KANT/KASH SCSCP Package

www.math.tu-berlin.de/~kant/kantscscp.html

Fourth RISC/SCIEnce training school – 2009

LESSENI SYLLA

TU Berlin - Fakultät II

Institut für Mathematik

Straße des 17. Juni 136

D-10623 Berlin, Germany

lesseni@math.tu-berlin.de

the SCIEnce project is supported by the EU FP6 Programme

Talk outline

Talk outline

- Overview of the SCIEnce project

Talk outline

- Overview of the SCIEnce project
- Introduction to KANT SCSCP

Talk outline

- Overview of the SCIEnce project
- Introduction to KANT SCSCP
- First steps

Talk outline

- Overview of the SCIEnce project
- Introduction to KANT SCSCP
- First steps
- KANT SCSCP client shell: kapy

Talk outline

- Overview of the SCIEnce project
- Introduction to KANT SCSCP
- First steps
- KANT SCSCP client shell: kapy
- kapy - Future Developments

Overview of the SCIEnce project

Overview of the SCIEnce project

- Brief description of the SCIEnce project

Overview of the SCIEnce project

- Brief description of the SCIEnce project
- What is SCSCP?

Overview of the SCIEnce project

- Brief description of the SCIEnce project
- What is SCSCP?
- The key features of SCSCP

Brief description of SCIEnce

SCIEnce stands for *Symbolic Computation Infrastructure for Europe*.
It brings together the developers of GAP, KANT, MAPLE, MuPAD and RISC. The project aims to promote the development of new software that is:

http:// www.symbolic-computation.org

Brief description of SCIEnce

SCIEnce stands for *Symbolic Computation Infrastructure for Europe*.
It brings together the developers of GAP, KANT, MAPLE, MuPAD and RISC. The project aims to promote the development of new software that is:

- made more efficient by sharing components and expertise;

http:// www.symbolic-computation.org

Brief description of SCIEnce

SCIEnce stands for *Symbolic Computation Infrastructure for Europe*.
It brings together the developers of GAP, KANT, MAPLE, MuPAD and RISC. The project aims to promote the development of new software that is:

- made more efficient by sharing components and expertise;
- made more interoperable in the modern Web services environment;

http:// www.symbolic-computation.org

Brief description of SCIEnce

SCIEnce stands for *Symbolic Computation Infrastructure for Europe*.
It brings together the developers of GAP, KANT, MAPLE, MuPAD and RISC. The project aims to promote the development of new software that is:

- made more efficient by sharing components and expertise;
- made more interoperable in the modern Web services environment;
- ready for the coming environment of Grid computing.

http:// www.symbolic-computation.org

What is SCSCP?

- SCSCP stands for *Symbolic Computation Software Composability Protocol*.
  It is sockets-based RPC protocol.
  It is a protocol by which a Computer Algebra System (CAS) may offer services and a client may employ them. The client can be another instance of the same CAS (in parallel computing context), a Grid middleware, a Web server or another CAS running on the same computer system or remotely.
  NB: Another implementation of the protocol is a proper Web service using standard SOAP/HTTP wrappings for SCSCP messages.

http://www.symbolic-computation.org/science/Scscp

The main features of SCSCP:

Communications take place using port 26133, reserved for SCSCP by the Internet Assigned Numbers Authority.
NB: OpenMath is an XML-based standard for representing mathematical objects with their semantics.

The main features of SCSCP:

- Mathematical data are encoded in OpenMath;

Communications take place using port 26133, reserved for SCSCP by the Internet Assigned Numbers Authority.
NB: OpenMath is an XML-based standard for representing mathematical objects with their semantics.

The main features of SCSCP:

- Mathematical data are encoded in OpenMath;
- The protocol messages are represented as OpenMath objects using the OpenMath Content Dictionaries scscp1 and scscp2, so that participating systems need to support only one language;

Communications take place using port 26133, reserved for SCSCP by the Internet Assigned Numbers Authority.
NB: OpenMath is an XML-based standard for representing mathematical objects with their semantics.

The main features of SCSCP:

- Mathematical data are encoded in OpenMath;
- The protocol messages are represented as OpenMath objects using the OpenMath Content Dictionaries scscp1 and scscp2, so that participating systems need to support only one language;
- the OpenMath support is wired directly into the joining systems. This is more efficient than the usual practice of producing wrapping programs to enable OpenMath support.

Communications take place using port 26133, reserved for SCSCP by the Internet Assigned Numbers Authority.
NB: OpenMath is an XML-based standard for representing mathematical objects with their semantics.

Basic OpenMath objects are:
integers, floats, strings, byte arrays, variables, symbols.
The OpenMath symbols consist of a name and a reference to the Content Dictionary containing this name.

For further details, see: *http://www.openmath.org/*

Introduction to KANT SCSCP

Introduction to KANT SCSCP

- Starting and Leaving KANT SCSCP

Introduction to KANT SCSCP

- Starting and Leaving KANT SCSCP
- KANT SCSCP server help system

Starting and Leaving KANT SCSCP

KANT SCSCP package, `autokash-3.1`, is freely available for Linux/x86 from:

http://www.math.tu-berlin.de/~kant/kantscscp.html

- README file contains general informations about the libraries to install

Starting and Leaving KANT SCSCP

KANT SCSCP package, `autokash-3.1`, is freely available for Linux/x86 from:

http://www.math.tu-berlin.de/~kant/kantscscp.html

- README file contains general informations about the libraries to install
- download the tarball autokash-3.1.tar.bz2

Starting and Leaving KANT SCSCP

KANT SCSCP package, `autokash-3.1`, is freely available for Linux/x86 from:

http://www.math.tu-berlin.de/~kant/kantscscp.html

- README file contains general informations about the libraries to install
- download the tarball autokash-3.1.tar.bz2
- tar xjf autokash-3.1.tar.bz2

Starting and Leaving KANT SCSCP

KANT SCSCP package, `autokash-3.1`, is freely available for Linux/x86 from:

http://www.math.tu-berlin.de/~kant/kantscscp.html

- README file contains general informations about the libraries to install
- download the tarball autokash-3.1.tar.bz2
- tar xjf autokash-3.1.tar.bz2
- cd autokash-3.1

Starting and Leaving KANT SCSCP

KANT SCSCP package, `autokash-3.1`, is freely available for Linux/x86 from:

http://www.math.tu-berlin.de/~kant/kantscscp.html

- README file contains general informations about the libraries to install
- download the tarball autokash-3.1.tar.bz2
- tar xjf autokash-3.1.tar.bz2
- cd autokash-3.1
- Start the KANT SCSCP server by simply typing:
  `bin/kashd lib/kant/openmath.la &`
  NB: the main binary is `kashd`, the KANT daemon.

Starting and Leaving KANT SCSCP

KANT SCSCP package, `autokash-3.1`, is freely available for Linux/x86 from:

http://www.math.tu-berlin.de/~kant/kantscscp.html

- README file contains general informations about the libraries to install
- download the tarball autokash-3.1.tar.bz2
- tar xjf autokash-3.1.tar.bz2
- cd autokash-3.1
- Start the KANT SCSCP server by simply typing: `bin/kashd lib/kant/openmath.la &`
  NB: the main binary is `kashd`, the KANT daemon.
- Leave the SCSCP server by pressing: `Ctrl C`

KANT SCSCP server help system

We can display the help by typing:

KANT SCSCP server help system

We can display the help by typing:

- `bin/kashd --help`

KANT SCSCP server help system

We can display the help by typing:

- `bin/kashd --help`

Usage: kashd [options] [module...]
Input Options
−l, −−−listen=spec listen on input according to SPEC, SPEC is a socat-like string, supported specs are: tcp4:<bind-address>:<port>, tcp6:<bind-address>:<port>, unix:<socket-file>. (default: "tcp6:[::]:26133")
Language Options -z, −−−no-namespaces dispatch the result objects without namespaces
Help options: -?, −−−help Show this help message −−−usage Display brief usage message

First steps

- Running the KANT SCSCP server:
  `bin/kashd lib/kant/openmath.la &`
  This command will also load the OpenMath module.

First steps

- Running the KANT SCSCP server:
  `bin/kashd lib/kant/openmath.la &`
  This command will also load the OpenMath module.
- Connection to the running server:
  `telnet hostname port`
  Example:
  `telnet issel.math.tu-berlin.de 26133`

First steps

- Running the KANT SCSCP server:
  `bin/kashd lib/kant/openmath.la &`
  This command will also load the OpenMath module.
- Connection to the running server:
  `telnet hostname port`
  Example:
  `telnet issel.math.tu-berlin.de 26133`
- `<?scscp version="1.3" ?>`

First steps

- Running the KANT SCSCP server:
  `bin/kashd lib/kant/openmath.la &`
  This command will also load the OpenMath module.
- Connection to the running server:
  `telnet hostname port`
  Example:
  `telnet issel.math.tu-berlin.de 26133`
- `<?scscp version="1.3" ?>`
- OpenMath objects are in some transaction blocks which have the form:
  `<?scscp start ?>`
  [a valid openMath object]
  `<?scscp end ?>`

or could have the form:

```
<?scscp start ?>
```
[something]
```
<?scscp cancel ?>
```

which indicates that the started transaction block is cancelled.

NB:
We are running a KANT SCSCP server in Berlin accessible at `issel.math.tu-berlin.de`, port 26133

Example

```
<?scscp start ?>
<OMOBJ><OMATTR>
<OMATP><OMS cd="scscp1" name="call_id"/>
<OMSTR>symcomp.org:26133:Sylla</OMSTR>
<OMS cd="scscp1" name="option_return_obj
<OMSTR></OMSTR> </OMATP> <OMA>
<OMS cd="scscp1" name="procedure_call"/>
<OMA><OMS cd="arith1" name="plus"/>
<OMI>1</OMI><OMI>2</OMI></OMA></OMA>
</OMATTR></OMOBJ>
<?scscp end ?>
```

To leave the SCSCP session (each side may close the socket):

```
<?scscp quit ?>
```

KANT SCSCP client shell: kapy

KANT SCSCP client shell: kapy

- The kapy use instructions

KANT SCSCP client shell: kapy

- The kapy use instructions
- The OpenMath tags constructors

KANT SCSCP client shell: kapy

- The kapy use instructions
- The OpenMath tags constructors
- Examples

KANT SCSCP client shell: kapy

- The kapy use instructions
- The OpenMath tags constructors
- Examples
- The function PyConvert

The kapy use instructions

We are developing a KANT SCSCP client shell, named kapy. The idea is to ease manual typing when handling OpenMath objects and also to be able to convert them to their python representation.

The use instructions:

- download and unpack the tarball kapy-1.0.1.tar.bz2

The kapy use instructions

We are developing a KANT SCSCP client shell, named kapy. The idea is to ease manual typing when handling OpenMath objects and also to be able to convert them to their python representation.

The use instructions:

- download and unpack the tarball kapy-1.0.1.tar.bz2
- `cd pysh`

The kapy use instructions

We are developing a KANT SCSCP client shell, named kapy. The idea is to ease manual typing when handling OpenMath objects and also to be able to convert them to their python representation.

The use instructions:

- download and unpack the tarball kapy-1.0.1.tar.bz2
- `cd pysh`
- `python`

The kapy use instructions

We are developing a KANT SCSCP client shell, named kapy. The idea is to ease manual typing when handling OpenMath objects and also to be able to convert them to their python representation.

The use instructions:

- download and unpack the tarball kapy-1.0.1.tar.bz2
- `cd pysh`
- `python`
- `»import kapy`

The kapy use instructions

We are developing a KANT SCSCP client shell, named kapy. The idea is to ease manual typing when handling OpenMath objects and also to be able to convert them to their python representation.

The use instructions:

- download and unpack the tarball kapy-1.0.1.tar.bz2
- `cd pysh`
- `python`
- `»import kapy`
- `»cas=kapy.connect("hostname",port);`

The kapy use instructions

We are developing a KANT SCSCP client shell, named kapy. The idea is to ease manual typing when handling OpenMath objects and also to be able to convert them to their python representation.

The use instructions:

- download and unpack the tarball kapy-1.0.1.tar.bz2
- `cd pysh`
- `python`
- `»import kapy`
- `»cas=kapy.connect("hostname",port);`
  - Example: The hostname could be: `issel.math.tu-berlin.de` with port 26133

The OpenMath tags constructors

- `»kapy.omi(int);` for the integers

The OpenMath tags constructors

- `»kapy.omi(int);` for the integers
- `»kapy.omf(float);` for the floats

The OpenMath tags constructors

- `»kapy.omi(int);` for the integers
- `»kapy.omf(float);` for the floats
- `»kapy.omstr("str");` for the strings

The OpenMath tags constructors

- `»kapy.omi(int);` for the integers
- `»kapy.omf(float);` for the floats
- `»kapy.omstr("str");` for the strings
- `»kapy.omv("var");` for the variables

The OpenMath tags constructors

- `»kapy.omi(int);` for the integers
- `»kapy.omf(float);` for the floats
- `»kapy.omstr("str");` for the strings
- `»kapy.omv("var");` for the variables
- `»kapy.oms("cdname", "symbolname");` for the OMS

The OpenMath tags constructors

- `»kapy.omi(int);` for the integers
- `»kapy.omf(float);` for the floats
- `»kapy.omstr("str");` for the strings
- `»kapy.omv("var");` for the variables
- `»kapy.oms("cdname", "symbolname");` for the OMS
- `» kapy.oma("cdname","symbolname",args);` for the OMA

Examples

- Example 1

```
»A=kapy.omi(23);
»print A;
»cas.compute(A);
»B=kapy.omf(5.7);
»print B;
»kapy.compute(B);
```

Examples

- Example 1

```
»A=kapy.omi(23);
»print A;
»cas.compute(A);
»B=kapy.omf(5.7);
»print B;
»kapy.compute(B);
```

- Example 2

```
»C=kapy.oms("nums1", "pi");
»print C;
»cas.compute(C);
»D=kapy.oma("arith1", "plus", 1, 2);
»print D;
»cas.compute(D);
```

The function PyConvert

Since kapy is written in python, it is natural to be able to convert basic OpenMath objects to their python representation using the function PyConvert.

Example

```
»E=kapy.oma("list1","list",4,"yes",9.4);
»print E;
»F=cas.compute(E);
»kapy.PyConvert(F);
```

kapy - Future Developments

kapy is freely available from:
http://www.math.tu-berlin.de/~kant/kantscscp.html

If you have any suggestions on how to improve the kapy development, please feel free to contact us at:
`kant@math.tu-berlin.de`

kapy - Future Developments

- Get access to all the functionality of KANT/KASH

kapy is freely available from:
http://www.math.tu-berlin.de/~kant/kantscscp.html

If you have any suggestions on how to improve the kapy development, please feel free to contact us at:
`kant@math.tu-berlin.de`

kapy - Future Developments

- Get access to all the functionality of KANT/KASH
- Ease the conversion of the KANT objects into OpenMath objects

kapy is freely available from:
http://www.math.tu-berlin.de/~kant/kantscscp.html

If you have any suggestions on how to improve the kapy development, please feel free to contact us at:
`kant@math.tu-berlin.de`

kapy - Future Developments

- Get access to all the functionality of KANT/KASH
- Ease the conversion of the KANT objects into OpenMath objects
- Be able to convert the OpenMath objects to their python representation and vice versa.

kapy is freely available from:
http://www.math.tu-berlin.de/~kant/kantscscp.html

If you have any suggestions on how to improve the kapy development, please feel free to contact us at:
`kant@math.tu-berlin.de`