outlet_anything(hs->hs_outlets[o], s, argc, argv);
                    return;
                }
            }
        }
    }
    post("hsext_outlet_wrapper: null pointer somewhere along the line");
}

/* ~~~~ helper for finding a .hs or .lhs file in Pd's path ~~~~ */
```

```
extern void libhsext_find(const char *name, unsigned int size, char *result) {
    int fd;
    char *nameresult;
    fd = open_via_path("", name, ".hs", result, &nameresult, size, 0);
    if (fd > 0) {
        close(fd);
    } else {
        nameresult = NULL;
        strcpy(result, "."); /* not found in path, last resort */
    }
    if (result == nameresult) {
        strcpy(result, "."); /* found in current dir */
    }
}
```

22 hsext/libhsext.h

```
/*
hsext -- Haskell objects for Pd
Copyright (C) 2007 Claude Heiland-Allen

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/

#ifndef LIBHSEXT_H
#define LIBHSEXT_H 1

#include <m_pd.h>

/* ~~~~ each instance of hsext is one of these in C-space ~~~~ */
typedef struct _libhsext {
    t_object x_obj;        /* just because Pd likes it this way */
    t_int hs_id;           /* passed by Haskell-space creator */
    t_int hs_inletcount;   /* passed by Haskell-space creator */
    t_int hs_receivecount; /* passed by Haskell-space creator */
    t_int hs_outletcount;  /* passed by Haskell-space creator */
    struct _libhsext_inlet   *hs_inlets;   /* we manage our inlets   */
    struct _libhsext_receive *hs_receives; /* we manage our receives */
    t_outlet                 **hs_outlets; /* we manage our outlets  */
} t_libhsext;

/* ~~~~ one of these for each inlet of a given hsext instance ~~~~ */
typedef struct _libhsext_inlet {
    t_pd hsi_owner;    /* this is set to a proxy class */
```

```
    t_int hsi_id;       /* passed by Haskell-space creator, == hs_id */
    t_int hsi_number;   /* 0 = leftmost, 1 = next-from-leftmost, ... */
} t_libhsext_inlet;

/* ~~~~ one of these for each receiver of a given hsext instance ~~~~ */
typedef struct _libhsext_receive {
    t_pd hsr_owner;     /* this is set to a proxy class */
    t_int hsr_id;       /* passed by Haskell-space creator, == hs_id */
    t_int hsr_number;   /* from the order in which names were passed */
} t_libhsext_receive;

/* ~~~~ function pointer types ~~~~ */
typedef void (t_newf  )(t_symbol *, int, t_atom *);
typedef void (t_freef )(t_libhsext *);
typedef void (t_inletf)(int, int, t_symbol *, int, t_atom *);
typedef void (t_receivef)(int, int, t_symbol *, int, t_atom *);

/* ~~~~ called from Haskell-space ~~~~ */
extern void libhsext_setup(
    t_newf newf, t_freef freef, t_inletf inletf, t_receivef receivef
);
extern t_libhsext *libhsext_new(
    int id, int inlets, int outlets, int receives, t_symbol **receivenames
);
extern void libhsext_free(t_libhsext *hs);
extern void libhsext_outlet(
    t_libhsext *hs, int o, t_symbol *s, int argc, t_atom *argv
);
extern void libhsext_find(const char *name, unsigned int size, char *result);

#endif
```

23 hsext/loader.h

```
#ifndef LOADER_H
#define LOADER_H 1
#include <m_pd.h>
typedef int (*loader_t)(t_canvas *, char *);
extern void sys_register_loader(loader_t loader);
#endif
```

24 hsext/LowLevel.h

```
/*
hsext -- Haskell objects for Pd
Copyright (C) 2007 Claude Heiland-Allen

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
```

```
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/

#ifndef LOWLEVEL_H
#define LOWLEVEL_H 1

#include <m_pd.h>

struct hsobject {
  t_object object;
};

struct hsinlet {
  t_pd proxy;
  struct hsobject *owner;
};

struct hsoutlet {
  t_outlet *outlet;
};

#endif
```

25 hsext/LowLevel.hsc

```
{-# OPTIONS -fffi #-}
-- hsext -- Haskell objects for Pd
-- Copyright (C) 2007 Claude Heiland-Allen
--
--
-- This program is free software; you can redistribute it and/or
-- modify it under the terms of the GNU General Public License
-- as published by the Free Software Foundation; either version 2
-- of the License, or (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, ↙
    ↳ USA.


module LowLevel (
  Class,
  Object,
  InletProxy,
  InletNumber,
  Inlet,
  OutletStruct,
```

```
  Outlet,
) where

#include "m_pd.h"
#include "s_stuff.h"

#include "LowLevel.h"

data Class = Class

data Object = Object

type InletProxy  = Ptr Class

type InletNumber = (#type int)

data Inlet  = Inlet InletProxy (Ptr Object) InletNumber
instance Storable Inlet where
  alignment = 1
  sizeOf _ = (#size struct hsinlet)
  peek inlet = do
    proxy  = (#peek struct hsinlet, proxy)  inlet
    owner  = (#peek struct hsinlet, owner)  inlet
    number = (#peek struct hsinlet, number) inlet
    return (Inlet proxy owner number)
  poke inlet (Inlet proxy owner number) = do
    (#poke struct hsinlet, proxy)  inlet proxy
    (#poke struct hsinlet, owner)  inlet owner
    (#poke struct hsinlet, number) inlet number

data OutletStruct = OutletStruct

data Outlet = Outlet (Ptr OutletStruct)
instance Storable Outlet where
  alignment = 1
  sizeOf _ = (#size struct hsoutlet)
  peek outlet = do
    outletptr = (#peek struct hsoutlet, outlet) outlet
    return (Outlet outletptr)
  poke outlet (Outlet outletptr) = do
    (#poke struct hsoutlet, outlet) outlet outletptr
```

26 hsext/Pd.hs

```
-- hsext -- Haskell objects for Pd
-- Copyright (C) 2007 Claude Heiland-Allen
--
--
-- This program is free software; you can redistribute it and/or
-- modify it under the terms of the GNU General Public License
-- as published by the Free Software Foundation; either version 2
-- of the License, or (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
-- GNU General Public License for more details.
```

```
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, ↙
    ↳ USA.

module Pd (
  Setup,
  PdInt,
  PdFloat,
  PdFloatArg,
  ClassFlags,
  AtomType,
  GimmeTerminator,
  Symbol,
  Atom(AFloat,ASymbol),
  Class,
  Object,
  NewMethodGimme,
  FreeMethod,
  ClassNewGimme,
  peekSymbol,
  gensym,
  peekPath,
) where

import Foreign.Ptr
import Foreign.C.Types

import PdStructs
-- typedef struct _symbol { char *s_name; } t_symbol;

type Setup           = IO ()

type PdInt           = CLong
type PdFloat         = CFloat
type PdFloatArg      = CFloat
type ClassFlags      = CInt
type AtomType        = CInt
type GimmeTerminator = CInt

type Class           = ()
type Object          = ()

type NewMethodGimme  = Ptr Symbol -> CInt -> Ptr Atom -> IO (Ptr Object)
type FreeMethod      = Ptr Object -> IO ()
type ClassNewGimme   = Ptr Symbol -> NewMethodGimme -> FreeMethod ->
  CSize -> ClassFlags -> AtomType -> GimmeTerminator -> IO (Ptr Class)
```

27 hsext/PdStructs.hsc

```
{-# OPTIONS -fffi #-}
-- hsext -- Haskell objects for Pd
-- Copyright (C) 2007 Claude Heiland-Allen
--
--
-- This program is free software; you can redistribute it and/or
```

```
-- modify it under the terms of the GNU General Public License
-- as published by the Free Software Foundation; either version 2
-- of the License, or (at your option) any later version.
--
-- This program is distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
-- GNU General Public License for more details.
--
-- You should have received a copy of the GNU General Public License
-- along with this program; if not, write to the Free Software
-- Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, ↙
    ↳ USA.

module PdStructs (
  peekSymbol,
  gensym,
  Symbol,
  Atom(AFloat,ASymbol),
  peekPath,
) where

#include "m_pd.h"
#include "s_stuff.h"

import Foreign.C.Types
import Foreign.C.String
import Foreign.Ptr
import Foreign.Storable

data PdVersion = PdVersion {
  majorVersion  :: Int,
  minorVersion  :: Int,
  bugfixVersion :: Int,
  testVersion   :: String }
    deriving (Show, Read, Eq, Ord)

compiledAgainst :: PdVersion
compiledAgainst = PdVersion {
  majorVersion  = (#const PD_MAJOR_VERSION),
  minorVersion  = (#const PD_MINOR_VERSION),
  bugfixVersion = (#const PD_BUGFIX_VERSION),
  testVersion   = (#const_str PD_TEST_VERSION) }

maxPdString :: Int
maxPdString = (#const MAXPDSTRING)

maxPdArg :: Int
maxPdArg = (#const MAXPDARG)

type PdInt = (#type t_int)
type PdFloat = (#type t_float)
type PdFloatArg = (#type t_floatarg)

data PdSymbol = PdSymbol {
  sName  :: Ptr CChar,
  sThing :: Ptr (Ptr PdClass),
```

```
    sNext  :: Ptr PdSymbol }

instance Storable PdSymbol where
  sizeOf    _ = (#size struct _symbol)
  alignment _ = 1    -- FIXME
  peek sp     = do
    name  <- (#peek struct _symbol, s_name)  sp
    thing <- (#peek struct _symbol, s_thing) sp
    next  <- (#peek struct _symbol, s_next)  sp
    return (PdSymbol { sName = name, sThing = thing, sNext = next })
  poke sp  (PdSymbol { sName = name, sThing = thing, sNext = next }) = do
    (#poke struct _symbol, s_name)  sp name
    (#poke struct _symbol, s_thing) sp thing
    (#poke struct _symbol, s_next)  sp next


foreign import ccall "m_pd.h gensym" c_gensym
  :: CString -> IO (Ptr PdSymbol)

gensym :: String -> IO (Ptr PdSymbol)
gensym s = withCString s c_gensym


type PdGStubType = (#type int)

stubNone  :: PdGStubType
stubNone  = (#const GP_NONE)

stubGList :: PdGStubType
stubGList = (#const GP_GLIST)

stubArray :: PdGStubType
stubArray = (#const GP_ARRAY)

data PdGStub = PdGStubNone  { gsRefCount :: CInt }
             | PdGStubGList { gsRefCount :: CInt, gsGList :: (Ptr PdGList) }
             | PdGStubArray { gsRefCount :: CInt, gsArray :: (Ptr PdArray) }

instance Storable PdGStub where
  sizeOf    _ = (#size struct _gstub)
  alignment _ = 1    -- FIXME
  peek gsp = do
    glist    <- (#peek struct _gstub, gs_un.gs_glist) gsp
    array    <- (#peek struct _gstub, gs_un.gs_array) gsp
    which    <- (#peek struct _gstub, gs_which)       gsp
    refCount <- (#peek struct _gstub, gs_refcount)    gsp
    return (case which of
      | stubNone  -> PdGStubNone  { gsRefCount = refcount })
      | stubGList -> PdGStubGList { gsRefCount = refcount, gsGList glist })
      | stubArray -> PdGStubArray { gsRefCount = refcount, gsArray array }) )
  poke gsp (PdGStubNone  { gsRefCount = refcount }) = do
    (#poke struct _gstub, gs_which)       gsp stubNone
    (#poke struct _gstub, gs_refcount)    gsp refcount
  poke gsp (PdGStubGList { gsRefCount = refcount, gsGList = glist }) = do
    (#poke struct _gstub, gs_un.gs_glist) gsp glist
    (#poke struct _gstub, gs_which)       gsp stubGList
```

```
      (#poke struct _gstub, gs_refcount)    gsp refcount
    poke gsp (PdGStubArray { gsRefCount = refcount, gsArray = array }) = do
      (#poke struct _gstub, gs_un.gs_array) gsp glist
      (#poke struct _gstub, gs_which)       gsp stubArray
      (#poke struct _gstub, gs_refcount)    gsp refcount

data PdGPointer = PdGPointer

instance Storeable PdGPointer where
  sizeOf    _ = (#size struct _gpointer)
  alignment _ = 1    -- FIXME
  peek      _ = error "peek unimplemented for 'struct _gpointer' (Storeable ↙
      ↳ PdGPointer)"
  poke    _ _ = error "poke unimplemented for 'struct _gpointer' (Storeable ↙
      ↳ PdGPointer)"

typedef union word
{
    t_float w_float;
    t_symbol *w_symbol;
    t_gpointer *w_gpointer;
    t_array *w_array;
    struct _glist *w_list;
    int w_index;
} t_word;

typedef enum
{
    A_NULL,
    A_FLOAT,
    A_SYMBOL,
    A_POINTER,
    A_SEMI,
    A_COMMA,
    A_DEFFLOAT,
    A_DEFSYM,
    A_DOLLAR,
    A_DOLLSYM,
    A_GIMME,
    A_CANT
}  t_atomtype;

#define A_DEFSYMBOL A_DEFSYM    /* better name for this */

typedef struct _atom
{
    t_atomtype a_type;
    union word a_w;
} t_atom;

data Atom = AFloat CFloat
          | ASymbol (Ptr Symbol)
--  deriving (Eq, Show)


instance Storable Atom where
```

```
175
    sizeOf _ = (#size struct _atom)

    alignment _ = 1     -- FIXME

180 peek p = do
      t <- ((#peek struct _atom, a_type) p)::IO CInt
      case t of

        (#const A_FLOAT) -> do
185       f <- (#peek struct _atom, a_w.w_float) p
          return (AFloat f)

        (#const A_SYMBOL) -> do
          s <- (#peek struct _atom, a_w.w_symbol) p
190       return (ASymbol s)

        _ -> error "Atom.peek: unsupported atom type"

    poke p (AFloat f) = do
195   (#poke struct _atom, a_type) p ((#const A_FLOAT)::CInt)
      (#poke struct _atom, a_w.w_float) p f

    poke p (ASymbol s) = do
      (#poke struct _atom, a_type) p ((#const A_SYMBOL)::CInt)
200   (#poke struct _atom, a_w.w_symbol) p s

    poke _ _ = error "Atom.poke: unsupported atom type"


205 -- typedef struct _namelist    /* element in a linked list of stored strings */
    -- {
    --     struct _namelist *nl_next;  /* next in list */
    --     char *nl_string;            /* the string */
    -- } t_namelist;
210 -- extern t_namelist *sys_searchpath;

    type NameList = ()

    nlNext :: Ptr NameList -> IO (Ptr NameList)
215 nlNext p = (#peek struct _namelist, nl_next) p
    nlName :: Ptr NameList -> IO (CString)
    nlName p = (#peek struct _namelist, nl_string) p

    foreign import ccall "s_stuff.h &sys_searchpath" raw_searchpath
220   :: Ptr (Ptr NameList)

    peekPath :: IO [String]
    peekPath = do
      p <- peek raw_searchpath
225   f p []
      where
        f lp ps
          | lp == nullPtr = do
            return ps
230       | otherwise     = do
            nlp <- nlNext lp
```

```
        q   <- nlName lp
        s   <- peekCString q
        ss  <- f nlp ps
235     return (s:ss)
```

28 Makefile

```
# hsext -- Haskell objects for Pd
# Copyright (C) 2007 Claude Heiland-Allen
#
#
5 # This program is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License
# as published by the Free Software Foundation; either version 2
# of the License, or (at your option) any later version.
#
10 # This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
15 # You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA↙
    ↳ .

PDSRC = $(HOME)/src/pd-0.40-2/src
20
CFLAGS = -I$(PDSRC) -I/opt/haskell/lib/ghc-6.6.1/include -I./build -Wall
HCFLAGS = -I$(PDSRC) -I./hsext
#HFLAGS = -odir ./build -hidir ./build -stubdir ./build -Wall -fffi
HFLAGS = -i./api -i./hsext -i./build -odir ./build -hidir ./build -Wall -fffi
25 #

all: build/LowLevel.hs

notall: hsext.pd_linux
30
clean:
        rm -f build/bootstrap.o build/HsExt.hi build/HsExt.o \
build/HsExt_stub.c build/HsExt_stub.h build/HsExt_stub.o build/libhsext.o \
build/Pd.hi build/Pd.o build/PdStructs.hi build/PdStructs.hs build/PdStructs.o \
35 build/PureData.hi build/PureData.o examples/Print.hi examples/Print.o \
examples/Swap.hi examples/Swap.o

dist-clean:
        rm -f hsext.pd_linux api/Pd.hi api/PureData.hi api/PdStructs.hi \
40 examples/Swap.hi examples/Swap.o examples/Print.hi examples/Print.o

#

hsext.pd_linux: build/bootstrap.o build/HsExt.o build/HsExt_stub.o build/Pd.o ↙
    ↳ build/PdStructs.o build/libhsext.o build/PureData.o
45      ghc $(HFLAGS) -no-hs-main -package plugins -optl -shared -o hsext.↙
            ↳ pd_linux build/bootstrap.o build/HsExt.o build/HsExt_stub.o build/↙
            ↳ Pd.o build/PdStructs.o build/libhsext.o build/PureData.o
        strip hsext.pd_linux
```

```
build/HsExt.o build/Pd.o build/PdStructs.o build/HsExt_stub.h build/HsExt_stub.o↙
    ↳ : hsext/HsExt.hs hsext/Pd.hs api/PureData.hs hsext/libhsext.h build/↙
    ↳ PdStructs.hs
        ghc $(HFLAGS) -no-hs-main --make -c hsext/HsExt.hs
        mv -f hsext/HsExt_stub.c build/HsExt_stub.c
        mv -f hsext/HsExt_stub.h build/HsExt_stub.h
        cp -f build/PureData.hi  api/PureData.hi
        cp -f build/PdStructs.hi api/PdStructs.hi
        cp -f build/Pd.hi        api/Pd.hi

build/PdStructs.hs: hsext/PdStructs.hsc
        hsc2hs $(HCFLAGS) -o build/PdStructs.hs hsext/PdStructs.hsc

build/libhsext.o: hsext/libhsext.c hsext/libhsext.h
        gcc $(CFLAGS) -c -o build/libhsext.o hsext/libhsext.c

build/bootstrap.o: hsext/bootstrap.c build/HsExt_stub.h
        gcc $(CFLAGS) -c -o build/bootstrap.o hsext/bootstrap.c

#

build/LowLevel.hs: hsext/LowLevel.hsc hsext/LowLevel.h
        hsc2hs $(HCFLAGS) -o build/LowLevel.hs hsext/LowLevel.hsc
```

29 README

```
hsext -- Haskell objects for Pd
Copyright (C) 2007 Claude Heiland-Allen


This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~ General Documentation ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

what works:
- [hsext Swap]
- more than one [hsext]
- deleting some [hsext]

untested:
- chaining
```

```
- reentrancy

make && cd examples && pd -path .. Swap-help.pd

Edit Makefile if it doesn't work.


Tested on GNU/Linux/Gentoo x86/athlon-xp with:

ghc-6.4.2
hs-plugins-1.0-rc0
gcc-4.1.1
pd-0.39-2
(may be broken now...)

Tested on GNU/Linux/pure:dyne x86/pentium-m with:

ghc-6.6.1
hs-plugins-darcs
gcc-3.4.4
pd-0.40-2
(with haskell.dyne version 3, works now...)


(using DT_TEXTREL in a shared library is a huge hack and
won't work with some loaders on some architectures, you have
been warned)


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~ Contents ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  README    -- this file
  COPYING   -- legal

  Makefile  -- how to build

  api/      -- files required by [hsext] at runtime
  build/    -- place to store temporary files during build
  examples/ -- some examples
  hsext/    -- source code


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~ State Semantics ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~ CASE 0 ~ CONSTRUCTION ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  >-build->
                new
  <-rturn-<

~~~~ CASE 1 ~ SIMPLE INLET ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  >-inlet->
```

```
                take
                modify
                put
  <-rturn-<

~~~~ CASE 2a ~ COMPLEX INLET ~ SIMPLE ~~~~~~~~~~~~~~~~~~~~~~~~~~

  >-inlet->
                take
                modify
                put
                        >-outlet->
                                    |simple|
                        <-return-<
  <-rturn-<

~~~~ CASE 2b ~ COMPLEX INLET ~ REENTRANT ~~~~~~~~~~~~~~~~~~~~~~~

  >-inlet->
                take
                modify
                put (i)
                        >-outlet->
            |reentrant|
  >-inlet->
                take
                modify
                put (ii)
  <-rturn-<
            |reentrant|
                        <-return-<
                choose(*) one of
                  (i)  from scope
                  (ii) from state
  <-rturn-<

  (*) generally choose (i) for more outlets and (ii) for return


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~ Multiple Receive Names ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


irc.freenode.net/#dataflow 2007-03-05

(03:46:41) ClaudiusMaximus: matju: do you have any idea on
this? http://lists.puredata.info/pipermail/pd-dev/2007-03/008534.html
(i ask because you've read more of pd's source than i have ;) )
(03:48:27) matju: ClaudiusMaximus: pass the proxy as first
argument of pd_bind
(03:49:02) matju: ClaudiusMaximus: proxy inlets are not fake
objects, they're real
(03:49:30) ClaudiusMaximus: matju: oh wow, i didn't think it
would be so simple!
(03:50:34) ClaudiusMaximus: matju: i guess if i create the proxy
receiver with CLASS_NOINLET, that would do what i want
(03:50:56) matju: ClaudiusMaximus: actually pd confuses "object"
```

```
as in t_pd (in the OOP sense), and "object" in the t_text sense
(inletted, outletted, canvasable)
(03:51:22) ClaudiusMaximus: matju: i think that was what was
confusing me...
(03:51:49) matju: ClaudiusMaximus: i create the proxy with
CLASS_PD|CLASS_NOINLET
(03:52:50) ClaudiusMaximus: matju: ok, i'd have to check what
i'm creating my proxy inlets with at the moment, i have a
feeling it might be just 0
(03:53:00) ClaudiusMaximus: matju: thanks :)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
EOF
```

30 test/ffi/Bar.c

```
#include <stdio.h>
#include <HsFFI.h>

#include "Bar.h"

#ifdef __GLASGOW_HASKELL__
#include "Foo_stub.h"
#endif

#ifdef __GLASGOW_HASKELL__
extern void __stginit_Foo ( void );
#endif

int test(int argc, char *argv[])
{
  int i;

  hs_init(&argc, &argv);
#ifdef __GLASGOW_HASKELL__
  hs_add_root(__stginit_Foo);
#endif

  for (i = 0; i < 5; i++) {
    printf("%d\n", foo(2500));
  }

  hs_exit();
  return 0;
}
```

31 test/ffi/Barf.c

```
#include <stdio.h>
#include <HsFFI.h>

#include "Barf.h"

#ifdef __GLASGOW_HASKELL__
#include "Foof_stub.h"
#endif
```

```
#ifdef __GLASGOW_HASKELL__
extern void __stginit_Foof ( void );
#endif

int test(int argc, char *argv[])
{
  int i;

  hs_init(&argc, &argv);
#ifdef __GLASGOW_HASKELL__
  hs_add_root(__stginit_Foof);
#endif

  for (i = 0; i < 5; i++) {
    printf("%e\n", (double) foof(2500.0));
  }

  hs_exit();
  return 0;
}
```

32 test/ffi/Barf.h

```
extern int test(int argc, char *argv[]);
```

33 test/ffi/Bar.h

```
extern int test(int argc, char *argv[]);
```

34 test/ffi/Baz.c

```
#include "Bar.h"

#include <stdio.h>
#include <dlfcn.h>


int main(int argc, char *argv[]) {
    void *bar;
    int (*test)(int*,char**[]);
    fprintf(stderr, "Loading 'Bar.so'... ");
    bar = dlopen("./Bar.so", RTLD_NOW);
    if (!bar) {
        fprintf(stderr, "FAILED!\n");
        return 1;
    }
    fprintf(stderr, "OK!\n");
    fprintf(stderr, "Looking up 'test'... ");
    test = dlsym(bar, "test");
    if (!test) {
        fprintf(stderr, "FAILED!\n");
        dlclose(bar);
        return 2;
    }
    fprintf(stderr, "OK!\n");
```

```
    fprintf(stderr, "Running 'test'...\n");
    test(&argc, &argv);
    fprintf(stderr, "Test completed!\n");
    dlclose(bar);
    return 0;
}
```

35 test/ffi/Bazf.c

```
#include "Barf.h"

#include <stdio.h>
#include <dlfcn.h>


int main(int argc, char *argv[]) {
    void *bar;
    int (*test)(int*,char**[]);
    fprintf(stderr, "Loading 'Barf.so'... ");
    bar = dlopen("./Barf.so", RTLD_NOW);
    if (!bar) {
        fprintf(stderr, "FAILED!\n");
        return 1;
    }
    fprintf(stderr, "OK!\n");
    fprintf(stderr, "Looking up 'test'... ");
    test = dlsym(bar, "test");
    if (!test) {
        fprintf(stderr, "FAILED!\n");
        dlclose(bar);
        return 2;
    }
    fprintf(stderr, "OK!\n");
    fprintf(stderr, "Running 'test'...\n");
    test(&argc, &argv);
    fprintf(stderr, "Test completed!\n");
    dlclose(bar);
    return 0;
}
```

36 test/ffi/Foof.hs

```
module Foof where

foreign export ccall foof :: Float -> IO Float

foof :: Float -> IO Float
foof n = return (fromIntegral (length (f n)))

f :: Float -> [Float]
f n
  | n<1       = []
  | otherwise = n:(f (n-1))
```

37 test/ffi/Foo.hs

```
module Foo where

foreign export ccall foo :: Int -> IO Int

foo :: Int -> IO Int
foo n = return (length (f n))

f :: Int -> [Int]
f 0 = []
f n = n:(f (n-1))
```

38 test/ffi/Makefile

```
all: Baz Bar.so Bazf Barf.so

clean:
	rm -f Baz Bar.so Bar.o Foo.o Foo_stub.o Foo_stub.c Foo_stub.h Foo.hi
	rm -f Bazf Barf.so Barf.o Foof.o Foof_stub.o Foof_stub.c Foof_stub.h ↙
	    ↳ Foof.hi


Baz: Baz.c Bar.h
	gcc -o Baz Baz.c -ldl

Bar.so: Bar.o Foo.o Foo_stub.o
	gcc -shared -o Bar.so -DDONT_WANT_WIN32_DLL_SUPPORT Bar.o Foo.o Foo_stub↙
	    ↳ .o \
-L/usr/lib/ghc-6.4.2 -lHShaskell98 -lHSbase -lHSbase_cbits -lHSrts -lm -lgmp -↙
    ↳ ldl -u GHCziBase_Izh_static_info -u GHCziBase_Czh_static_info -u ↙
    ↳ GHCziFloat_Fzh_static_info -u GHCziFloat_Dzh_static_info -u ↙
    ↳ GHCziPtr_Ptr_static_info -u GHCziWord_Wzh_static_info -u ↙
    ↳ GHCziInt_I8zh_static_info -u GHCziInt_I16zh_static_info -u ↙
    ↳ GHCziInt_I32zh_static_info -u GHCziInt_I64zh_static_info -u ↙
    ↳ GHCziWord_W8zh_static_info -u GHCziWord_W16zh_static_info -u ↙
    ↳ GHCziWord_W32zh_static_info -u GHCziWord_W64zh_static_info -u ↙
    ↳ GHCziStable_StablePtr_static_info -u GHCziBase_Izh_con_info -u ↙
    ↳ GHCziBase_Czh_con_info -u GHCziFloat_Fzh_con_info -u ↙
    ↳ GHCziFloat_Dzh_con_info -u GHCziPtr_Ptr_con_info -u ↙
    ↳ GHCziPtr_FunPtr_con_info -u GHCziStable_StablePtr_con_info -u ↙
    ↳ GHCziBase_False_closure -u GHCziBase_True_closure -u ↙
    ↳ GHCziPack_unpackCString_closure -u GHCziIOBase_stackOverflow_closure -u ↙
    ↳ GHCziIOBase_heapOverflow_closure -u GHCziIOBase_NonTermination_closure -u ↙
    ↳ GHCziIOBase_BlockedOnDeadMVar_closure -u ↙
    ↳ GHCziIOBase_BlockedIndefinitely_closure -u GHCziIOBase_Deadlock_closure -u↙
    ↳  GHCziIOBase_NestedAtomically_closure -u ↙
    ↳ GHCziWeak_runFinalizzerBatch_closure -u __stginit_Prelude

Bar.o: Bar.c Bar.h Foo_stub.h
	gcc -I/usr/lib/ghc-6.4.2/include -D__GLASGOW_HASKELL__ -c Bar.c

Foo.o Foo_stub.o Foo_stub.h: Foo.hs
	ghc -fffi -o Foo.o -c Foo.hs


Bazf: Bazf.c Barf.h
	gcc -o Bazf Bazf.c -ldl
```

```
Barf.so: Barf.o Foof.o Foof_stub.o
	gcc -shared -o Barf.so -DDONT_WANT_WIN32_DLL_SUPPORT Barf.o Foof.o ↙
	    ↳ Foof_stub.o \
-L/usr/lib/ghc-6.4.2 -lHShaskell98 -lHSbase -lHSbase_cbits -lHSrts -lm -lgmp -↙
    ↳ ldl -u GHCziBase_Izh_static_info -u GHCziBase_Czh_static_info -u ↙
    ↳ GHCziFloat_Fzh_static_info -u GHCziFloat_Dzh_static_info -u ↙
    ↳ GHCziPtr_Ptr_static_info -u GHCziWord_Wzh_static_info -u ↙
    ↳ GHCziInt_I8zh_static_info -u GHCziInt_I16zh_static_info -u ↙
    ↳ GHCziInt_I32zh_static_info -u GHCziInt_I64zh_static_info -u ↙
    ↳ GHCziWord_W8zh_static_info -u GHCziWord_W16zh_static_info -u ↙
    ↳ GHCziWord_W32zh_static_info -u GHCziWord_W64zh_static_info -u ↙
    ↳ GHCziStable_StablePtr_static_info -u GHCziBase_Izh_con_info -u ↙
    ↳ GHCziBase_Czh_con_info -u GHCziFloat_Fzh_con_info -u ↙
    ↳ GHCziFloat_Dzh_con_info -u GHCziPtr_Ptr_con_info -u ↙
    ↳ GHCziPtr_FunPtr_con_info -u GHCziStable_StablePtr_con_info -u ↙
    ↳ GHCziBase_False_closure -u GHCziBase_True_closure -u ↙
    ↳ GHCziPack_unpackCString_closure -u GHCziIOBase_stackOverflow_closure -u ↙
    ↳ GHCziIOBase_heapOverflow_closure -u GHCziIOBase_NonTermination_closure -u ↙
    ↳ GHCziIOBase_BlockedOnDeadMVar_closure -u ↙
    ↳ GHCziIOBase_BlockedIndefinitely_closure -u GHCziIOBase_Deadlock_closure -u↙
    ↳  GHCziIOBase_NestedAtomically_closure -u ↙
    ↳ GHCziWeak_runFinalizzerBatch_closure -u __stginit_Prelude

Barf.o: Barf.c Barf.h Foof_stub.h
	gcc -I/usr/lib/ghc-6.4.2/include -D__GLASGOW_HASKELL__ -c Barf.c

Foof.o Foof_stub.o Foof_stub.h: Foof.hs
	ghc -fffi -o Foof.o -c Foof.hs
```

39 test/PluginTest.hs

```
import System.Plugins
import PureData

src     = "../examples/Example1.hs"
apipath = "../api"

main = do status <- make src ["-i"++apipath]
          case status of
                MakeSuccess _ _ -> f
                MakeFailure e -> mapM_ putStrLn e

  where f = do v <- pdynload "../examples/Example1.o" [apipath] [] "PureData.↙
      ↳ Class" "setup"
               case v of
                  LoadSuccess _ a  -> putStrLn "loaded .. yay!"
                  _                -> putStrLn "wrong types"
```

40 test/TypeableTest.hs

```
{-# OPTIONS_GHC -fglasgow-exts #-}

module TypeableTest ( Foo(FooFloat, FooString) ) where

import Data.Typeable
```

```
data Foo = FooFloat Float | FooString String
  deriving (Read, Show, Eq, Typeable)
```

41 test/TypeableTestUser.hs

```
module TypeableTestUser ( main ) where

import Data.Typeable
import TypeableTest

main = print . show $ typeOf (FooFloat 1.4)
```