SIMON Source Code

1. ByteStore.h 2
2. ByteStore.cxx 3
3. fill-ss-with-hotlist 5
4. functions.h 6
5. functions.cxx 6
6. Makefile 7
7. Makefile.binary 10
8. Makefile.source 12
9. README 14
10. README.upgrading_to_0.4 15
11. send-simon 15
12. simon.bit 15
13. simon.cxx 16
14. simple-simon.tcl 21
15. smosaic 32
16. smosaic.skel 33
17. ss.h 33
18. ss.cxx 38
19. ssbrowser.cxx 48
20. ssci-lang.h 51
21. ssci.cxx 52
22. ssci.l 57
23. ssdbm-tcl.h 58
24. ssdbm-tcl.cxx 58
25. ssdbm.h 62
26. ssdbm.cxx 65
27. ssindexer.cxx 77
28. test-ss.dump 78
29. text-dump-0.3-format-ss.cxx 79
30. upgrade-from-0.5-to-0.6 79

1 ByteStore.h

```
//This class is a dynamic byte box.  You can put things in & get them out,
//but you must do it in the same order, as it doen't store types, just bytes.
//It uses no padding, and expects C char* to be null terminated strings.
//You can use it several times by emptying it.
//It can 'take over' a chunk of memory, and volenteer itself as memory, by
//using take_over and pointer.
//It will ALWAYS try to deallocate the memory it uses with free, when its
//destructor is called, so it should only take over memory that it can free,
//and anthing that uses it's memory should not free it (or mess with it in
//any way except looking).
//Define your own version by public inheritance of this class, add functions
//to store and get aplication specific objects.  Note that you can't overload
//functions in this class with your own versions of get & store, as the  versions
//in the derived class will hide the ones here, but thats OK, just write fns
//that call these fns like this ByteStore::store( <stuff> ).

//define _ByteStore_platform_dependant when writting code to work
//with data saved using old versions of Bytestore that work with platform
//dependant code

#ifndef _ByteStore_h
#define _ByteStore_h 1



#include <stdlib.h>
#include <String.h>

class ByteStore
  {
  public:
    ByteStore()
                {start=NULL; expand_by=0; size=0;
                 next_get_index=0; next_free_index=0;};
    ByteStore(int start_size)
                {start=(char*)malloc(start_size); expand_by=0;  size=start_size;
                 next_get_index=0; next_free_index=0;};
    ByteStore(int start_size, int expand_by_)
                {start=(char*)malloc(start_size);  expand_by=expand_by_;
                 size=start_size;
                 next_get_index=0; next_free_index=0;};
    ~ByteStore();

    //returns address for OK, or neg for error
    int store(const char ch);
    int store(const char* ch_p);
    // this stores 32 leaste significant bits of an int,  reguardless of
    // sizeof(int), unless _ByteStore_platform_dependant is  defined
    int store(const int int_n);
    int store(const String& string);

    void replace(const int int_n, const int address);

    int resize(int required_bytes);
    void next_get_is_first() {next_get_index=0;};

    void get(char& ch);
    void get(char* ch_p);
    void get(int& int_n);
    void get(String& string);

    char get_char();
    int get_int();
    char* get_char_ptr();

    int allocated_size() {return size;};
    int used_size() {return next_free_index;};
    void take_over (char* ch_p, int the_size)
```

```
                {if (start!=NULL) free(start); start=ch_p;  size=the_size;
                 next_free_index=the_size; next_get_index=0;};
    char* pointer() {return start;};
    void empty() {next_free_index=0;};
    void get_from_begining() {next_get_index=0;};
    int next_get_address() {return next_get_index;};

  protected:
    int size;
    char* start;
    int expand_by;
    int next_free_index;
    int next_get_index;
    int err;
  };


#endif
```

2 ByteStore.cxx

```
#include "ByteStore.h"


int ByteStore::store(const String& string)
  {
  int str_len = string.length();
  err = resize(str_len+1);
  if (err==0)
    {
    err=next_free_index;
    for (int n=0; n<str_len; n++)
                *(start+(next_free_index++)) = string.elem(n);
    *(start+(next_free_index++)) = '\0';  //add null!
    };
  return err;
  };

int ByteStore::store(const char ch)
  {
  err = resize(1);
  if (err==0)
    {
    err=next_free_index;
    *(start+(next_free_index++)) = ch;
    };
  return err;
  }

int ByteStore::store(const char* ch_p)
  {
  int str_len = strlen(ch_p)+1;
  err = resize(str_len);
  if (err==0)
    {
    err=next_free_index;
    for (int n=0; n<str_len; n++)
                {*(start+(next_free_index++)) = *(ch_p+n);};
    };
  return err;
  };

#ifdef _ByteStore_platform_dependant
int ByteStore::store(const int int_n)
  {
  err = resize(sizeof(int));
  if (err==0)
```

```
    {
    err=next_free_index;
    const char* int_p = (char*) &int_n;
    for (int n=1; n<=sizeof(int); n++)
      {
      *(start+(next_free_index++)) = *(int_p++);
      };
    };
  return err;
  };
#else
int ByteStore::store(const int int_n)
  {
  err = resize(4);
  if (err==0)
    {
    err=next_free_index;
    int remainder, n, divisor;
    divisor = 256;
    n=int_n;
    for (int a=1; a<=4; a++)
      {
      remainder=n%divisor;
      n=n/divisor;
      store((char)remainder);
      };
    };
  return err;
  };

#endif

void ByteStore::replace(const int int_n, const int address)
  {
  int temp=next_free_index;
  next_free_index=address;
  err=store(int_n);
  next_free_index=temp;
  };


void ByteStore::get(char& ch)
  {
  ch = *(start+(next_get_index++));
  };

void ByteStore::get(char* ch_p)
  {
  strcpy (ch_p, (start+next_get_index));
  next_get_index += (strlen (ch_p)+1);
  };

void ByteStore::get(String& string)
  {
  string = (start+next_get_index);
  next_get_index += (string.length()+1);
  };

#ifdef _ByteStore_platform_dependant
void ByteStore::get(int& int_n)
  {
//cout << "old get method ";
  char* int_p = (char*) &int_n;
  for (int n=1; n<=sizeof(int); n++)
    {
    *(int_p++) = *(start+(next_get_index++));
    };
  };
#else
void ByteStore::get(int& int_n)
  {
//cout << "new get method ";
  int remainder, multiplier;
  multiplier = 1;
```

```
  int_n=0;
  for (int a=1; a<=4; a++)
    {
    int_n=int_n+((unsigned char)(get_char())*multiplier);
    multiplier=multiplier*256;
    };
//cout << "get int :: "<<int_n<<"\n";
  };
#endif

char ByteStore::get_char()
  {
  char ch = *(start+(next_get_index++));
  return ch;
  };

int ByteStore::get_int()
  {
  int int_n;
  get(int_n);
  return int_n;
  };

char* ByteStore::get_char_ptr()
  {
  char* ch_p = (start+next_get_index);
  next_get_index += (strlen (ch_p)+1);
  return ch_p;
  };

ByteStore::~ByteStore()
  {
  free(start);
  };


int ByteStore::resize(int required_bytes)
    {
    if (required_bytes>size-next_free_index)

                {
                size = required_bytes+expand_by+size;
                start=(char*) realloc(start, size);
                };

    if (start==NULL) return -1;
                else return 0;
    }; //end resize
```

3 fill-ss-with-hotlist

```
#! /bin/sh

if [ $# -eq 3 ]
then

  HOTLIST=$1
  SS=$2
  SUBJECT=$3

  #open a ssci commands file, make sure the subject exists
  echo newsubject \"$SUBJECT\" > $$.tmp
```

```
  #open hotlist, bind to standard in & read past first two lines
  exec <$HOTLIST
  read null
  read null

  while read LINE1
     do
     read LINE2
     TITLE=$LINE2
     set $LINE1
     URL=$1
     echo newresource \"$URL\" \"$TITLE\" \"$SUBJECT\" >> $$.tmp
     done

  #run the commands on ssci
  ssci $SS < $$.tmp
  rm $$.tmp

else
  echo need three args:
  echo   _hotlist file_ _name of subject space_ _subject to fill_
  echo  eg fill-ss-with-hotlist ~/.mosaic-hotlist-default test-ss none
fi
```

4 functions.h

```
// usefull functions

#ifndef _functions_h
#define _functions_h 1

#include <String.h>

String simplified(const String& in);
void simplify(String& st);

String quoted(const String& in);
void quote(String& st);

String unquoted(const String& in);
void unquote(String& st);

unsigned long factorial(unsigned n);

unsigned long factorial_div_factorial(unsigned n, unsigned m);

unsigned nCr(unsigned n, unsigned r);

#endif
```

5 functions.cxx

```
// usefull functions

#include <String.h>
#include <functions.h>

String simplified(const String& in)
  {
```

```
  String out;
  char ch;
  for (int n=0; n<in.length(); n++)
    {
    ch = in.elem(n);
    if (    (ch>='a')&&(ch<='z')
                  || (ch>='A')&&(ch<='Z')
                  || (ch>='0')&&(ch<='9') )
      out += ch;
    }; // end for
  out.downcase();
  return out;
  };

void simplify(String& st)
  {
  st=simplified(st);
  };


String quoted(const String& in)
  {
  String out;
  cat ("",in,out);
  cat (out,"",out);
  return out;
  };

void quote(String& st)
  {
  cat ("",st,st);
  cat (st,"",st);
  };


String unquoted(const String& in)
  {
  String out;
  out=in;
  out.gsub("","");
  return out;
  };

void unquote(String& st)
  {
  st.gsub("","");
  };

unsigned long factorial(unsigned n)
  {if ((n==1) || (n==0)) return 1; else return (n*factorial(n-1));};

unsigned long factorial_div_factorial(unsigned n, unsigned m)
  {
  unsigned r=1;
  for (unsigned a=n; a>m; a--)
    r=r*a;
  return r;
  };

unsigned nCr(unsigned n, unsigned r)
  {return factorial_div_factorial(n,n-r)/factorial(r);};
```

6 Makefile

```
# YOU MUST EDIT THE VARIABLES BELOW.
# You should be able to get a working version without editing anything else.
# If you just want to build a binary, use make bintar.
# If you are using a binary distribution, you only need to edit MOSAIC =  mosaic,
# then do make smosaic.  You may want to change DEFAULT_SS = to
```

```
#    --- HOW TO USE THIS MAKEFILE ---
#
# make        - make binaries, use existing tcl library dir
#
# make bintar  - makes a binary tar package that is independant of gdbm,
#              tcl & tk, use it to make a binary package for others to use.
#
# make install - makes and installs simon.
#              This will make the install dir, copy everything you need to it,
#              to make simon independant of gdbm, tcl and tk.  It can also
#              makes a symbolic link to smosaic in your bin dir.
#
# make smosaic - (re)makes smosaic script, for when you change your setup.
#              Note that you may want:
#                   TCL_TCL_LIBRARY_DIR=tcl-lib
#                   TK_TK_LIBRARY_DIR=tk-lib
#              When you do this.
#
# make clean   - deletes binaries & object files
#
# make sourcetar - just for me!


# ------------------- EDIT THESE FOR YOUR SYSTEM ------------------------
# Please put full path names in, I don't think that ~ will work

# where object libs (eg libtcl.a) are
TCL_LIB_DIR = ../tcl
TK_LIB_DIR = ../tk
GDBM_LIB_DIR = ../gdbm

#where include files (eg tcl.h) are,
TCL_INCLUDE_DIR = ../tcl
TK_INCLUDE_DIR = ../tk
GDBM_INCLUDE_DIR = ../gdbm

#where to install simon files upon Make install MUST BE FULL NON RELATIVE PATH
# ie no .. . or ~
INSTALL_DIR = /users/student/undergrad/zgee9119/simon

#where special tcl and tk files (eg init.tcl) are
TCL_TCL_LIBRARY_DIR = $(TCL_LIB_DIR)/library
TK_TK_LIBRARY_DIR = $(TK_LIB_DIR)/library

#where to put a symbolic link to smosaic, so it will be in the PATH
#upon Make install.
BIN_DIR = /users/student/undergrad/zgee9119/bin

#command used to start mosaic 2.2 or later
MOSAIC = mosaic

#default subject space to use  (files are (eg) .my_ss.record .my_ss.subject)
#don't put a path front of it
DEFAULT_SS = /users/student/undergrad/zgee9119/.my-ss

#the compiler, should be gnu c++ compiler, g++ or gcc
CC = g++

#extra c flags, eg -g for debugging
EXTRA_CFLAGS = -W -O

#name for binary package, should be hardware & operating system specific,
#but it is just used to give the tar file a name, so be as specific as you can,
#but don't worry about it, eg  sparc-sunos4.1
BIN_VERSION = unknown


# ----------------- no edits below this line (I hope!) -----------------

VERSION=0.7

BINARIES = ssci simon simple-simon.tcl \
```

```
	README README.upgrading_to_0.4 smosaic.skel \
	simon.bit send-simon text-dump-0.3-format-ss test-ss.dump

CFLAGS = -L$(TCL_LIB_DIR) -L$(TK_LIB_DIR) -L$(GDBM_LIB_DIR) \
    -I$(TCL_INCLUDE_DIR) -I$(TK_INCLUDE_DIR) I$(GDBM_INCLUDE_DIR) \
    -I. $(EXTRA_CFLAGS)


all : ssci simon ssbrowser ssindexer text-dump-0.3-format-ss test-ss smosaic;


install: ssci simon;
    -mkdir -p $(INSTALL_DIR)
    -mkdir $(INSTALL_DIR)/tcl-lib
    -mkdir $(INSTALL_DIR)/tk-lib
    -cp $(TCL_TCL_LIBRARY_DIR)/* $(INSTALL_DIR)/tcl-lib
    -cp $(TK_TK_LIBRARY_DIR)/* $(INSTALL_DIR)/tk-lib
    -cp $(BINARIES) $(INSTALL_DIR)
    -cp Makefile.binary $(INSTALL_DIR)/Makefile
    -cat smosaic.skel | sed  -e 's?_TCL_LIB_?$(INSTALL_DIR)/tcllib?g' \
                             -e  's?_TK_LIB_?$(INSTALL_DIR)/tk-lib?g' \
                             -e  's?_MOSAIC_?$(MOSAIC)?g' \
                             -e  's?_DEFAULT_SS_?$(DEFAULT_SS)?g' \
                             -e  's?_PATH_?$(INSTALL_DIR)?g' \
                                                 > $(INSTALL_DIR)/smosaic
    chmod u+x $(INSTALL_DIR)/smosaic
    ln -s $(INSTALL_DIR)/smosaic $(BIN_DIR)/smosaic


simon.alpha.binary.$(BIN_VERSION).tar: bintar;

clean : ;
    rm -f *.o ssci simon smosaic *.tar

sourcetar : ;
    -rm -fr simon.$(VERSION)
    mkdir simon.$(VERSION)
    cp $(BINARIES) simon.$(VERSION)
    -rm -f simon.$(VERSION)/ssci simon.$(VERSION)/simon
    cp *.cxx simon.$(VERSION)
    cp *.h simon.$(VERSION)
    cp ssci.l simon.$(VERSION)
    -rm -f simon.$(VERSION)/Makefile
    cp Makefile.source simon.$(VERSION)/Makefile
    cp Makefile.binary simon.$(VERSION)
    -rm simon.$(VERSION).tar
    tar -cf simon.$(VERSION).tar simon.$(VERSION)


smosaic : smosaic.skel Makefile ;
    rm -f smosaic
    cat smosaic.skel | sed  -e  's?_TCL_LIB_?$(TCL_TCL_LIBRARY_DIR)?g' \
                             -e  's?_TK_LIB_?$(TK_TK_LIBRARY_DIR)?g' \
                             -e  's?_DEFAULT_SS_?$(DEFAULT_SS)?g' \
                             -e 's?_PATH_?.?g' \
                             -e  's?_MOSAIC_?$(MOSAIC)?g' > smosaic
    chmod u+x smosaic

simon : simon.o ByteStore.o functions.o ss.o ssdbm.o ssdbm-tcl.o;
    $(CC) $(CFLAGS) -o simon simon.o ByteStore.o functions.o ss.o \
                                                ssdbm.o ssdbm-tcl.o  \
                                                 ltcl -ltk -lm -lgdbm -lX11
    strip simon

text-dump-0.3-format-ss : ByteStore.cxx text-dump-0.3-format-ss.cxx ssdbm.o;
    $(CC) $(CFLAGS) -D_ByteStore_platform_dependant \
                -o text-dump-0.3-format-ss \
                text-dump-0.3-format-ss.cxx ByteStore.cxx  ssdbm.o ss.o \
                             functions.o -lgdbm lg++

test-ss : ssci test-ss.dump;
    cat test-ss.dump | ssci test-ss
```

```
simon.o : simon.cxx ss.h
    $(CC) $(CFLAGS) -c simon.cxx

ssdbm-tcl.o : ssdbm-tcl.cxx ssdbm-tcl.h ss.h;
    $(CC) $(CFLAGS) -c ssdbm-tcl.cxx

ssci : ssci.o ssdbm.o ByteStore.o functions.o ss.o lex.yy.o;
    $(CC) $(CFLAGS) -o ssci ssci.o ssdbm.o ByteStore.o lex.yy.o \
                            functions.o ss.o -lgdbm -lg++
    strip ssci

ssbrowser : ssbrowser.o ssdbm.o ByteStore.o functions.o ss.o lex.yy.o;
    $(CC) $(CFLAGS) -o ssbrowser ssbrowser.o ssdbm.o ByteStore.o  lex.yy.o \
                            functions.o ss.o -lgdbm -lg++
    strip ssbrowser

ssindexer : ssdbm.o ByteStore.o functions.o ss.o ssindexer.cxx;
    $(CC) $(CFLAGS) -o ssindexer ssindexer.cxx ssdbm.o ByteStore.o  \
                           functions.o ss.o -lgdbm -lg++
    strip ssindexer

lex.yy.o : lex.yy.cxx ssci-lang.h;
    $(CC) $(CFLAGS) -c lex.yy.cxx

ssci.o : ssci.cxx ssci-lang.h ss.h;
    $(CC) $(CFLAGS) -c ssci.cxx

ssbrowser.o : ssbrowser.cxx ssci-lang.h ss.h;
    $(CC) $(CFLAGS) -c ssbrowser.cxx

ssdbm.o : ssdbm.cxx ssdbm.h ss.h;
    $(CC) $(CFLAGS) -c ssdbm.cxx

ByteStore.o : ByteStore.h ByteStore.cxx;
    $(CC) $(CFLAGS) -c ByteStore.cxx

functions.o : functions.cxx functions.h;
    $(CC) $(CFLAGS) -c functions.cxx

ss.o : ss.cxx ss.h;
    $(CC) $(CFLAGS) -c ss.cxx

lex.yy.cxx : ssci.l;
    flex -I ssci.l
    mv -f lex.yy.c lex.yy.cxx

ssg : ssg.cxx ssdbm.o ByteStore.o functions.o ss.o;
    $(CC) $(CFLAGS) -o ssg ssg.cxx ssdbm.o ByteStore.o functions.o  ss.o \
                                    -lgdbm -lg++
```

7 Makefile.binary

```
# YOU MUST EDIT THE VARIABLES BELOW.
# You should be able to get a working version without editing anything else.
# If you just want to build a binary, use make bintar.
# If you are using a binary distribution, you only need to edit MOSAIC =  mosaic,
# then do make smosaic.  You may want to change DEFAULT_SS = to


#   --- HOW TO USE THIS MAKEFILE ---
#
# make install - makes and installs simon.
#             This will make the install dir, copy everything you need to it,
#             to make simon independant of gdbm, tcl and tk.  It can also
#             makes a symbolic link to smosaic in your bin dir.
```

```
#
# make smosaic - (re)makes smosaic script, for when you change your setup.
#               Note that you may want:
#                    TCL_TCL_LIBRARY_DIR=tcl-lib
#                    TK_TK_LIBRARY_DIR=tk-lib
#               When you do this.
#


# ------------------- EDIT THESE FOR YOUR SYSTEM -----------------------
# NB ~ may not work in some places work


#where to install simon files upon Make install MUST BE FULL NON RELATIVE PATH
# ie no .. . or ~
INSTALL_DIR=.

#where to put a symbolic link to smosaic, so it will be in the PATH
#upon Make install.
BIN_DIR=.

#command used to start mosaic 2.2 or later
MOSAIC=mosaic

#default subject space to use  (files are (eg) my_ss.record my_ss.subject)
#give it a full path, or leave $(INSTALL_DIR)/ there
DEFAULT_SS=$(INSTALL_DIR)/test-ss



# ----------------- no edits below this line (I hope!) -----------------

#where special tcl and tk files (eg init.tcl) are
TCL_TCL_LIBRARY_DIR = $(INSTALL_DIR)/tcl-lib
TK_TK_LIBRARY_DIR = $(INSTALL_DIR)/tk-lib

VERSION=0.7

BINARIES = ssci simon simple-simon.tcl \
       README README.upgrading_to_0.4 smosaic.skel \
       simon.bit send-simon text-dump-0.3-format-ss test-ss.dump \
       test-ss.subject test-ss.record

message :;
    echo must do "make install" or "make smosaic"

install: ssci simon;
    -mkdir -p $(INSTALL_DIR)
    -mkdir $(INSTALL_DIR)/tcl-lib
    -mkdir $(INSTALL_DIR)/tk-lib
    -cp tcl-lib/* $(INSTALL_DIR)/tcl-lib
    -cp tk-lib/* $(INSTALL_DIR)/tk-lib
    -cp $(BINARIES) $(INSTALL_DIR)
    -cp Makefile.binary $(INSTALL_DIR)/Makefile
    cat smosaic.skel | sed  -e 's?_TCL_LIB_?$(INSTALL_DIR)/tcllib?g' \
                                    -e  's?_TK_LIB_?$(INSTALL_DIR)/tk-lib?g' \
                                    -e  's?_MOSAIC_?$(MOSAIC)?g' \
                                    -e  's?_DEFAULT_SS_?$(DEFAULT_SS)?g' \
                                    -e  's?_PATH_?$(INSTALL_DIR)?g' \
                                                        >  $(INSTALL_DIR)/smosaic
    chmod u+x $(INSTALL_DIR)/smosaic
    ln -s  $(INSTALL_DIR)/smosaic $(BIN_DIR)/smosaic


smosaic : smosaic.skel Makefile ;
    rm -f smosaic
    cat smosaic.skel | sed  -e  's?_TCL_LIB_?$(TCL_TCL_LIBRARY_DIR)?g' \
                                    -e  's?_TK_LIB_?$(TK_TK_LIBRARY_DIR)?g' \
                                    -e  's?_DEFAULT_SS_?$(DEFAULT_SS)?g' \
                                    -e  's?_PATH_?$(INSTALL_DIR)?g' \
                                    -e  's?_MOSAIC_?$(MOSAIC)?g' > smosaic
    chmod u+x smosaic
    -ln -s $(INSTALL_DIR)/smosaic $(BIN_DIR)/smosaic
```

8 Makefile.source

```
# YOU MUST EDIT THE VARIABLES BELOW.
# You should be able to get a working version without editing anything else.
# If you just want to build a binary, use make bintar.
# If you are using a binary distribution, you only need to edit MOSAIC =  mosaic,
# then do make smosaic.  You may want to change DEFAULT_SS = to


#    --- HOW TO USE THIS MAKEFILE ---
#
# make        - make binaries, use existing tcl library dir
#
# make bintar  - makes a binary tar package that is independant of gdbm,
#             tcl & tk, use it to make a binary package for others to use.
#
# make install - makes and installs simon.
#             This will make the install dir, copy everything you need to it,
#             to make simon independant of gdbm, tcl and tk.  It can also
#             makes a symbolic link to smosaic in your bin dir.
#
# make smosaic - (re)makes smosaic script, for when you change your setup.
#             Note that you may want:
#                  TCL_TCL_LIBRARY_DIR=tcl-lib
#                  TK_TK_LIBRARY_DIR=tk-lib
#             When you do this.
#
# make clean   - deletes binaries & object files
#


# ------------------- EDIT THESE FOR YOUR SYSTEM ----------------------
# Please put full path names in, I don't think that ~ will work

# where object libs (eg libtcl.a) are
TCL_LIB_DIR=.
TK_LIB_DIR=.
GDBM_LIB_DIR=.

#where include files (eg tcl.h) are,
TCL_INCLUDE_DIR=.
TK_INCLUDE_DIR=.
GDBM_INCLUDE_DIR=.

#where to install simon files upon Make install. MUST BE FULL NON RELATIVE
PATH
# ie no .. . or ~
INSTALL_DIR=.

#where special tcl and tk files (eg init.tcl) are
TCL_TCL_LIBRARY_DIR = $(TCL_LIB_DIR)/library
TK_TK_LIBRARY_DIR = $(TK_LIB_DIR)/library

#where to put a symbolic link to smosaic, so it will be in the PATH
#upon Make install. MUST BE FULL NON RELATIVE PATH ie no .. . or ~
BIN_DIR=.

#command used to start mosaic 2.2 or later
MOSAIC=mosaic

#default subject space to use  (files are (eg) .my_ss.record .my_ss.subject)
#Give it a full path, or put $(INSTALL_DIR)/test-ss there
DEFAULT_SS=test-ss

#the compiler, should be gnu c++ compiler, g++ or gcc
CC=g++

#extra c flags, eg -g for debugging,
#poiters to obsure directories for libX11, etc
```

```
EXTRA_CFLAGS= -w -O

#name for binary package, should be hardware & operating system specific,
#but it is just used to give the tar file a name, so be as specific as you can,
#but don't worry about it, eg  sparc-sunos4.1
BIN_VERSION=unknown


# ----------------- no edits below this line (I hope!) -----------------

VERSION=0.7

BINARIES = ssci simon simple-simon.tcl \
        README README.upgrading_to_0.4 smosaic.skel \
        simon.bit send-simon text-dump-0.3-format-ss test-ss.dump \
        test-ss.subject test-ss.record

CFLAGS = -L$(TCL_LIB_DIR) -L$(TK_LIB_DIR) -L$(GDBM_LIB_DIR) \
     -I$(TCL_INCLUDE_DIR) -I$(TK_INCLUDE_DIR) I$(GDBM_INCLUDE_DIR) \
     -I. $(EXTRA_CFLAGS)


all : ssci simon text-dump-0.3-format-ss test-ss smosaic;

bintar : ssci simon ;
    -mkdir simon.$(VERSION).binary
    -rm -rf simon.$(VERSION).binary/*
    -mkdir simon.$(VERSION).binary/tcl-lib
    -mkdir simon.$(VERSION).binary/tk-lib
    -rm -f simon.$(VERSION).binary/tcl-lib/*
    -rm -f simon.$(VERSION).binary/tk-lib/*
    -cp  $(TCL_TCL_LIBRARY_DIR)/* simon.$(VERSION).binary/tcl-lib
    -cp  $(TK_TK_LIBRARY_DIR)/* simon.$(VERSION).binary/tk-lib
    cp  $(BINARIES) simon.$(VERSION).binary
    cp Makefile.binary simon.$(VERSION).binary/Makefile
    cp test-ss.record simon.$(VERSION).binary/test-ss.record
    cp test-ss.subject simon.$(VERSION).binary/test-ss.subject
    cat smosaic.skel | sed  -e 's?_TCL_LIB_?tcl-lib?g' \
                                    -e 's?_TK_LIB_?tklib?g' \
                                    -e  's?_MOSAIC_?mosaic?g' \
                                    -e  's?_DEFAULT_SS_?test-ss?g' \
                                    -e 's?_PATH_?.?g' \
                                            >  simon.$(VERSION).binary/smosaic
    chmod u+x simon.$(VERSION).binary/smosaic
    tar -cf simon.$(VERSION).binary.$(BIN_VERSION).tar \
                                                    simon.$(VERSION).binary
    rm -fr simon.$(VERSION).binary


install: ssci simon;
    -mkdir -p $(INSTALL_DIR)
    -mkdir $(INSTALL_DIR)/tcl-lib
    -mkdir $(INSTALL_DIR)/tk-lib
    -cp $(TCL_TCL_LIBRARY_DIR)/* $(INSTALL_DIR)/tcl-lib
    -cp $(TK_TK_LIBRARY_DIR)/* $(INSTALL_DIR)/tk-lib
    -cp $(BINARIES) $(INSTALL_DIR)
    cp test-ss.record $(INSTALL_DIR)
    cp test-ss.subject $(INSTALL_DIR)
    -cat Makefile.binary | sed \
                -e 's?INSTALL_DIR =?INSTALL_DIR =  $(INSTALL_DIR)?' \
                -e 's?BIN_DIR =?BIN_DIR = $(BIN_DIR)?' \
                -e 's?MOSAIC =?MOSAIC = $(MOSAIC)?' \
                                                    >  $(INSTALL_DIR)/Makefile
    -cat smosaic.skel | sed  -e 's?_TCL_LIB_?$(INSTALL_DIR)/tcllib?g' \
                                    -e  's?_TK_LIB_?$(INSTALL_DIR)/tk-lib?g' \
                                    -e  's?_MOSAIC_?$(MOSAIC)?g' \
                                    -e  's?_DEFAULT_SS_?$(DEFAULT_SS)?g' \
                                    -e  's?_PATH_?$(INSTALL_DIR)?g' \
                                                    >  $(INSTALL_DIR)/smosaic
    chmod u+x $(INSTALL_DIR)/smosaic
    -ln -s $(INSTALL_DIR)/smosaic $(BIN_DIR)/smosaic


simon.$(VERSION).binary.$(BIN_VERSION).tar: bintar;
```

```
clean : ;
    rm -f *.o ssci simon smosaic *.tar



smosaic : smosaic.skel Makefile ;
    rm -f smosaic
    cat smosaic.skel | sed  -e  's?_TCL_LIB_?$(TCL_TCL_LIBRARY_DIR)?g' \
                                -e  's?_TK_LIB_?$(TK_TK_LIBRARY_DIR)?g' \
                                -e  's?_DEFAULT_SS_?$(DEFAULT_SS)?g' \
                                -e 's?_PATH_?.?g' \
                                -e  's?_MOSAIC_?$(MOSAIC)?g' > smosaic
    chmod u+x smosaic
    -ln -s $(INSTALL_DIR)/smosaic $(BIN_DIR)/smosaic

simon : simon.o ByteStore.o functions.o ss.o ssdbm.o ssdbm-tcl.o;
    $(CC) $(CFLAGS) -o simon simon.o ByteStore.o functions.o ss.o \
                                                ssdbm.o ssdbm-tcl.o  \
                                                ltcl -ltk -lm -lgdbm -lX11
    strip simon

simon.o : simon.cxx ss.h;
    $(CC) $(CFLAGS) -c simon.cxx

ssdbm-tcl.o : ssdbm-tcl.cxx ssdbm-tcl.h ss.h;
    $(CC) $(CFLAGS) -c ssdbm-tcl.cxx

ssci : ssci.o ssdbm.o ByteStore.o functions.o ss.o lex.yy.o;
    $(CC) $(CFLAGS) -o ssci ssci.o ssdbm.o ByteStore.o lex.yy.o \
                            functions.o ss.o -lgdbm -lg++
    strip ssci


lex.yy.o : lex.yy.cxx ssci-lang.h;
    $(CC) $(CFLAGS) -c lex.yy.cxx

ssci.o : ssci.cxx ssci-lang.h ss.h;
    $(CC) $(CFLAGS) -c ssci.cxx

ssdbm.o : ssdbm.cxx ssdbm.h ss.h;
    $(CC) $(CFLAGS) -c ssdbm.cxx

ByteStore.o : ByteStore.h ByteStore.cxx;
    $(CC) $(CFLAGS) -c ByteStore.cxx

functions.o : functions.cxx functions.h;
    $(CC) $(CFLAGS) -c functions.cxx

ss.o : ss.cxx ss.h;
    $(CC) $(CFLAGS) -c ss.cxx

text-dump-0.3-format-ss : ByteStore.cxx text-dump-0.3-format-ss.cxx ssdbm.o;
    $(CC) $(CFLAGS) -D_ByteStore_platform_dependant \
                -o text-dump-0.3-format-ss \
                text-dump-0.3-format-ss.cxx ByteStore.cxx  ssdbm.o ss.o \
                                        functions.o -lgdbm lg++

test-ss : ssci test-ss.dump;
    cat test-ss.dump | ssci test-ss

#lex.yy.cxx : ssci.l;
#       flex -I ssci.l
#       mv -f lex.yy.c lex.yy.cxx
```

9 README

SIMON is written by Mark Johnson, m.j.johnson@qmw.ac.uk
You must use a WWW client, eg mosaic, to read the documentaion
for installation & use. Use 'open URL' on mosaic's file menu and paste this in:
http://web.elec.qmw.ac.uk:12121/
You must edit the Makefile, but there are comments in it to help you, & notes
on it's use are embeded to.

10 README.upgrading_to_0.4

Version 0.4 of simon is is planned to be a stable release.
New versions will have bug fixes and new features, but major features and
file formats will not change.
BUT - 0.4 uses a new file format that is incompatible with 0.3 and other
previous formats. The change is to make subject spaces portable across
different platforms. It is easy to upgrade your subject space:

Assuming that your subject space is called ~/.my-ss,
ie there are file ~/.my-ss.record and ~/.my-ss.subject, you are inside
a simon.0.4 (source or binary) directory, and you have executables, type :

```
text-dump-0.3-format-ss ~/.my-ss | ssci new.my-ss
mv new.my-ss.record ~/.my-ss.record
mv new.my-ss.subject ~/.my-ss.subject
```

11 send-simon

```
#! /bin/sh

MAIL_ADDRESS='zgee9119@elec.qmw.ac.uk'
SUBJECT='SIMON_SUBJECT_SPACE_TEXT_DUMP'
MAIL='/bin/mail'
TMP='/tmp'

if [ $# -eq 1 ]
then
  SS=$1
  echo text dump | ssci $SS >> $TMP/send-simon.tmp.$$
  $MAIL -s $SUBJECT $MAIL_ADDRESS < $TMP/send-simon.tmp.$$
  rm -f $TMP/send-simon.tmp.$$
else
  echo need one arg, name of subject space
fi
```

12 simon.bit

```
#define simon_width 40
#define simon_height 40
static char simon_bits[] = {
   0x00, 0xe0, 0xff, 0xff, 0xff, 0x00, 0x60, 0x3e, 0xc0, 0xff, 0x00, 0xe0,
   0xa8, 0xea, 0xff, 0x00, 0x60, 0xfe, 0xc1, 0xef, 0x00, 0xe0, 0xff, 0xeb,
   0x87, 0x00, 0x60, 0xfe, 0xc3, 0xcf, 0x00, 0xe0, 0xfe, 0xeb, 0xef, 0x00,
   0x40, 0xfc, 0x81, 0x87, 0x00, 0xc0, 0xfa, 0x2b, 0xaf, 0x00, 0xc0, 0xf9,
   0x00, 0xc3, 0x00, 0x80, 0xfa, 0xaa, 0xab, 0x00, 0x80, 0xc1, 0x00, 0x80,
   0x00, 0x00, 0xab, 0xaa, 0x6a, 0x00, 0x00, 0x02, 0x00, 0x2c, 0x00, 0xf0,
```

```
  0xac, 0xab, 0x1e, 0x80, 0x1f, 0x98, 0xd4, 0x0d, 0xf0, 0x0f, 0xe0, 0x7f,
  0x00, 0xff, 0x03, 0x00, 0xe7, 0x0f, 0xe2, 0x0f, 0x00, 0x24, 0x7c, 0x86,
  0x3f, 0x00, 0x24, 0xf0, 0x1c, 0xfc, 0x01, 0x24, 0xfe, 0xfc, 0xe3, 0xff,
  0xe7, 0x3f, 0xf8, 0x1f, 0xff, 0xef, 0x0b, 0xf8, 0xff, 0xdb, 0xe7, 0x0f,
  0x78, 0xfc, 0xdf, 0xe5, 0x07, 0x70, 0xf0, 0xff, 0xe7, 0x00, 0x70, 0xc0,
  0xff, 0x27, 0x00, 0x60, 0x00, 0xf8, 0x24, 0x0c, 0x60, 0x00, 0x9c, 0x24,
  0x12, 0x40, 0x00, 0x94, 0xa5, 0x19, 0x40, 0x00, 0xb4, 0x6c, 0x1c, 0x00,
  0x00, 0x9e, 0x24, 0x06, 0x00, 0x00, 0xf6, 0x37, 0x02, 0x00, 0x00, 0x9e,
  0x94, 0x01, 0x00, 0x00, 0xd6, 0x86, 0x00, 0x00, 0x00, 0x76, 0x83, 0x00,
  0x00, 0x00, 0x1e, 0x80, 0x00, 0x00, 0x00, 0x02, 0xc4, 0x00, 0x00, 0x00,
  0x04, 0x42, 0x00, 0x00, 0x00, 0x5c, 0x6b, 0x00};
```

13 simon.cxx

```
#include <string.h>

/*
 * main.c --
 *
 *      This file contains the main program for "wish", a windowing
 *      shell based on Tk and Tcl.  It also provides a template that
 *      can be used as the basis for main programs for other Tk
 *      applications.
 *
 * Copyright (c) 1990-1993 The Regents of the University of California.
 * All rights reserved.
 *
 * Permission is hereby granted, without written agreement and without
 * license or royalty fees, to use, copy, modify, and distribute this
 * software and its documentation for any purpose, provided that the
 * above copyright notice and the following two paragraphs appear in
 * all copies of this software.
 *
 * IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY
PARTY FOR
 * DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES
ARISING OUT
 * OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE
UNIVERSITY OF
 * CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 * THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY
WARRANTIES,
 * INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY
 * AND FITNESS FOR A PARTICULAR PURPOSE.  THE SOFTWARE PROVIDED
HEREUNDER IS
 * ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO
OBLIGATION TO
 * PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR
MODIFICATIONS.
 */

#include <stdlib.h>
#include <stdio.h>
#include <iostream.h>
#include <tcl.h>
#include <tk.h>
#include <string.h>
#include "ssdbm.h"
#include "ssdbm-tcl.h"


/*
 * Declarations for various library procedures and variables (don't want
 * to include tkInt.h or tkConfig.h here, because people might copy this
 * file out of the Tk source directory to make their own modified versions).
 */
```

```
//extern void         exit _ANSI_ARGS_((int status));
//extern int          isatty _ANSI_ARGS_((int fd));
//extern int          read _ANSI_ARGS_((int fd, char *buf, size_t size));
//extern char *       strrchr _ANSI_ARGS_((CONST char *string, int c));

/*
 * Global variables used by the main program:
 */

static Tk_Window mainWindow;    /* The main window for the application.  If
                                 * NULL then the  application no longer
                                 * exists. */
static Tcl_Interp *interp;      /* Interpreter for this application. */
char *tcl_RcFileName = NULL;    /* Name of a user-specific startup script
                                 * to source if the  application is being run
                                 * interactively  (e.g. "~/.wishrc").  Set
                                 * by Tcl_AppInit.  NULL means don't source
                                 * anything ever.  */

static char errorExitCmd[] = "exit 1";

/*
 * Command-line options:
 */

static int synchronize = 0;
static char *fileName = NULL;
static char *name = NULL;
static char *display = NULL;
static char *geometry = NULL;
static char *mpid = NULL;
static char *mtmp = NULL;
static char *mhotlist = NULL;
static char *subject_space = NULL;
static char *icon = NULL;
static char *simon_path = NULL;
static char *help = NULL;

static Tk_ArgvInfo argTable[] = {
    {"-file", TK_ARGV_STRING, (char *) NULL, (char *) &fileName,
     "File from which to read commands"},
    {"-ss", TK_ARGV_STRING, (char *) NULL, (char *) &subject_space,
     "Subject space to open on startup"},
    {"-geometry", TK_ARGV_STRING, (char *) NULL, (char *) &geometry,
     "Initial geometry for window"},
    {"-display", TK_ARGV_STRING, (char *) NULL, (char *) &display,
     "Display to use"},
    {"-name", TK_ARGV_STRING, (char *) NULL, (char *) &name,
     "Name to use for application"},
    {"-sync", TK_ARGV_CONSTANT, (char *) 1, (char *) &synchronize,
     "Use synchronous mode for display server"},
    {"-mpid", TK_ARGV_STRING, (char *) NULL, (char *) &mpid,
     "PID of runing mosaic process"},
    {"-mhotlist", TK_ARGV_STRING, (char *) NULL, (char *) &mhotlist,
     "mosaic hotlist file"},
    {"-mtempdir", TK_ARGV_STRING, (char *) NULL, (char *) &mtmp,
     "tmp directory of runing mosaic process"},
    {"-icon", TK_ARGV_STRING, (char *) NULL, (char *) &icon,
     "optional icon to use (bitmap file)"},
    {"-path", TK_ARGV_STRING, (char *) NULL, (char *) &simon_path,
     "optional path where simon looks for it's files (makes this  path home dir)"},

    {"-help", TK_ARGV_STRING, (char *) NULL, (char *) &help,
     "optional URL for help (defaults to SIMON home server)"},
    {(char *) NULL, TK_ARGV_END, (char *) NULL, (char *) NULL,
     (char *) NULL}
};


/*
 * Forward declarations for procedures defined later in this file:
 */

/*
```

```
*----------------------------------------------------------------------
*
* main --
*
*      Main program for Wish.
*
* Results:
*      None. This procedure never returns (it exits the process when
*      it's done
*
* Side effects:
*      This procedure initializes the wish world and then starts
*      interpreting commands;  almost anything could happen, depending
*      on the script being interpreted.
*
*----------------------------------------------------------------------
*/

int
main(int argc, char** argv)
{
   char *args, *p, *msg;
   char buf[20];
   int code;

   interp = Tcl_CreateInterp();
#ifdef TCL_MEM_DEBUG
   Tcl_InitMemory(interp);
#endif

    /* Set the "tcl_interactive" variable.
   Tcl_SetVar(interp, "tcl_interactive", 1, TCL_GLOBAL_ONLY);


   /*
    * Parse command-line arguments.
    */

   if (Tk_ParseArgv(interp, (Tk_Window) NULL, &argc, argv, argTable, 0)
       != TCL_OK) {
    fprintf(stderr, "%s\n", interp->result);
    exit(1);
   }

   // set application name
   if (name == NULL) {
    if (fileName != NULL) {
       p = fileName;
    } else {
       p = argv[0];
    }
    name = strrchr(p, '/');
    if (name != NULL) {
       name++;
    } else {
       name = p;
    }
   }


   /*
    * If a display was specified, put it into the DISPLAY
    * environment variable so that it will be available for
    * any sub-processes created by us.
    */

   if (display != NULL) {
    Tcl_SetVar2(interp, "env", "DISPLAY", display,  TCL_GLOBAL_ONLY);
   }

   /*
    * Initialize the Tk application.
    */
```

```
mainWindow = Tk_CreateMainWindow(interp, display, name, "Tk");
if (mainWindow == NULL) {
 fprintf(stderr, "%s\n", interp->result);
 exit(1);
}
if (synchronize) {
 XSynchronize(Tk_Display(mainWindow), True);
}
Tk_GeometryRequest(mainWindow, 200, 200);

/*
 * Make command-line arguments available in the Tcl variables "argc"
 * and "argv".  Also set the "geometry" variable from the geometry
 * specified on the command line.
 */

args = Tcl_Merge(argc-1, argv+1);
Tcl_SetVar(interp, "argv", args, TCL_GLOBAL_ONLY);
ckfree(args);
sprintf(buf, "%d", argc-1);
Tcl_SetVar(interp, "argc", buf, TCL_GLOBAL_ONLY);
Tcl_SetVar(interp, "argv0", (fileName != NULL) ? fileName : argv[0],
    TCL_GLOBAL_ONLY);
if (geometry != NULL) {
 Tcl_SetVar(interp, "geometry", geometry, TCL_GLOBAL_ONLY);
}


// set tcl variables

if (mpid!=NULL)
 Tcl_SetVar (interp, "mpid", mpid, TCL_GLOBAL_ONLY);
  else Tcl_SetVar (interp, "mpid", "", TCL_GLOBAL_ONLY);
if (mtmp!=NULL)
 Tcl_SetVar (interp, "mtmp", mtmp, TCL_GLOBAL_ONLY);
  else Tcl_SetVar (interp, "mtmp", "/tmp", TCL_GLOBAL_ONLY);
if (mhotlist!=NULL)
 Tcl_SetVar (interp, "mhotlist", mhotlist, TCL_GLOBAL_ONLY);
  else Tcl_SetVar (interp, "mhotlist", "~/.mosaic-hotlist-default",
                                                        TCL_GLOBAL_ONLY);
if (help!=NULL)
 Tcl_SetVar (interp, "help", help, TCL_GLOBAL_ONLY);
  else Tcl_SetVar (interp, "help",
              "http://web.elec.qmw.ac.uk:12121/ simon.help.html",
                                                        TCL_GLOBAL_ONLY);
if (icon!=NULL)
 Tcl_SetVar (interp, "icon", icon, TCL_GLOBAL_ONLY);
  else Tcl_SetVar (interp, "icon", "null", TCL_GLOBAL_ONLY);
if (simon_path!=NULL)
 Tcl_SetVar (interp, "simon_path", simon_path,  TCL_GLOBAL_ONLY);
  else Tcl_SetVar (interp, "simon_path", "", TCL_GLOBAL_ONLY);
if (subject_space!=NULL)
 Tcl_SetVar (interp, "subject_space", subject_space,  TCL_GLOBAL_ONLY);
  else Tcl_SetVar (interp, "subject_space", "", TCL_GLOBAL_ONLY);

/*
 * Invoke application-specific initialization.
 */

if (Tcl_AppInit(interp) != TCL_OK) {
 fprintf(stderr, "Tcl_AppInit failed: %s\n", interp->result);
}


/*
 * Invoke the script specified on the command line
 */

if (fileName != NULL) {
 String tcl_file;
 if (simon_path!=NULL) {tcl_file+=simon_path; tcl_file+="/";};
 tcl_file+=fileName;
 code = Tcl_VarEval(interp, "source ", (char*)tcl_file, (char  *) NULL);
 if (code != TCL_OK) {
```

```
        cout<<"error in, or can't find .tcl script\n"; exit (-1) ;
    }
   }
    else {cout << "I must have a -file arg, eg -file simple-simon.tcl\n";
                                                                    exit (-1);};


   /*
    * Set the geometry of the main window, if requested.
    */

   if (geometry != NULL) {
    code = Tcl_VarEval(interp, "wm geometry . ", geometry, (char  *) NULL);
    if (code != TCL_OK) {
        fprintf(stderr, "%s\n", interp->result);
    }
   }

 // create a subject space database manager

   if (subject_space == NULL)
    {
    cout << "I must have a -ss arg, eg -ss .simon_ss_default\n";
    exit (-1);
    };
   SS_Error error;
   SSDBM ssdbm(error, subject_space);
   if (error!=OK)
    {cout<<"error opening subject space\n"; exit(-1);};
   ssdbm_ptr=&ssdbm;


   fflush(stdout);

   /*
    * Loop infinitely, waiting for commands to execute.  When there
    * are no windows left, Tk_MainLoop returns and we exit.
    */

   Tk_MainLoop();

   /*
    * Don't exit directly, but rather invoke the Tcl "exit" command.
    * This gives the application the opportunity to redefine "exit"
    * to do additional cleanup.
    */

   Tcl_Eval(interp, "exit");
   exit(1);

}



/*
 *----------------------------------------------------------------------
 *
 * Tcl_AppInit --
 *
 *      This procedure performs application-specific initialization.
 *      Most applications, especially those that incorporate additional
 *      packages, will have their own version of this procedure.
 *
 * Results:
 *      Returns a standard Tcl completion code, and leaves an error
 *      message in interp->result if an error occurs.
 *
 * Side effects:
 *      Depends on the startup script.
 *
 *----------------------------------------------------------------------
 */

int
```

```
Tcl_AppInit(Tcl_Interp *interp)
{
   Tk_Window main;

   main = Tk_MainWindow(interp);

   /*
    * Call the init procedures for included packages.  Each call should
    * look like this:
    *
    * if (Mod_Init(interp) == TCL_ERROR) {
    *     return TCL_ERROR;
    * }
    *
    * where "Mod" is the name of the module.
    */

   if (Tcl_Init(interp) == TCL_ERROR) {
    return TCL_ERROR;
   }
   if (Tk_Init(interp) == TCL_ERROR) {
    return TCL_ERROR;
   }

   /*
    * Call Tcl_CreateCommand for application-specific commands, if
    * they weren't already created by the init procedures called above.
    */

   /*
    * Specify a user-specific startup file to invoke if the application
    * is run interactively.  Typically the startup file is "~/.apprc"
    * where "app" is the name of the application.  If this line is deleted
    * then no user-specific startup file will be run under any conditions.
    */

   register_new_tcl_commands(interp);

  // tcl_RcFileName = "simonrc";
   return TCL_OK;
}
```

14 simple-simon.tcl

```
bind Entry <Left> {%W icursor [expr [%W index insert]-1]}
bind Entry <Right> {%W icursor [expr [%W index insert]+1]}
bind Entry <Return> {;}

proc intersection_alphanumeric_bindings {} \
{
bind .intersection a {.intersection insert insert %A ; resources_now_old}
bind .intersection b {.intersection insert insert %A ; resources_now_old}
bind .intersection c {.intersection insert insert %A ; resources_now_old}
bind .intersection d {.intersection insert insert %A ; resources_now_old}
bind .intersection e {.intersection insert insert %A ; resources_now_old}
bind .intersection f {.intersection insert insert %A ; resources_now_old}
bind .intersection g {.intersection insert insert %A ; resources_now_old}
bind .intersection h {.intersection insert insert %A ; resources_now_old}
bind .intersection i {.intersection insert insert %A ; resources_now_old}
bind .intersection j {.intersection insert insert %A ; resources_now_old}
bind .intersection k {.intersection insert insert %A ; resources_now_old}
bind .intersection l {.intersection insert insert %A ; resources_now_old}
bind .intersection m {.intersection insert insert %A ; resources_now_old}
bind .intersection n {.intersection insert insert %A ; resources_now_old}
bind .intersection o {.intersection insert insert %A ; resources_now_old}
bind .intersection p {.intersection insert insert %A ; resources_now_old}
bind .intersection q {.intersection insert insert %A ; resources_now_old}
bind .intersection r {.intersection insert insert %A ; resources_now_old}
```

```
bind .intersection s {.intersection insert insert %A ; resources_now_old}
bind .intersection t {.intersection insert insert %A ; resources_now_old}
bind .intersection u {.intersection insert insert %A ; resources_now_old}
bind .intersection v {.intersection insert insert %A ; resources_now_old}
bind .intersection w {.intersection insert insert %A ; resources_now_old}
bind .intersection x {.intersection insert insert %A ; resources_now_old}
bind .intersection y {.intersection insert insert %A ; resources_now_old}
bind .intersection z {.intersection insert insert %A ; resources_now_old}
bind .intersection A {.intersection insert insert %A ; resources_now_old}
bind .intersection B {.intersection insert insert %A ; resources_now_old}
bind .intersection C {.intersection insert insert %A ; resources_now_old}
bind .intersection D {.intersection insert insert %A ; resources_now_old}
bind .intersection E {.intersection insert insert %A ; resources_now_old}
bind .intersection F {.intersection insert insert %A ; resources_now_old}
bind .intersection G {.intersection insert insert %A ; resources_now_old}
bind .intersection H {.intersection insert insert %A ; resources_now_old}
bind .intersection I {.intersection insert insert %A ; resources_now_old}
bind .intersection J {.intersection insert insert %A ; resources_now_old}
bind .intersection K {.intersection insert insert %A ; resources_now_old}
bind .intersection L {.intersection insert insert %A ; resources_now_old}
bind .intersection M {.intersection insert insert %A ; resources_now_old}
bind .intersection N {.intersection insert insert %A ; resources_now_old}
bind .intersection O {.intersection insert insert %A ; resources_now_old}
bind .intersection P {.intersection insert insert %A ; resources_now_old}
bind .intersection Q {.intersection insert insert %A ; resources_now_old}
bind .intersection R {.intersection insert insert %A ; resources_now_old}
bind .intersection S {.intersection insert insert %A ; resources_now_old}
bind .intersection T {.intersection insert insert %A ; resources_now_old}
bind .intersection U {.intersection insert insert %A ; resources_now_old}
bind .intersection V {.intersection insert insert %A ; resources_now_old}
bind .intersection W {.intersection insert insert %A ; resources_now_old}
bind .intersection X {.intersection insert insert %A ; resources_now_old}
bind .intersection Y {.intersection insert insert %A ; resources_now_old}
bind .intersection Z {.intersection insert insert %A ; resources_now_old}
bind .intersection 0 {.intersection insert insert %A ; resources_now_old}
bind .intersection 1 {.intersection insert insert %A ; resources_now_old}
bind .intersection 2 {.intersection insert insert %A ; resources_now_old}
bind .intersection 3 {.intersection insert insert %A ; resources_now_old}
bind .intersection 4 {.intersection insert insert %A ; resources_now_old}
bind .intersection 5 {.intersection insert insert %A ; resources_now_old}
bind .intersection 6 {.intersection insert insert %A ; resources_now_old}
bind .intersection 7 {.intersection insert insert %A ; resources_now_old}
bind .intersection 8 {.intersection insert insert %A ; resources_now_old}
bind .intersection 9 {.intersection insert insert %A ; resources_now_old}
bind .intersection 0 {.intersection insert insert %A ; resources_now_old}
}


# ----------- some globals & general stuff ------------

set tcl_interactive 0
set frame_relief raised
set frame_borderwidth 2
set button_space 0
wm minsize . 60 140
wm maxsize . 800 1000
wm title . "simon"

# these are global variables set before simon goes into X event driven mode
# mpid mtmp mhotlist simon_path icon subject_space help



# ------------ main window widgits -----------------

frame .intersection_frame -relief $frame_relief -borderwidth  $frame_borderwidth
label .intersection_label -text {current subject intersection}
entry .intersection -width 40 -relief sunken
pack .intersection_label -in .intersection_frame
pack .intersection -fill x -in .intersection_frame

frame .buttons_frame -relief $frame_relief -borderwidth $frame_borderwidth
frame .button_bar_1
frame .button_bar_2
```

```
frame .button_bar_3
button .goto_b -width 8 -text "Goto"
button .new_resource_b -width 8 -text "New Resource"
button .select_subjects_b -width 8 -text "Select Subjects"
button .edit_resource_b -width 8 -text "Edit Resource"
button .exit_b -width 8 -text "Exit"
button .help_b -width 8 -text "Help"

pack .new_resource_b .edit_resource_b  -padx 0 -pady 0\
                -side left -fill x -expand true -in .button_bar_1
pack .goto_b .select_subjects_b -padx 0 -pady 0\
                -side left -fill x -expand true -in .button_bar_2
pack .exit_b .help_b -padx 0 -pady 0\
                -side left -fill x -expand true -in .button_bar_3

pack .button_bar_1 -side top -fill x -expand true -in .buttons_frame
pack .button_bar_2 -side bottom -fill x -expand true -in .buttons_frame

frame .resources_frame -relief $frame_relief -borderwidth $frame_borderwidth
frame .resources_label_button_frame
label .resources_label -text {resources}
button .resources_update_b -text " Update " -relief raised
listbox .resources -relief sunken
scrollbar .resources_vscroll -orient vertical
scrollbar .resources_hscroll -orient horizontal
entry .url -relief sunken
pack .resources_label -side left -in .resources_label_button_frame
pack .resources_update_b -side right -in .resources_label_button_frame
pack .resources_label_button_frame -padx 16 -side top -fill x \
                                                -in  .resources_frame
pack .url -fill x -side bottom -in .resources_frame
pack .resources_hscroll -fill x -side bottom -in .resources_frame
pack .resources_vscroll -fill y -side right -in .resources_frame
pack .resources -fill both -expand true -side left -in .resources_frame

# ---- definition of main window's widgets operation and interaction -----

# these variables hold the data for the main window's widgits to display, and
# for widgits to change
set intersection {}
set url {}
set urls {}
set resources {}
set select_subjects {}


.intersection configure -textvariable intersection


proc set_current_dir {} \
  {
  global simon_path file icon
  if {$simon_path!=""} \
    {
    if {![file isdirectory $simon_path]} \
                {puts "-path was not a directory\n"; exit}
    if {[catch {cd $simon_path}]} \
                {puts "couldn't change to -path directory\n"}
    if {($icon!="null") && (![file exists $icon])} \
                {puts "can't find -icon file\n"; exit}
    }
  }

proc display_main_window {} \
  {
  global icon
  if {$icon!="null"} {global icon; wm iconbitmap . @$icon}
  pack .intersection_frame -fill x
  pack .buttons_frame -fill x
  pack .resources_frame -fill both -expand true
  pack .button_bar_3 -fill x -side bottom
  focus .intersection
  }
```

```
proc exit_dialog {} \
   {
   global subject_space
   set exit_choice [tk_dialog .exit_win "exit simon?" \
     "Exit simon ?
(Remember to do a send simon about once a week)" "" 0 \
                    "    Yes     "\
                    " Yes + send simon " \
                    "    No     "]

   if {$exit_choice==0} \
     {ssdbm_destructor; exit}
   if {$exit_choice==1} \
     {
     puts "Doing a send-simon, this may take a few seconds!!"
     ssdbm_destructor
     if {[catch {exec send-simon $subject_space}]!=0} \
       {puts "\n\nsend-simon DID NOT WORK, try it from the command  line\n"}\
      else {puts "Thanks for your subject space"}
     exit
     }
   }


proc horizontal_scroll {index} \
   {
   .resources xview $index
   .url view $index
   }

proc update_resources {}  \
   {
   global resources urls intersection
   set resources {}
   set urls {}
   .resources_update_b configure -relief groove

  if {[llength $intersection]==0} \
       {ssdbm_resources_list urls resources} \
     else {ssdbm_resources_in_intersection urls resources $intersection}

   .resources delete 0 [.resources size]
   foreach resource $resources {.resources insert [.resources size] $resource}
   }

proc resources_now_old {}  \
   {
   .resources_update_b configure -relief raised
   }

.resources_update_b configure -command update_resources

intersection_alphanumeric_bindings
bind .intersection <Return> update_resources
bind .intersection <2> {%W insert insert [selection get]}

#the after command is just to supply a non null script that does nothing
bind .url <1> {after 1 {} }

.resources configure -yscrollcommand {.resources_vscroll set}
.resources_vscroll configure -command {.resources yview}
.resources configure -xscrollcommand {.resources_hscroll set}
.url configure -scrollcommand {.resources_hscroll set}
.resources_hscroll configure -command horizontal_scroll


proc goto_resource {} \
   {
   global url mpid mtmp
   if {$mpid!=""} \
     {
     exec rm -f $mtmp/Mosaic.$mpid
     set f [open $mtmp/Mosaic.$mpid w]
```

```
    puts $f "goto"
    puts $f $url
    close $f
    if {[catch {exec kill -USR1 $mpid}]!=0} \
                {error "Simon can't find mosaic"}
    }
  }


.goto_b configure -command goto_resource
.new_resource_b configure -command \
  {
  if {[winfo exists .nrw]} \
    {wm deiconify .nrw} \
   else {new_resource}
  }

.edit_resource_b configure -command \
  {
  if {[winfo exists .erw]} \
    { wm deiconify .erw } \
   else {
     if {[.resources curselect] != "" } \
       { edit_resource $url [.resources get [.resources curselect]]  }
     }
  }

.select_subjects_b configure -command \
  {ssdbm_subjects_list select_subjects
   if {[winfo exists .ss]} \
      { wm deiconify .ss } \
    else {list_selector .ss "subjects" intersection select_subjects \
                                  update_resources "Update"  resources_now_old}
  }

.exit_b configure -command exit_dialog
.help_b configure -command \
    {
    global url help
    set url $help
    goto_resource
    }

bind .resources <1> \
  {
  global url
  .resources select clear
  set index [.resources nearest %y]
  .resources select adjust $index
  set url [lindex $urls $index]
  .url delete 0 end
  .url insert 0 $url
  }
bind .resources <Double-1> \
  {
  global url
  .resources select clear
  set index [.resources nearest %y]
  .resources select adjust $index
  set url [lindex $urls $index]
  .url delete 0 end
  .url insert 0 $url
  goto_resource
  }


# ------- new_resource window ---------

set new_resource_subject_list {}
set new_resource_url {}
set new_resource_resource_name {}
set new_resource_entry_mode manual
set new_resource_hotlist_urls {}
set new_resource_hotlist_resources {}
```

```
proc new_resource_goto_hotlist_mode {} \
  {
  global new_resource_entry_mode
  set new_resource_entry_mode from_hotlist
  .nrw.manual_b configure -relief raised
  .nrw.from_hotlist_b configure -relief sunken
  get_new_hotlist_resources
  new_resource_goto_next_hotlist_item
  }

proc new_resource_goto_manual_mode {} \
  {
  global new_resource_entry_mode
  set new_resource_entry_mode manual
  .nrw.manual_b configure -relief sunken
  .nrw.from_hotlist_b configure -relief raised
  }

proc new_resource_ok_pressed {} \
  {
  global new_resource_subject_list new_resource_url \
     new_resource_resource_name new_resource_entry_mode
  if {[llength $new_resource_subject_list]!=0} \
    {
    if {[string length $new_resource_url]!=0} \
    {
    set templist {}
    ssdbm_new_resource $new_resource_url  $new_resource_resource_name \
                                                              $templist
    ssdbm_set_subjects_of_resource \
                          $new_resource_url  $new_resource_subject_list
    if {$new_resource_entry_mode=="manual"} \
      {
      set new_resource_subject_list {}
      set new_resource_resource_name {}
      set new_resource_url {}
      }
    if {$new_resource_entry_mode=="from_hotlist"} \
      {
      new_resource_goto_next_hotlist_item
      }
    focus .nrw.resource_new.resource
    }
    }
  }

proc new_resource_skip_pressed {} \
  {
  new_resource_goto_next_hotlist_item
  }

proc new_resource_goto_next_hotlist_item {} \
  {
  global new_resource_url new_resource_resource_name \
     new_resource_hotlist_urls new_resource_hotlist_resources \
     new_resource_subject_list
  set new_resource_subject_list {}
  set new_resource_url [lindex $new_resource_hotlist_urls 0]
  set new_resource_resource_name [lindex $new_resource_hotlist_resources 0]
  set new_resource_hotlist_urls \
    [lrange $new_resource_hotlist_urls 1 end]
  set new_resource_hotlist_resources \
    [lrange $new_resource_hotlist_resources 1 end]
  }

proc get_new_hotlist_resources {} \
  {
  global mhotlist
  global new_resource_hotlist_urls new_resource_hotlist_resources
  set new_resource_hotlist_urls {}
  set new_resource_hotlist_resources {}
  set hotlist_file [open $mhotlist]
  #skip first two lines
```

```
  gets $hotlist_file
  gets $hotlist_file

  while {[eof $hotlist_file] == 0} \
    {
    gets $hotlist_file next_url
    gets $hotlist_file next_resource
    set next_url [lindex $next_url 0]
    #if next resource doesn't already exist
    ssdbm_exists_resource $next_url resource_exists
    if {$resource_exists=="no"} \
      {
      lappend new_resource_hotlist_urls $next_url
      lappend new_resource_hotlist_resources $next_resource
      }
    }
  }


proc new_resource {} \
  {
  global new_resource_subject_list new_resource_url \
     new_resource_resource_name new_resource_entry_mode icon
  toplevel .nrw
  # nrw is Edit Resource Window
  wm title .nrw {new resource}
  wm maxsize .nrw 500 500
  wm minsize .nrw 80 80
  if {$icon!="null"}   {global icon; wm iconbitmap .nrw @$icon}

  set new_resource_subject_list {}
  set new_resource_url {}
  set new_resource_resource_name {}

  frame .nrw.mode_b_frame -borderwidth 2 -relief ridge
  resource_edit .nrw.resource_new
  .nrw.resource_new.resource configure \
                                    -textvariable  new_resource_resource_name
  .nrw.resource_new.subjects configure \
                                    -textvariable  new_resource_subject_list
  .nrw.resource_new.url configure \
                                    -textvariable  new_resource_url
  bind .nrw.resource_new.resource <2> {%W insert insert [selection get]}
  bind .nrw.resource_new.url <2> {%W insert insert [selection get]}
  bind .nrw.resource_new.subjects <2> {%W insert insert [selection get]}

  bind .nrw.resource_new.resource <Tab> \
                            {focus .nrw.resource_new.url }
  bind .nrw.resource_new.url <Tab> \
                            {focus .nrw.resource_new.subjects  }
  bind .nrw.resource_new.subjects <Tab> \
                            {focus .nrw.resource_new.resource  }
  bind .nrw.resource_new.subjects <Shift-Tab> \
                            {focus .nrw.resource_new.url}
  bind .nrw.resource_new.url <Shift-Tab> \
                            {focus .nrw.resource_new.resource  }
  bind .nrw.resource_new.resource <Shift-Tab> \
                            {focus .nrw.resource_new.subjects  }
  bind .nrw.resource_new.resource <Down> \
                            {focus .nrw.resource_new.url }
  bind .nrw.resource_new.url <Down> \
                            {focus .nrw.resource_new.subjects  }
  bind .nrw.resource_new.subjects <Down> \
                            {focus .nrw.resource_new.resource  }
  bind .nrw.resource_new.subjects <Up> \
                            {focus .nrw.resource_new.url}
  bind .nrw.resource_new.url <Up> \
                            {focus .nrw.resource_new.resource  }
  bind .nrw.resource_new.resource <Up> \
                            {focus .nrw.resource_new.subjects  }
```

```
  button .nrw.manual_b -text "    Manual     "
  button .nrw.from_hotlist_b -text " From Hotlist  "
  button .nrw.skip_b -text " Skip + Next "
  button .nrw.ok_b -text " OK + Next "
  button .nrw.dismiss_b -text " Dismiss "
  button .nrw.subjects_selector_b -text "  Select Subjects  "

  pack .nrw.manual_b -side left -expand true -fill x \
                                                  -in  .nrw.mode_b_frame
  pack .nrw.from_hotlist_b -side right -expand true -fill x \
                                                   -in  .nrw.mode_b_frame
  pack .nrw.mode_b_frame -fill x -expand true -side top -in .nrw

  pack .nrw.resource_new -expand true -side top -fill x
  pack .nrw.subjects_selector_b -side top -expand true -fill x

  pack .nrw.ok_b -side left -fill x -expand true
  pack .nrw.skip_b -side left  -fill x -expand true
  pack .nrw.dismiss_b -side left -fill x -expand true

  .nrw.from_hotlist_b configure -command new_resource_goto_hotlist_mode
  .nrw.manual_b configure -command new_resource_goto_manual_mode
  .nrw.dismiss_b configure -command {destroy .nrw; focus .intersection}
  .nrw.subjects_selector_b configure -command \
    {ssdbm_subjects_list add_list
     if {[winfo exists .nrw.subjects_selector]} \
                { wm deiconify .nrw.subjects_selector } \
       else {list_selector .nrw.subjects_selector "new subjects" \
                           new_resource_subject_list  add_list \
                           NULL NULL NULL }
    }

  .nrw.ok_b configure -command new_resource_ok_pressed
  .nrw.skip_b configure -command new_resource_skip_pressed

  bind .nrw.resource_new.subjects <Return>  new_resource_ok_pressed

  if {$new_resource_entry_mode=="from_hotlist"} \
                {new_resource_goto_hotlist_mode}
  if {$new_resource_entry_mode=="manual"} {new_resource_goto_manual_mode}
  focus .nrw.resource_new.resource
  }


# ----------- edit resource window --------------

set edit_resource_subject_list {}
set edit_resource_url {}
set edit_resource_resource_name {}

proc edit_resource {edit_url edit_resource_name}  \
  {
  global edit_resource_subject_list edit_resource_url \
                                    icon  edit_resource_resource_name
  toplevel .erw
  # erw is Edit Resource Window
  wm title .erw {edit resource}
  wm maxsize .erw 500 500
  wm minsize .erw 80 80
  if {$icon!="null"} {global icon; wm iconbitmap .erw @$icon}

  resource_edit .erw.resource_edit
  pack .erw.resource_edit -side top -expand true -fill x
  frame .erw.button_bar_1

  .erw.resource_edit.url configure -text edit_resource_url
  .erw.resource_edit.url configure -state disabled
  .erw.resource_edit.resource configure \
                                    -textvariable  edit_resource_resource_name
  .erw.resource_edit.subjects configure \
                                    -textvariable  edit_resource_subject_list

  bind .erw.resource_edit.resource <2> {%W insert insert [selection get]}
```

```
  bind .erw.resource_edit.subjects <2> {%W insert insert [selection get]}
  bind .erw.resource_edit.resource <Tab> {focus .erw.resource_edit.subjects}
  bind .erw.resource_edit.subjects <Tab> {focus .erw.resource_edit.resource}
  bind .erw.resource_edit.resource <Down> {focus  .erw.resource_edit.subjects}
  bind .erw.resource_edit.subjects <Down> {focus  .erw.resource_edit.resource}
  bind .erw.resource_edit.resource <Shift-Tab> \
                                          {focus .erw.resource_edit.subjects}
  bind .erw.resource_edit.subjects <Shift-Tab> \
                                          {focus .erw.resource_edit.resource}
  bind .erw.resource_edit.resource <Up> \
                                          {focus .erw.resource_edit.subjects}
  bind .erw.resource_edit.subjects <Up> \
                                          {focus .erw.resource_edit.resource}


  ssdbm_subjects_of_resource $edit_url edit_resource_subject_list
  set edit_resource_resource_name $edit_resource_name
  set edit_resource_url $edit_url

  button .erw.ok_b -text              "     OK     "
  button .erw.dismiss_b -text         "   Dismiss   "
  button .erw.delete_b -text          "   Delete    "
  button .erw.subjects_selector_b -text " Select Subjects "

  pack .erw.subjects_selector_b -side top -fill x -expand true \
                                                        -in  .erw.button_bar_1
  pack .erw.ok_b -fill x -expand true \
                                                        -in  .erw.button_bar_1
  pack .erw.button_bar_1 -fill x -expand true -side left
  pack .erw.delete_b -side top -fill x -expand true
  pack .erw.dismiss_b -side left -fill x -expand true

  .erw.dismiss_b configure -command {destroy .erw; focus .intersection}
  .erw.subjects_selector_b configure -command \
    {ssdbm_subjects_list add_list
     if {[winfo exists .erw.subjects_selector]} \
        { wm deiconify .erw.subjects_selector} \
      else {list_selector .erw.subjects_selector "edit subjects" \
                          edit_resource_subject_list  add_list \
                          NULL NULL NULL }
    }

  .erw.ok_b configure -command \
    {
    global edit_resource_subject_list
    if {[llength $edit_resource_subject_list]!=0} \
      {
      ssdbm_update_resource $edit_resource_url  $edit_resource_resource_name
      ssdbm_set_subjects_of_resource \
                          $edit_resource_url  $edit_resource_subject_list
      destroy .erw
      }
    }

  .erw.delete_b configure -command \
    {
    ssdbm_delete_resource $edit_resource_url
    destroy .erw
    }

  focus .erw.resource_edit.resource
  }


# ---------- resource_edit widgit --------------


proc resource_edit {widgit_name} \
  {
  #this procedure makes a resource edit widgit, for the caller to pack
  frame $widgit_name

  label $widgit_name.resource_label -text "resource name"
```

```
  entry $widgit_name.resource  -relief sunken -width 45
  label $widgit_name.url_label -text url
  entry $widgit_name.url  -relief sunken
  label $widgit_name.subjects_label -text subjects
  entry $widgit_name.subjects  -relief sunken

  pack $widgit_name.resource_label -anchor w -in $widgit_name
  pack $widgit_name.resource -fill x -in $widgit_name
  pack $widgit_name.url_label -anchor w -in $widgit_name
  pack $widgit_name.url -fill x -in $widgit_name
  pack $widgit_name.subjects_label -anchor w -in $widgit_name
  pack $widgit_name.subjects -fill x -in $widgit_name
  }


# ------- list selector window --------

proc list_selector {window label list_name add_list_name \
                    update_proc update_button_label  list_change_proc} \
  {
  # this procedure displays a general purpose list selector window,
  # and updates $list_name (a list used by caller) with selected items

  global frame_relief  frame_borderwidth  button_space
  global $list_name icon
  upvar $add_list_name add_list
  set add_list [lsort -ascii -decreasing $add_list]
  toplevel $window
  wm title $window [concat $label]
  wm maxsize $window 500 500
  wm minsize $window 80 80
  if {$icon!="null"} {global icon; wm iconbitmap $window @$icon}

  frame $window.add_frame -relief $frame_relief \
                                        borderwidth $frame_borderwidth
  frame $window.remove_frame -relief $frame_relief \
                                        borderwidth $frame_borderwidth
  frame $window.button_frame -relief $frame_relief \
                                        borderwidth $frame_borderwidth

  listbox $window.add_items -relief sunken -geometry 16x10
  scrollbar $window.add_scroll -orient vertical
  frame $window.add_label_button_frame
  label $window.add_label -text " add  "
  button $window.add_all_b -text " All  "
  pack $window.add_all_b -side right  \
                                -padx $button_space pady $button_space \
                                -in  $window.add_label_button_frame
  pack $window.add_label -side left -fill x -in  $window.add_label_button_frame
  pack $window.add_label_button_frame -fill x  -padx 16 \
                                -in  $window.add_frame
  pack $window.add_items -side left -fill both -expand true \
                                                -in  $window.add_frame
  pack $window.add_scroll -side right -fill y -in $window.add_frame

  listbox $window.remove_items -relief sunken -geometry 16x5
  scrollbar $window.remove_scroll -orient vertical
  frame $window.remove_label_button_frame
  label $window.remove_label -text  " remove  "
  button $window.remove_all_b -text " All  "
  pack $window.remove_all_b -side right \
            -padx $button_space -pady $button_space \
            -in $window.remove_label_button_frame
  pack $window.remove_label -side left -fill x \
            -in $window.remove_label_button_frame
  pack $window.remove_label_button_frame -side top -fill x \
             -padx 16 -in $window.remove_frame
  pack $window.remove_items -side left -fill both -expand true \
                                                -in  $window.remove_frame
  pack $window.remove_scroll -side right -fill y -in $window.remove_frame

  button $window.dismiss_b -text " Dismiss "
  if {$update_proc != "NULL"}  \
```

```
  { button $window.update_b -text $update_button_label }
 pack $window.dismiss_b -side left -fill x -expand true \
              -padx $button_space -pady $button_space \
              -in $window.button_frame
 if {$update_proc != "NULL"}  \
  {pack $window.update_b -side left -fill x -expand true \
              -padx $button_space -pady $button_space \
              -in $window.button_frame }

 foreach i $add_list {$window.add_items insert 0 $i}
 foreach i [expr $$list_name] {$window.remove_items insert 0  $i}

 pack $window.add_frame -side top -expand true -fill both
 pack $window.button_frame -fill x
 pack $window.remove_frame -side bottom -expand true -fill both

 $window.add_scroll configure -command [concat $window.add_items yview]
 $window.remove_scroll configure -command [concat $window.remove_items  yview]
 $window.add_items configure -yscrollcommand [concat $window.add_scroll  set]
 $window.remove_items configure \
                        -yscrollcommand [concat  $window.remove_scroll set]


 $window.dismiss_b configure -command [concat destroy $window {; return}]

 if {$update_proc != "NULL"} \
   {$window.update_b configure -command $update_proc}
 if {$list_change_proc!= "NULL"}  \
   {$window.add_all_b configure -command \
   [concat selector_add_all $list_name $window ";"  $list_change_proc] }

 if {$list_change_proc == "NULL"} \
   {
   $window.add_all_b configure -command \
              [concat selector_add_all $list_name $window ]
   $window.remove_all_b configure -command \
              [concat selector_remove_all $list_name $window ]
   bind $window.add_items <1> \
              [concat selector_add $list_name $window %y]
   bind $window.remove_items <1> \
              [concat selector_remove $list_name $window %y]
   }
 if {$list_change_proc != "NULL"} \
   {
   $window.add_all_b configure -command \
              [concat selector_add_all $list_name $window \
                                                  ";"  $list_change_proc ]
   $window.remove_all_b configure -command \
              [concat selector_remove_all $list_name $window \
                                                  ";"  $list_change_proc]
   bind $window.add_items <1> \
              [concat selector_add $list_name $window %y \
                                                  ";"  $list_change_proc]
   bind $window.remove_items <1> \
              [concat selector_remove $list_name $window %y \
                                                  ";"  $list_change_proc]
   }



 }


proc selector_add {list_name selector_window y} \
   {
   global $list_name
   set index [$selector_window.add_items nearest $y]
   set item [$selector_window.add_items get $index]
   $selector_window.add_items select clear
   set $list_name [concat [expr $$list_name] $item]
   $selector_window.add_items delete $index
   $selector_window.remove_items insert 0 $item
   }
```

```
proc selector_remove {list_name selector_window y} \
  {
  global $list_name
  set index [$selector_window.remove_items nearest $y]
  set item [$selector_window.remove_items get $index]
  $selector_window.remove_items select clear
  set item_to_remove [lsearch [expr $$list_name] $item]
  set $list_name [concat [lrange [expr $$list_name] \
                                        0 [expr  $item_to_remove-1]] \
                              [lrange [expr $$list_name] \
                                        [expr  $item_to_remove+1] \
                                        [llength [expr  $$list_name]] ]]
  $selector_window.remove_items delete $index
  $selector_window.add_items insert 0 $item
  }

proc selector_add_all {list_name selector_window} \
  {
  global $list_name
  for  {set n [$selector_window.add_items size]} \
    {$n>0} {incr n -1} \
    {selector_add $list_name $selector_window 12}
  }

proc selector_remove_all {list_name selector_window} \
  {
  global $list_name
  for  {set n [$selector_window.remove_items size]} \
    {$n>0} {incr n -1} \
    {selector_remove $list_name $selector_window 12}
  }


# ------------ MAIN ---------------

set_current_dir
display_main_window

#bind . <Unmap> \
#       {
#       if {$icon!="null"} {global icon; wm iconbitmap . @$icon}
#
#       }
```

15 smosaic

```
#! /bin/sh

TCL_LIBRARY='../tcl/library'
TK_LIBRARY='../tk/library'
export TCL_LIBRARY
export TK_LIBRARY

rm -f ./users/student/undergrad/zgee9119/.my-ss.subject~
rm -f ./users/student/undergrad/zgee9119/.my-ss.record~
cp /users/student/undergrad/zgee9119/.my-ss.subject /users/student/
undergrad/zgee9119/.my-ss.subject~
cp /users/student/undergrad/zgee9119/.my-ss.record /users/student/undergrad/
zgee9119/.my-ss.record~

mosaic &
```

```
./simon -path . -file simple-simon.tcl -ss /users/student/undergrad/zgee9119/ .my-ss
-icon simon.bit -mpid $! &
```

16 smosaic.skel

```
#! /bin/sh

TCL_LIBRARY='_TCL_LIB_'
TK_LIBRARY='_TK_LIB_'
export TCL_LIBRARY
export TK_LIBRARY

rm -f _PATH__DEFAULT_SS_.subject~
rm -f _PATH__DEFAULT_SS_.record~
cp _DEFAULT_SS_.subject _DEFAULT_SS_.subject~
cp _DEFAULT_SS_.record _DEFAULT_SS_.record~

_MOSAIC_ &
_PATH_/simon -path _PATH_ -file simple-simon.tcl -ss _DEFAULT_SS_ -icon
simon.bit -mpid $! &
```

17 ss.h

```
#ifndef _ss_h
#define _ss_h 1


#include <String.h>
#include "ByteStore.h"
#include "functions.h"

enum Add_Flag { TEST_FOR_DUPLICATE, NO_TEST_FOR_DUPLICATE };


class Subject;
class Resource;
class Record;

// replacements for Set<T>


class Set_String_
  {
  public:

  //usual stuff for dynamic class:

    Set_String_ (unsigned expand_rate_=10);  // contructor
    ~Set_String_ ();   // destructor
    Set_String_ (const Set_String_& original);  // copy constructor
    void operator=(const Set_String_& original);

  //usual stuff for container classes:

    void add_member (const String& object,
                                        const Add_Flag flag  = TEST_FOR_DUPLICATE);
    int has_member(const String& object); // 1=true 0=flase
    void delete_member (const String& object); //beware, this cause  an order
                                                                  // change
    void empty(); //deletes all members
```

```
  //use these to get objects out, use a for loop with no_of_members(),
  //or a while loop with can_get_next_member().
  //always call next_member_returns_first just before the loop, to reset
  //the internal itorator, so get_next_member starts with the first member

    int no_of_members (void) const {return last_member+1;};
    int can_use_get_next_member (void) const
                {return (last_accessed_member<last_member);};
    void next_member_returns_first(void)
                {last_accessed_member=-1;};   // doesn't change  members
    String get_first_member (void); // doesn't change members
    String get_next_member (void);  // doesn't change members
    void get_next_member (String& object);
    void get_first_member (String& object);

    int operator== (Set_String_& set_b);
    void operator+= (const Set_String_& set_b);   //union_ with  assign
    void operator*= (const Set_String_& set_b);   //intersect with  assign
    void operator-= (const Set_String_& set_b);   //difference with  assign
    Set_String_ operator+ (const Set_String_& set_b)
                {Set_String_ result=(*this); result+=set_b;  return result;};
    Set_String_ operator* (const Set_String_& set_b)
                {Set_String_ result=(*this); result*=set_b;  return result;};
    Set_String_ operator- (const Set_String_& set_b)
                {Set_String_ result=(*this); result-=set_b;  return result;};


  //for memory management - ignore it if you like, but the class won't
  //bother deallocating until its destructor is called

    void shrink();  // deallocates all memory for spare object  slots
    void resize_to (unsigned newsize);    // to make it bigger, or
                //smaller, but you can't use it as a short cut to
                //deleting members
    int allocated_size() {return size;}; // how many objects it has  room
                                                    // for in total

  //usefull
    void screen_dump();  //use for debugging
    void fill_with (String array[], int start_index, int  end_index);


  protected:
    void do_delete_member (const int index);
    int find_member(const String& possible_member) const;
    unsigned expand_rate;
    int size;
    int last_member;
    int last_accessed_member;
    String** object_array;
  };

typedef Set_String_ Node_id; //Node_id is a set of subject_id's


class Set_Resource_
  {
  public:

  //usual stuff for dynamic class:

    Set_Resource_ (unsigned expand_rate_=10);  // contructor
    ~Set_Resource_ ();   // destructor
    Set_Resource_ (const Set_Resource_& original);  // copy  constructor
    void operator=(const Set_Resource_& original);

  //usual stuff for container classes:

    void add_member (const Resource& object,
                                        const Add_Flag  flag = TEST_FOR_DUPLICATE);
    int has_member(const Resource& object); // 1=true 0=flase
    void delete_member (const Resource& object); // beware, this  cause an order
                                                    //change
    void empty(); //deletes all members
```

```
  //use these to get objects out, use a for loop with no_of_members(),
  //or a while loop with can_get_next_member().
  //always call next_member_returns_first just before the loop, to reset
  //the internal itorator, so get_next_member starts with the first member

    int no_of_members (void) const {return last_member+1;};
    int can_use_get_next_member (void) const
                {return (last_accessed_member<last_member);};
    void next_member_returns_first(void)
                {last_accessed_member=-1;};   // doesn't change  members
    Resource get_first_member (void); // doesn't change members
    Resource get_next_member (void);  // doesn't change members
    void get_next_member (Resource& object);
    void get_first_member (Resource& object);

    void operator+= (const Set_Resource_& set_b);   //union_ with  assign
    void operator*= (const Set_Resource_& set_b);   //intersect  with assign
    void operator-= (const Set_Resource_& set_b);   //difference  with assign
    Set_Resource_ operator+ (const Set_Resource_& set_b)
                {Set_Resource_ result=(*this); result+=set_b;  return result;};
    Set_Resource_ operator* (const Set_Resource_& set_b)
                {Set_Resource_ result=(*this); result*=set_b;  return result;};
    Set_Resource_ operator- (const Set_Resource_& set_b)
                {Set_Resource_ result=(*this); result-=set_b;  return result;};


  //for memory management - ignore it if you like, but the class won't
  //bother deallocating until its destructor is called

    void shrink();  // deallocates all memory for spare object  slots
    void resize_to (unsigned newsize);    // to make it bigger, or
                //smaller, but you can't use it as a short cut to
                //deleting members
    int allocated_size() {return size;}; // how many objects it has  room
                                                    // for in total

  //usefull
    void screen_dump();  //use for debugging
    void fill_with (Resource array[], int start_index, int  end_index);


  protected:
    void do_delete_member (const int index);
    int find_member(const Resource& possible_member) const;
    unsigned expand_rate;
    int size;
    int last_member;
    int last_accessed_member;
    Resource** object_array;
  };



class Subject
  {
  public:
    String subject_id;
    Set_String_ urls;
    Set_String_ child_subjects;
    Set_String_ parent_subjects;
  };


class Resource
  {
  public:
    String url;
    String name;
    //String description;
    void fill_with (String array[], int start_index)
                {
                int i=start_index;
```

```
                url=array[i++];
                name=array[i++];
                //description=array[i++];
                };
    int operator== (const Resource& res_b)
                {return url==res_b.url;};
    void operator= (const Resource& orig)
                {url=orig.url; name=orig.name;
                        / *description=orig.description;*/ };
  };


class Record
  {
  public:
    Resource resource;
    Set_String_ subject_ids;
    Record() {};
    Record (Resource resource_, Set_String_ subject_ids_)
                {resource=resource_;  subject_ids=subject_ids_;};
  };



/*
class RelatedSubject
  {
  public:
    String subject_id;
    int no_of_common_resources;
    //int link_use;
    //int link_agreement;
    int operator!= (const RelatedSubject& other);
    RelatedSubject(const RelatedSubject& original)
                 {no_of_common_resources=original.no_of_common_resources;
                 subject_id=original.subject_id;};
    RelatedSubject() {};

    void operator=(const RelatedSubject& rs)
                {subject_id=rs.subject_id;
                 no_of_common_resources=rs.no_of_common_resources;};
  };

int RelatedSubject::operator!= (const RelatedSubject& other)
    {
    return (subject_id!=other.subject_id);
    };

int Set_RelatedSubject_::find_member(const RelatedSubject& possible_member)
                                                                                const
  //returns the member number or -1
  {
  int n=last_member;
  while ( n!=(-1) &&
        ((*object_array[n]).subject_id!=possible_member.subject_id))
    {n--;};
  return n;
  };
*/


class SSByteStore : public ByteStore
  {
  public:
    SSByteStore(int start_size, int expand_by_)
      :ByteStore(start_size, expand_by_) {;}; //just calls base  constructor
    void store(Set_String_& string_set);
    void store(Subject& subject);
    void store(Record& record);
    void get(Set_String_& string_set);
    void get(Subject& subject);
    void get(Record& record);

    // these are needed as the fns above don't overload base fns,
```

```
    // they hide them.
    void store(String& string)
                {ByteStore::store(string);};
    void get(String& string)
                {ByteStore::get(string);};

  };


void string_set_to_string(Set_String_ set, String& string);

void resource_set_to_string(Set_Resource_ set, String& string);



class Set_Set_String_
  {
  public:

  //usual stuff for dynamic class:

    Set_Set_String_ (unsigned expand_rate_=10);  // contructor
    ~Set_Set_String_ ();   // destructor
    Set_Set_String_ (const Set_Set_String_& original);  // copy  constructor
    void operator=(const Set_Set_String_& original);

  //usual stuff for container classes:

    void add_member (const Set_String_& object,
                                        const Add_Flag  flag = TEST_FOR_DUPLICATE);
    int has_member(const Set_String_& object); // 1=true 0=flase
    void delete_member (const Set_String_& object); // beware, this  cause an order
                                                    //change
    void empty(); //deletes all members

  //use these to get objects out, use a for loop with no_of_members(),
  //or a while loop with can_get_next_member().
  //always call next_member_returns_first just before the loop, to reset
  //the internal itorator, so get_next_member starts with the first member

    int no_of_members (void) const {return last_member+1;};
    int can_use_get_next_member (void) const
                {return (last_accessed_member<last_member);};
    void next_member_returns_first(void)
                {last_accessed_member=-1;};   // doesn't change  members
    Set_String_ get_first_member (void); // doesn't change members
    Set_String_ get_next_member (void);  // doesn't change members
    void get_next_member (Set_String_& object);
    void get_first_member (Set_String_& object);

    void operator+= (const Set_Set_String_& set_b);   //union_ with  assign
    void operator*= (const Set_Set_String_& set_b);   //intersect  with assign
    void operator-= (const Set_Set_String_& set_b);   //difference  with assign
    Set_Set_String_ operator+ (const Set_Set_String_& set_b)
                {Set_Set_String_ result=(*this); result+=set_b;  return result;};
    Set_Set_String_ operator* (const Set_Set_String_& set_b)
                {Set_Set_String_ result=(*this); result*=set_b;  return result;};
    Set_Set_String_ operator- (const Set_Set_String_& set_b)
                {Set_Set_String_ result=(*this); result-=set_b;  return result;};


  //for memory management - ignore it if you like, but the class won't
  //bother deallocating until its destructor is called

    void shrink();  // deallocates all memory for spare object  slots
    void resize_to (unsigned newsize);    // to make it bigger, or
                //smaller, but you can't use it as a short cut to
                //deleting members
    int allocated_size() {return size;}; // how many objects it has  room
                                                    // for in total

  //usefull
    void screen_dump();  //use for debugging
    void fill_with (Set_String_ array[], int start_index, int  end_index);
```

```
  protected:
    void do_delete_member (const int index);
    int find_member(const Set_String_& possible_member) const;
    unsigned expand_rate;
    int size;
    int last_member;
    int last_accessed_member;
    Set_String_** object_array;
  };

typedef Set_Set_String_ Set_Node_id_; //Node_id is a set of subject_id's


#endif
```

18 ss.cxx

```
#include <iostream.h>
#include <stdlib.h>
#include <String.h>
#include "ss.h"

void string_set_to_string(Set_String_ set, String& string)
  {
  set.next_member_returns_first();
  while (set.can_use_get_next_member())
    cat (string, quoted(set.get_next_member()), " ", string);
  };

void resource_set_to_string(Set_Resource_ set, String& string)
  {
  Resource temp;
  set.next_member_returns_first();
  while (set.can_use_get_next_member())
    {
    temp=set.get_next_member();
    cat (string, quoted(temp.url), " ", string);
    cat (string, quoted(temp.name), " ", string);
    //cat (string, quoted(temp.description), " ", string);
    };
  };



void Set_String_::fill_with
    (String array[], int start_index, int end_index)
  {
  for (int n=start_index; n<=end_index; n++)
    add_member(array[n]);
  };

Set_String_::Set_String_ (unsigned expand_rate_=10)
  {
  expand_rate=expand_rate_;
  size=0;
  last_member=-1;
  last_accessed_member=-1;
  object_array=NULL;   // no point in mallocing them memory yet, as the set
                       // has no members, so we must NULL  them so as not to
                       // confuse other member functions,  eg empty
  };


Set_String_::Set_String_ (const Set_String_& original)
  {
```

```
  expand_rate=original.expand_rate;
  size=0;
  last_member=-1;
  last_accessed_member=-1;
  object_array=NULL;
  operator=(original);
  };


Set_String_::~Set_String_ ()
  {
  for (int n=0; n<=last_member; n++)
    {
    delete(object_array[n]);   // delete memory of a set member
    };
  if (object_array!=NULL) free (object_array); //delete object pointer array
  };


void Set_String_::resize_to (unsigned newsize)
  {
  if (newsize>size)   // ie only resize if we need to
    {
    // this if is necessary as some versions of realloc
    // don't work with NULL pointers
    if (object_array==NULL)
    object_array = (String**)
                              malloc((newsize+expand_rate+1)*sizeof(String**));
     else
    object_array = (String**) realloc(object_array,
                                      (newsize+expand_rate+1)*sizeof(String**) );
     size=newsize+expand_rate+1;
     };
  };

void Set_String_::add_member (const String& object,
                                      const Add_Flag  flag = TEST_FOR_DUPLICATE)
  {
  if (flag==NO_TEST_FOR_DUPLICATE)
    {
    resize_to(no_of_members()+1);   // make sure theres room
    object_array[++last_member] = new(String); // make a new object  & set up
                                                // a pointer to it after the
                                                // last pointer in object_array
    *object_array[last_member] = object; //copy the object to our  new one
    }
   else
    {
    if (find_member(object)==-1)  // ie can't have duplicates in a set!
    {
    resize_to(no_of_members()+1);   // make sure theres room
    object_array[++last_member] = new(String); // make a new object  & set up
                                                // a pointer to it after the
                                                // last pointer in object_array
    *object_array[last_member] = object; //copy the object to our  new one
    };
    };
  };

void Set_String_::get_next_member (String& object)
  {
  if (last_accessed_member<last_member)
    {
    object=*object_array[++last_accessed_member];
    };
  };

String Set_String_::get_next_member()
  {
  if (last_accessed_member<last_member)
    return (*object_array[++last_accessed_member]);
  else {String dummy; return dummy;};
  };
```

```
void Set_String_::get_first_member (String& object)
  {
  next_member_returns_first();
  get_next_member(object);
  };

String Set_String_::get_first_member ()
  {
  next_member_returns_first();
  return (get_next_member());
  };


void Set_String_::empty()
  {
  for (int n=0; n<=last_member; n++)
    {
    delete(object_array[n]);  // delete a set member
    };
  last_member=-1;
  };

void Set_String_::shrink()
  {
  if (size==0) // if the set is empty, just free up all allocated memory
    {
    if (object_array!=NULL) {free (object_array);  object_array=NULL;};
    }
    else     // free up memory allocated for spare pointer & mv's
    {
    if (object_array!=NULL)
      object_array=(String**)realloc(object_array,  sizeof(String**)*(last_member+1));
    };
  size=last_member+1;
  };

void Set_String_::operator=(const Set_String_& original)
  {
  empty();
  for (int n=0; n<=original.last_member; n++)
    {
    add_member(*original.object_array[n], NO_TEST_FOR_DUPLICATE);
    };
  };


int Set_String_::find_member(const String& possible_member) const
  {
  // this returns -1 if it can't find the member, else the object_array
  // index for the member
  int n=last_member;
  while (n>-1)
    {
//cout <<"C: "<< *object_array[n]<<"  "<<possible_member<<"\n";
    if (*object_array[n]==possible_member) break;
    n--;
    };
  return n;
  };


void Set_String_::delete_member (const String& object)
  {
  // delete member if we are storing it
  do_delete_member (find_member(object));
  };

void Set_String_::do_delete_member (const int index)
  {
  if (index!=-1)
    {
    delete(object_array[index]);
    object_array[index]=object_array[last_member];
    last_member--;
```

```
    };
  };


int Set_String_::has_member(const String& object)
  {return( (find_member(object)!=-1) ? 1 : 0);};


void Set_String_::operator+= (const Set_String_& set_b)
  {
  //int this_index;
  for (int n=0; n<=set_b.last_member; n++)
    {
    //this_index=find_member(*set_b.object_array[n]);
    //if (this_index==-1)
        add_member(*set_b.object_array[n]);
    };
  };

void Set_String_::operator-= (const Set_String_& set_b)
  {
  //int this_index;
  for (int n=0; n<=set_b.last_member; n++)
    {
    //this_index=find_member(*set_b.object_array[n]);
    //if (this_index==-1)
        delete_member(*set_b.object_array[n]);
    };
  };


int Set_String_::operator== (Set_String_& set_b)
  {
  if (no_of_members()!=set_b.no_of_members()) return 0;
  next_member_returns_first();
  set_b.next_member_returns_first();
  while (can_use_get_next_member())
    {
    if ( !set_b.has_member(get_next_member()) ) return 0;
    };
  return 1;
  };

void Set_String_::operator*= (const Set_String_& set_b)
  {
  int b_index;
  next_member_returns_first();
  for (int n=last_member; n>=0; n--)    //NB last_member changing too!
    {
    b_index=set_b.find_member(*object_array[n]);
    if (b_index==-1)
        do_delete_member(n);
    };
  };


void Set_String_::screen_dump()  //use for debugging
  {
  cout << '\n';
  for (int n=0; n<=last_member; n++)
    {
    cout << *object_array[n] << "  " << '\n';
    };
  cout << '\n';
  };




void Set_Resource_::fill_with
    (Resource array[], int start_index, int end_index)
  {
  for (int n=start_index; n<=end_index; n++)
```

```
    add_member(array[n]);
  };

Set_Resource_::Set_Resource_ (unsigned expand_rate_=10)
  {
  expand_rate=expand_rate_;
  size=0;
  last_member=-1;
  last_accessed_member=-1;
  object_array=NULL;   // no point in mallocing them memory yet, as the set
                       // has no members, so we must NULL  them so as not to
                       // confuse other member functions,  eg empty
  };


Set_Resource_::Set_Resource_ (const Set_Resource_& original)
  {
  expand_rate=original.expand_rate;
  size=0;
  last_member=-1;
  last_accessed_member=-1;
  object_array=NULL;
  operator=(original);
  };


Set_Resource_::~Set_Resource_ ()
  {
  for (int n=0; n<=last_member; n++)
    {
    delete(object_array[n]);   // delete memory of a set member
    };
  if (object_array!=NULL) free (object_array); //delete object pointer array
  };


void Set_Resource_::resize_to (unsigned newsize)
  {
  if (newsize>size)   // ie only resize if we need to
    {
    // this if is necessary as some versions of realloc
    // don't work with NULL pointers
    if (object_array==NULL)
    object_array = (Resource**)
                        malloc((newsize+expand_rate+1)*sizeof(Resource**));
     else
    object_array = (Resource**) realloc(object_array,
                                (newsize+expand_rate+1)*sizeof(Resource**) );
     size=newsize+expand_rate+1;
     };
  };

void Set_Resource_::add_member (const Resource& object,
                                const Add_Flag  flag = TEST_FOR_DUPLICATE)
  {
  if (flag==NO_TEST_FOR_DUPLICATE)
    {
    resize_to(no_of_members()+1);   // make sure theres room
    object_array[++last_member] = new(Resource);
                                                // make a new object & set up
                                                // a pointer to it after the
                                                // last pointer in object_array
    *object_array[last_member] = object; //copy the object to our  new one
    }
   else
    {
    if (find_member(object)==-1)  // ie can't have duplicates in a set!
    {
    resize_to(no_of_members()+1);   // make sure theres room
    object_array[++last_member] = new(Resource);
                                                // make a new object & set up
                                                // a pointer to it after the
                                                // last pointer in object_array
    *object_array[last_member] = object; //copy the object to our  new one
```

```
    };
    };
  };

void Set_Resource_::get_next_member (Resource& object)
  {
  if (last_accessed_member<last_member)
    {
    object=*object_array[++last_accessed_member];
    };
  };

Resource Set_Resource_::get_next_member()
  {
  if (last_accessed_member<last_member)
    return (*object_array[++last_accessed_member]);
  else {Resource dummy; return dummy;};
  };

void Set_Resource_::get_first_member (Resource& object)
  {
  next_member_returns_first();
  get_next_member(object);
  };

Resource Set_Resource_::get_first_member ()
  {
  next_member_returns_first();
  return (get_next_member());
  };


void Set_Resource_::empty()
  {
  for (int n=0; n<=last_member; n++)
    {
    delete(object_array[n]);  // delete a set member
    };
  last_member=-1;
  };

void Set_Resource_::shrink()
  {
  if (size==0) // if the set is empty, just free up all allocated memory
    {
    if (object_array!=NULL) {free (object_array);  object_array=NULL;};
    }
    else     // free up memory allocated for spare pointer & mv's
    {
    if (object_array!=NULL)
      object_array=(Resource**)realloc(object_array,
sizeof(Resource**)*(last_member+1));
    };
  size=last_member+1;
  };

void Set_Resource_::operator=(const Set_Resource_& original)
  {
  empty();
  for (int n=0; n<=original.last_member; n++)
    {
    add_member(*original.object_array[n], NO_TEST_FOR_DUPLICATE);
    };
  };


int Set_Resource_::find_member(const Resource& possible_member) const
  {
  // this returns -1 if it can't find the member, else the object_array
  // index for the member
  int n=last_member;
  while (n>-1)
    {
    if (*object_array[n]==possible_member) break;
```

```
    n--;
    };
  return n;
  };


void Set_Resource_::delete_member (const Resource& object)
  {
  // delete member if we are storing it
  do_delete_member (find_member(object));
  };

void Set_Resource_::do_delete_member (const int index)
  {
  if (index!=-1)
    {
    delete(object_array[index]);
    object_array[index]=object_array[last_member];
    last_member--;
    };
  };


int Set_Resource_::has_member(const Resource& object)
  {return( (find_member(object)!=-1) ? 1 : 0);};


void Set_Resource_::operator+= (const Set_Resource_& set_b)
  {
  //int this_index;
  for (int n=0; n<=set_b.last_member; n++)
    {
    //this_index=find_member(*set_b.object_array[n]);
    //if (this_index==-1)
        add_member(*set_b.object_array[n]);
    };
  };

void Set_Resource_::operator-= (const Set_Resource_& set_b)
  {
  //int this_index;
  for (int n=0; n<=set_b.last_member; n++)
    {
    //this_index=find_member(*set_b.object_array[n]);
    //if (this_index==-1)
        delete_member(*set_b.object_array[n]);
    };
  };


void Set_Resource_::operator*= (const Set_Resource_& set_b)
 {
 int b_index;
 next_member_returns_first();
 for (int n=last_member; n>=0; n--)    //NB last_member changing too!
    {
    b_index=set_b.find_member(*object_array[n]);
    if (b_index==-1)
        do_delete_member(n);
    };
  };


void Set_Resource_::screen_dump()  //use for debugging
  {
  cout << '\n';
  for (int n=0; n<=last_member; n++)
    {
    cout << (*object_array[n]).url << "  " << '\n';
    cout << (*object_array[n]).name << "  " << '\n';
    //cout << (*object_array[n]).description << "  " << '\n';
    };
  cout << '\n';
  };
```

```
void SSByteStore::store(Set_String_& string_set)
  {
  ByteStore::store (string_set.no_of_members());
  string_set.next_member_returns_first();
  for (int n=1; n<=string_set.no_of_members(); n++)
    ByteStore::store(string_set.get_next_member());
  };

void SSByteStore::store(Subject& subject)
  {
  ByteStore::store(subject.subject_id);
  store(subject.urls);
  store(subject.child_subjects);
  store(subject.parent_subjects);
  };

void SSByteStore::store(Record& record)
  {
  ByteStore::store(record.resource.name);
  //ByteStore::store(record.resource.description);
  store(record.subject_ids);
  };

void SSByteStore::get(Set_String_& string_set)
  {
  int members=get_int();
  string_set.empty();
  string_set.resize_to(members);
  String temp;
  for (int n=1; n<=members; n++)
    {
    get(temp);
    string_set.add_member(temp, NO_TEST_FOR_DUPLICATE);
    };
  };

void SSByteStore::get(Subject& subject)
  {
  ByteStore::get(subject.subject_id);
  get(subject.urls);
  get(subject.child_subjects);
  get(subject.parent_subjects);
  };

void SSByteStore::get(Record& record)
  {
  ByteStore::get(record.resource.name);
  //ByteStore::get(record.resource.description);
  get(record.subject_ids);
  };




void Set_Set_String_::fill_with
    (Set_String_ array[], int start_index, int end_index)
  {
  for (int n=start_index; n<=end_index; n++)
    add_member(array[n]);
  };

Set_Set_String_::Set_Set_String_ (unsigned expand_rate_=10)
  {
  expand_rate=expand_rate_;
  size=0;
  last_member=-1;
```

```
  last_accessed_member=-1;
  object_array=NULL;   // no point in mallocing them memory yet, as the set
                       // has no members, so we must NULL  them so as not to
                       // confuse other member functions,  eg empty
  };


Set_Set_String_::Set_Set_String_ (const Set_Set_String_& original)
  {
  expand_rate=original.expand_rate;
  size=0;
  last_member=-1;
  last_accessed_member=-1;
  object_array=NULL;
  operator=(original);
  };


Set_Set_String_::~Set_Set_String_ ()
  {
  for (int n=0; n<=last_member; n++)
    {
    delete(object_array[n]);   // delete memory of a set member
    };
  if (object_array!=NULL) free (object_array); //delete object pointer array
  };


void Set_Set_String_::resize_to (unsigned newsize)
  {
  if (newsize>size)   // ie only resize if we need to
    {
    // this if is necessary as some versions of realloc
    // don't work with NULL pointers
    if (object_array==NULL)
    object_array = (Set_String_**)
                          malloc((newsize+expand_rate+1)*sizeof(Set_String_**));
     else
    object_array = (Set_String_**) realloc(object_array,
                                  (newsize+expand_rate+1)*sizeof(Set_String_**) );
     size=newsize+expand_rate+1;
     };
  };

void Set_Set_String_::add_member (const Set_String_& object,
                                  const Add_Flag  flag = TEST_FOR_DUPLICATE)
  {
  if (flag==NO_TEST_FOR_DUPLICATE)
    {
    resize_to(no_of_members()+1);   // make sure theres room
    object_array[++last_member] = new(Set_String_);
                                                  // make a new object & set up
                                                  // a pointer to it after the
                                                  // last pointer in object_array
    *object_array[last_member] = object; //copy the object to our  new one
    }
   else
    {
    if (find_member(object)==-1)  // ie can't have duplicates in a set!
    {
    resize_to(no_of_members()+1);   // make sure theres room
    object_array[++last_member] = new(Set_String_);
                                                  // make a new object & set up
                                                  // a pointer to it after the
                                                  // last pointer in object_array
    *object_array[last_member] = object; //copy the object to our  new one
    };
    };
  };

void Set_Set_String_::get_next_member (Set_String_& object)
  {
  if (last_accessed_member<last_member)
    {
```

```
    object=*object_array[++last_accessed_member];
    };
  };

Set_String_ Set_Set_String_::get_next_member()
  {
  if (last_accessed_member<last_member)
    return (*object_array[++last_accessed_member]);
  else {Set_String_ dummy; return dummy;};
  };

void Set_Set_String_::get_first_member (Set_String_& object)
  {
  next_member_returns_first();
  get_next_member(object);
  };

Set_String_ Set_Set_String_::get_first_member ()
  {
  next_member_returns_first();
  return (get_next_member());
  };


void Set_Set_String_::empty()
  {
  for (int n=0; n<=last_member; n++)
    {
    delete(object_array[n]);  // delete a set member
    };
  last_member=-1;
  };

void Set_Set_String_::shrink()
  {
  if (size==0) // if the set is empty, just free up all allocated memory
    {
    if (object_array!=NULL) {free (object_array);  object_array=NULL;};
    }
    else     // free up memory allocated for spare pointer & mv's
    {
    if (object_array!=NULL)
      object_array=(Set_String_**)realloc(object_array,
sizeof(Set_String_**)*(last_member+1));
    };
  size=last_member+1;
  };

void Set_Set_String_::operator=(const Set_Set_String_& original)
  {
  empty();
  for (int n=0; n<=original.last_member; n++)
    {
    add_member(*original.object_array[n], NO_TEST_FOR_DUPLICATE);
    };
  };


int Set_Set_String_::find_member(const Set_String_& possible_member) const
  {
  // this returns -1 if it can't find the member, else the object_array
  // index for the member
  int n=last_member;
  while (n>-1)
    {
    if (*object_array[n]==possible_member) break;
    n--;
    };
  return n;
  };


void Set_Set_String_::delete_member (const Set_String_& object)
  {
```

```
  // delete member if we are storing it
  do_delete_member (find_member(object));
  };

void Set_Set_String_::do_delete_member (const int index)
  {
  if (index!=-1)
    {
    delete(object_array[index]);
    object_array[index]=object_array[last_member];
    last_member--;
    };
  };


int Set_Set_String_::has_member(const Set_String_& object)
  {return( (find_member(object)!=-1) ? 1 : 0);};


void Set_Set_String_::operator+= (const Set_Set_String_& set_b)
  {
  //int this_index;
  for (int n=0; n<=set_b.last_member; n++)
    {
    //this_index=find_member(*set_b.object_array[n]);
    //if (this_index==-1)
        add_member(*set_b.object_array[n]);
    };
  };

void Set_Set_String_::operator-= (const Set_Set_String_& set_b)
  {
  //int this_index;
  for (int n=0; n<=set_b.last_member; n++)
    {
    //this_index=find_member(*set_b.object_array[n]);
    //if (this_index==-1)
        delete_member(*set_b.object_array[n]);
    };
  };


void Set_Set_String_::operator*= (const Set_Set_String_& set_b)
  {
 int b_index;
 next_member_returns_first();
 for (int n=last_member; n>=0; n--)    //NB last_member changing too!
    {
    b_index=set_b.find_member(*object_array[n]);
    if (b_index==-1)
        do_delete_member(n);
    };
  };


void Set_Set_String_::screen_dump()  //use for debugging
  {
  cout << '\n';
  for (int n=0; n<=last_member; n++)
    {
    (*object_array[n]).screen_dump();
    };
  cout << '\n';
  };
```

19 ssbrowser.cxx

```
#include <iostream.h>
#include <gdbm.h>
#include <stdio.h>
#include "ssdbm.h"
#include "ss.h"
#include "ssci-lang.h"

// ************************************************************************ *
// IMPORTANT -- THESE ARE INPUT COMMAND SIZE LIMITS ---
// ************************************************************************ *

const BUFSIZE = 4000; // max size of request in chars
String strings[500];    // this should be in service_request, but the compiler
                        // doesn't like it there!  :-(

// ************************************************************************ **


String string_token;  // used to pass the value of strings back from the lexer
char* flex_input_ptr;     // flex reads from here, and advances this ptr
                          // NB flex must read a '\n'  before it stops!!!

SS_Error service_request (SS_Map_Manager& ssmm, char* request, String&  result,
                                                        String& message);

extern int yylex();  // declaration of lexer - yylex() will actualy return
                     // an SS_Token



int main(int argc, char* argv[])
  {
  if (argc!=2) cout <<"arg 1 = database name \n";
  else
    {
    SS_Error error;
    int file_mode;
    char in_buffer[BUFSIZE];
    SS_Map_Manager ssmm(error, argv[1], GDBM_WRCREAT);
    if (error==OK)
      {
      if (setvbuf(stdin, NULL, _IOLBF, 0) != 0)
                {cerr<<"setvbuf error\n"; exit -1;};
      while (fgets(in_buffer, BUFSIZE, stdin) != NULL)
    {
    String result;
    String message;
    error=service_request(ssmm, in_buffer, result, message);
    if (result!="")
      {
      if (result.lastchar()!='\n') result+='\n';
      while (result.elem(result.length()-2)==' ')
                              result.del(result.length()-2,1);
      cout << result;
      }
     else cout << '\n';
    cout.flush();
    if (message!="")
      {
      cerr <<argv[0]<<": "<<message<<" : "<<in_buffer<<'\n';
      cerr.flush();
      };
    }
      }
     else
    {cerr<<argv[0]<< " error opening subject space - can only open  subject space by one
program at a time\n";};
   };
  };



SS_Error service_request
```

```
                (SS_Map_Manager& ssmm, char* request_ptr,  String& result,
                                                          String& message)
// request_ptr must point into a char buffer ending in '\n'
 {
flex_input_ptr=request_ptr;
SS_Error error;

 // find out what the service request is
SS_Token service = (SS_Token) yylex();
 // parse the request data, an unknown number of strings
int last_string=-1;

SS_Token token = service;
while ((token!=EOL))
  {
  token = (SS_Token) yylex();
  if (token==STRING) strings[++last_string]=string_token;
  };

int no_of_strings=last_string+1;

 // process the service request
switch (service)
  {
  case SUBJECTS_LIST:
  {
  if (no_of_strings==0)
    {
    Set_String_ subject_ids;
    error = ssmm.subjects_list(subject_ids);
    if (error==OK) string_set_to_string(subject_ids, result);
    }
   else error=UNRECOGNISED_REQUEST;
  break;
  }
  case INTERSECTORS_OF_SUBJECT:
  {
  if (no_of_strings==1)
    {
    Set_String_ subject_ids;
    error = ssmm.intersectors_of_subject(strings[0],  subject_ids);
    if (error==OK) string_set_to_string(subject_ids, result);
    }
   else error=UNRECOGNISED_REQUEST;
  break;
  }
  case NEIGHBOR_NODES:
  {
  if (no_of_strings>=1)
    {
    Set_Node_id_ neighbors;
    Node_id node_id;
    node_id.fill_with(strings, 0, last_string);
    error=ssmm.neighbor_nodes(node_id, neighbors);
    if (error==OK)
      {
      String temp;
      neighbors.next_member_returns_first();
      while (neighbors.can_use_get_next_member())
                {
                string_set_to_string
                          (neighbors.get_next_member(),  temp);
                result += temp;
                result += ",";
                };
     };
    }
   else error=UNRECOGNISED_REQUEST;
  break;
  }
  case URLS_IN_INTERSECTION:
  {
  if (no_of_strings>=1)
    {
```

```
    Set_String_ urls;
    Set_String_ subject_ids;
    subject_ids.fill_with(strings, 0, last_string);
    error = ssmm.urls_in_intersection(subject_ids, urls);
    if (error==OK) string_set_to_string(urls, result);
    }
   else error=UNRECOGNISED_REQUEST;
  break;
  }
  case RESOURCES_IN_NODE:
  {
  if (no_of_strings>=1)
    {
    Set_Resource_ resources;
    Set_String_ node_id;
    node_id.fill_with(strings, 0, last_string);
    error = ssmm.resources_in_node(node_id, resources);
    if (error==OK) resource_set_to_string(resources, result);
    }
   else error=UNRECOGNISED_REQUEST;
  break;
  }
  case TEXT_DUMP:
  {
  ssmm.text_dump(result);
  error=OK;
  break;
  }

  default:
  {
  error=UNRECOGNISED_REQUEST;
  };
  }; //end switch

 switch (error)
  {
  case OK:
            {message=""; break;}
  case UNRECOGNISED_REQUEST:
            {message="request not understood"; break;}
  case ILEGAL_REQUEST:
            {message="ilegal request"; break;}
  case RESOURCE_NOT_FOUND:
            {message="resource not found"; break;}
  case SUBJECT_NOT_FOUND:
            {message="subject not found"; break;}
  case RESOURCE_ALREADY_EXISTS:
            {message="resource already extists"; break;}
  case SUBJECT_ALREADY_EXISTS:
            {message="subject already exists"; break;}
  case NO_SUCH_NODE:
            {message="no such node"; break;}
  default: message="error";
  };
 return error;
 };
```

20 ssci-lang.h

```
#include <String.h>

enum SS_Token {       ERROR,
                      STRING,
                      EOL,
```

```
                    GET_RESOURCE,
                    NEW_RESOURCE,
                    UPDATE_RESOURCE,
                    DELETE_RESOURCE,
                    NEW_SUBJECT,
                    ADD_SUBJECT,
                    DELETE_SUBJECT,
                    REMOVE_SUBJECT,
                    ADD_RESOURCE,
                    REMOVE_RESOURCE,
                    RESOURCES_IN_INTERSECTION,
                    URLS_IN_INTERSECTION,
                    SUBJECTS_OF_RESOURCE,
                    SET_SUBJECTS_OF_RESOURCE,
                    PARENTS_OF_SUBJECT,
                    CHILDREN_OF_SUBJECT,
                    INTERSECTORS_OF_SUBJECT,
                    SUBJECTS_LIST,
                    URLS_LIST,
                    REORGANISE,
                    TEXT_DUMP,
                    NEIGHBOR_NODES,
                    RESOURCES_IN_NODE
            };
```

21 ssci.cxx

```
#include <iostream.h>
#include <gdbm.h>
#include <stdio.h>
#include "ssdbm.h"
#include "ss.h"
#include "ssci-lang.h"

// ************************************************************************ *
// IMPORTANT -- THESE ARE INPUT COMMAND SIZE LIMITS ---
// ************************************************************************ *

const BUFSIZE = 4000; // max size of request in chars
String strings[500];    // this should be in service_request, but the compiler
                        // doesn't like it there!  :-(

// ************************************************************************ **


String string_token;  // used to pass the value of strings back from the lexer
char* flex_input_ptr;     // flex reads from here, and advances this ptr
                          // NB flex must read a '\n'  before it stops!!!


SS_Error service_request (SSDBM& sd, char* request, String& result,
                                                        String& message);

extern int yylex();  // declaration of lexer - yylex() will actualy return
                     // an SS_Token



int main(int argc, char* argv[])
  {
  if (argc!=2) cout <<"arg 1 = database name \n";
  else
    {
    SS_Error error;
    int file_mode;
    char in_buffer[BUFSIZE];
    SSDBM ssdbm(error, argv[1], GDBM_WRCREAT);
```

```
    if (error==OK)
      {
      if (setvbuf(stdin, NULL, _IOLBF, 0) != 0)
                {cerr<<"setvbuf error\n"; exit -1;};
      while (fgets(in_buffer, BUFSIZE, stdin) != NULL)
    {
    String result;
    String message;
    error=service_request(ssdbm, in_buffer, result, message);
    if (result!="")
      {
      if (result.lastchar()!='\n') result+='\n';
      while (result.elem(result.length()-2)==' ')
                                result.del(result.length()-2,1);
      cout << result;
      }
     else cout << '\n';
    cout.flush();
    if (message!="")
      {
      cerr <<argv[0]<<": "<<message<<" : "<<in_buffer<<'\n';
      cerr.flush();
      };
    }
     }
     else
    {cerr<<argv[0]<< " error opening subject space - can only open  subject space by one
program at a time\n";};
   };
  };



SS_Error service_request (SSDBM& sd, char* request_ptr, String& result,
                                                        String& message)
 // request_ptr must point into a char buffer ending in '\n'
  {
  flex_input_ptr=request_ptr;
  SS_Error error;

  // find out what the service request is
  SS_Token service = (SS_Token) yylex();
  // parse the request data, an unknown number of strings
  int last_string=-1;

  SS_Token token = service;
  while ((token!=EOL))
    {
    token = (SS_Token) yylex();
    if (token==STRING) strings[++last_string]=string_token;
    };

  int no_of_strings=last_string+1;

  // process the service request
  switch (service)
    {
    case NEW_RESOURCE:
    {
    if (no_of_strings>=3)
      {
      Resource resource;
      resource.fill_with(strings, 0);
      Set_String_ subject_ids;
      subject_ids.fill_with(strings, 2, last_string);
      error = sd.new_resource (resource, subject_ids);
      }
    else error=UNRECOGNISED_REQUEST;
    break;
    };
    case GET_RESOURCE:
    {
    if (no_of_strings==1)
      {
```

```
  Resource resource;
  String url=strings[0];
  error = sd.get_resource(url, resource);
  if (error==OK)
                {
                result+="\""; result+=resource.url; result+="\"  ";
                result+="\""; result+=resource.name;  result+="\"";
                };
  }
 else error=UNRECOGNISED_REQUEST;
break;
}
case UPDATE_RESOURCE:
{
if (no_of_strings==2)
  {
  Resource resource;
  resource.fill_with(strings, 0);
  error = sd.update_resource (resource);
  }
else error=UNRECOGNISED_REQUEST;
break;
}
case DELETE_RESOURCE:
{
if (no_of_strings==1)
  {
  String url = strings[0];
  error = sd.delete_resource_from_all_subjects(url);
  }
else error=UNRECOGNISED_REQUEST;
break;
}
case NEW_SUBJECT:
{
if (no_of_strings==1)
  error = sd.new_subject(strings[0], "");
else if (no_of_strings==2)
  error = sd.new_subject(strings[0], strings[1]);
else error=UNRECOGNISED_REQUEST;
break;
}
case ADD_SUBJECT:
{
if (no_of_strings>=2)
  {
  Set_String_ parent_ids;
  parent_ids.fill_with(strings, 1, last_string);
  error = sd.add_subject(strings[0], parent_ids);
  }
 else error = UNRECOGNISED_REQUEST;
break;
}
case DELETE_SUBJECT:
{
if (no_of_strings==1)
  error = sd.delete_subject(strings[0]);
 else error = UNRECOGNISED_REQUEST;
break;
}
case REMOVE_SUBJECT:
{
if (no_of_strings==2)
  error = sd.remove_subject(strings[0], strings[1]);
 else error = UNRECOGNISED_REQUEST;
break;
}
case ADD_RESOURCE:
{
if (no_of_strings>=2)
  {
  Set_String_ subject_ids;
  subject_ids.fill_with(strings, 1, last_string);
  error = sd.add_resource_to_subjects(strings[0],  subject_ids);
```

```
  }
else error=UNRECOGNISED_REQUEST;
break;
}
case SET_SUBJECTS_OF_RESOURCE:
{
if (no_of_strings>=2)
  {
  Set_String_ subject_ids;
  subject_ids.fill_with(strings, 1, last_string);
  error = sd.set_subjects_of_resource(strings[0],  subject_ids);
  }
else error=UNRECOGNISED_REQUEST;
break;
}
case REMOVE_RESOURCE:
{
if (no_of_strings>=2)
  {
  Set_String_ subject_ids;
  subject_ids.fill_with(strings, 1, last_string);
  error = sd.remove_resource_from_subjects(strings[0],  subject_ids);
  }
else error=UNRECOGNISED_REQUEST;
break;
}
case URLS_IN_INTERSECTION:
{
if (no_of_strings>=1)
  {
  Set_String_ urls;
  Set_String_ subject_ids;
  subject_ids.fill_with(strings, 0, last_string);
  error = sd.urls_in_intersection(subject_ids, urls);
  if (error==OK) string_set_to_string(urls, result);
  }
 else error=UNRECOGNISED_REQUEST;
break;
}
case RESOURCES_IN_INTERSECTION:
{
if (no_of_strings>=1)
  {
  Set_Resource_ resources;
  Set_String_ subject_ids;
  subject_ids.fill_with(strings, 0, last_string);
  error = sd.resources_in_intersection(subject_ids,  resources);
  if (error==OK) resource_set_to_string(resources, result);
  }
 else error=UNRECOGNISED_REQUEST;
break;
}
case SUBJECTS_OF_RESOURCE:
{
if (no_of_strings==1)
  {
  Set_String_ subject_ids;
  error = sd.subjects_of_resource(strings[0], subject_ids);
  if (error==OK) string_set_to_string(subject_ids, result);
  }
 else error=UNRECOGNISED_REQUEST;
break;
}
case PARENTS_OF_SUBJECT:
{
if (no_of_strings==1)
  {
  Set_String_ subject_ids;
  error = sd.parents_of_subject(strings[0], subject_ids);
  if (error==OK) string_set_to_string(subject_ids, result);
  }
 else error=UNRECOGNISED_REQUEST;
break;
}
```

```
 case CHILDREN_OF_SUBJECT:
 {
 if (no_of_strings==1)
   {
   Set_String_ subject_ids;
   error = sd.children_of_subject(strings[0], subject_ids);
   if (error==OK) string_set_to_string(subject_ids, result);
   }
  else error=UNRECOGNISED_REQUEST;
 break;
 }
 case INTERSECTORS_OF_SUBJECT:
 {
 if (no_of_strings==1)
   {
   Set_String_ subject_ids;
   error = sd.intersectors_of_subject(strings[0], subject_ids);
   if (error==OK) string_set_to_string(subject_ids, result);
   }
  else error=UNRECOGNISED_REQUEST;
 break;
 }
 case SUBJECTS_LIST:
 {
 if (no_of_strings==0)
   {
   Set_String_ subject_ids;
   error = sd.subjects_list(subject_ids);
   if (error==OK) string_set_to_string(subject_ids, result);
   }
  else error=UNRECOGNISED_REQUEST;
 break;
 }
 case URLS_LIST:
 {
 if (no_of_strings==0)
   {
   Set_String_ urls;
   error = sd.urls_list(urls);
   if (error==OK) string_set_to_string(urls, result);
   }
  else error=UNRECOGNISED_REQUEST;
 break;
 }
 case REORGANISE:
 {
 sd.reorganise();
 error=OK;
 break;
 }
 case TEXT_DUMP:
 {
 sd.text_dump(result);
 error=OK;
 break;
 }

 default:
 {
 error=UNRECOGNISED_REQUEST;
 };
 }; //end switch

switch (error)
 {
 case OK:
        {message=""; break;}
 case UNRECOGNISED_REQUEST:
        {message="request not understood"; break;}
 case ILEGAL_REQUEST:
        {message="ilegal request"; break;}
 case RESOURCE_NOT_FOUND:
        {message="resource not found"; break;}
 case SUBJECT_NOT_FOUND:
```

```
                {message="subject not found"; break;}
    case RESOURCE_ALREADY_EXISTS:
                {message="resource already extists"; break;}
    case SUBJECT_ALREADY_EXISTS:
                {message="subject already exists"; break;}
    default: message="error";
    };
  return error;
  };
```

22 ssci.l

```
%{
#include "ssci-lang.h"
#include "functions.h"
#include <stdlib.h>


extern char* flex_input_ptr;

int new_yyinput (char* buf, int max_size); // NB assumes input ends with null  as
                                                        //  first char
int new_yyinput (char* buf, int max_size) // NB assumes input ends with null  as
                                                        //  first char
  {
  int n=0;
  while ( ((*flex_input_ptr)!='\0') && (n<=max_size) )
    {*(buf+(n++))=*(flex_input_ptr++);};
  return n;
  };

int yywrap()
    {
    return 1;
    }

#undef YY_INPUT
#define YY_INPUT(buf,result,max_size) (result=new_yyinput(buf, max_size))

extern String string_token;
    // used to pass the value of strings back from the lexer

%}

%%

\"[^\"\n]*[\"] {string_token=yytext; unquote(string_token); return STRING;}

\n                                  {return EOL;}

"get"[ \t]*"resource"|"gr"                  {return GET_RESOURCE;}
"new"[ \t]*"resource"|"nr"                  {return NEW_RESOURCE;}
"delete"[ \t]*"resource"|"dr"               {return DELETE_RESOURCE;}
"update"[ \t]*"resource"|"ur"               {return UPDATE_RESOURCE;}
"add"[ \t]*"resource"|"ar"                  {return ADD_RESOURCE;}
"remove"[ \t]*"resource"|"rr"               {return REMOVE_RESOURCE;}
"resources"[ \t]*"in""intersection"|"ri"    {return  RESOURCES_IN_INTERSECTION;}
"urls"[ \t]*"in""intersection"|"ui"         {return URLS_IN_INTERSECTION;}
"subjects"[ \t]*"of"[ \t]*"resource"|"sr"      {return SUBJECTS_OF_RESOURCE;}
"set"[ \t]*"subjects"[ \t]*"of"[ \t]*"resource"|"ss"    {
                                                return SET_SUBJECTS_OF_RESOURCE;}
"new"[ \t]*"subject"|"ns"                   {return NEW_SUBJECT;}
"delete"[ \t]*"subject"|"ds"                {return DELETE_SUBJECT;}
"add"[ \t]*"subject"|"as"                   {return ADD_SUBJECT;}
"remove"[ \t]*"subject"|"rs"                {return REMOVE_SUBJECT;}
"parents"[ \t]*"of"[ \t]*"subject"|"ps"        {return PARENTS_OF_SUBJECT;}
```

```
"children"[ \t]*"of"[ \t]*"subject"|"cs"       {return CHILDREN_OF_SUBJECT;}
"intersectors"[ \t]*"of"[ \t]*"subject"|"is"  {return  INTERSECTORS_OF_SUBJECT;}
"subjects"[ \t]*"list"|"sl"                   {return SUBJECTS_LIST;}
"urls"[ \t]*"list"|"ul"                      {return URLS_LIST;}
"reorganise"|"ro"                             {return REORGANISE;}
"text"[ \t]*"dump"|"td"                          {return TEXT_DUMP;}

"neighbor"[ \t]*"nodes"|"nn"                 {return NEIGHBOR_NODES;}
"resources"[ \t]*"in"[ \t]*"node"|"rn"         {return RESOURCES_IN_NODE;}

[ \t] ;  /* ignore whitespace */

. return ERROR;
%%
```

23 ssdbm-tcl.h

```
#include "ssdbm.h"
#include "ss.h"


SSDBM* ssdbm_ptr;  // this must be set by any code using these fns, it is
                   // the SSDBM that they use! wow!

void register_new_tcl_commands (Tcl_Interp* interp);


int ssdbm_resources_in_intersection (ClientData clientData, Tcl_Interp*  interp,
                                                              int  argc, char* argv[]);
int ssdbm_subjects_list
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[]);
int ssdbm_subjects_of_resource
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[]);
int ssdbm_new_resource
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[]);
int ssdbm_exists_resource
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[]);
int ssdbm_destructor
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[]);
int ssdbm_delete_resource
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[]);
int ssdbm_subjects_list
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[]);
int ssdbm_resources_list
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[]);



void fill_string_set_with_tcl_list (Set_String_& intersection, char* list);

void Resource_to_tcl_list_element (Resource resource, String string);
void String_Set_to_tcl_list
                (Tcl_Interp* interp, Set_String_& set, char*  listname);
void subject_id_set_to_tcl_list
                (Tcl_Interp* interp, Set_String_& set, char*  listname);
```

24 ssdbm-tcl.cxx

```
#include "gdbm.h"
#include "ssdbm.h"
#include "ss.h"
#include <tcl.h>
```

```
#include "ssdbm-tcl.h"


int ssdbm_resources_in_intersection
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[])
  {
  // arg[1] is the urls list variable name,
  // arg[2] is the resource list var name,
  // arg[3] is the intersection
  Set_Resource_ resources;
  Set_String_ intersection;
  fill_string_set_with_tcl_list (intersection, argv[3]);
  (*ssdbm_ptr).resources_in_intersection(intersection, resources);
  Resource tmp;
  Tcl_SetVar(interp, argv[1], "", 0);  //empty the list
  Tcl_SetVar(interp, argv[2], "", 0);  //empty the list
  while (resources.can_use_get_next_member())
    {
    resources.get_next_member(tmp);
    Tcl_SetVar(interp, argv[1], "{", TCL_APPEND_VALUE);
    Tcl_SetVar(interp, argv[1], tmp.url, TCL_APPEND_VALUE);
    Tcl_SetVar(interp, argv[1], "}", TCL_APPEND_VALUE);
    Tcl_SetVar(interp, argv[1], " ", TCL_APPEND_VALUE);
    Tcl_SetVar(interp, argv[2], "{", TCL_APPEND_VALUE);
    Tcl_SetVar(interp, argv[2], tmp.name, TCL_APPEND_VALUE);
    Tcl_SetVar(interp, argv[2], "}", TCL_APPEND_VALUE);
    Tcl_SetVar(interp, argv[2], " ", TCL_APPEND_VALUE);
    };
  return TCL_OK;
  };

int ssdbm_subjects_list
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[])
  {
  // argv[1] is subjects_list (to be filled)
  Set_String_ subjects;
  (*ssdbm_ptr).subjects_list(subjects);
  String temp;
  subject_id_set_to_tcl_list(interp, subjects, argv[1]);
  return TCL_OK;
  };

int ssdbm_subjects_of_resource
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[])
  // argv[1] is url
  // argv[2] is name of subjects_list
  {
  Set_String_ subjects;
  (*ssdbm_ptr).subjects_of_resource(argv[1], subjects);
  subject_id_set_to_tcl_list (interp, subjects, argv[2]);
  return TCL_OK;
  };

int ssdbm_set_subjects_of_resource
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[])
  {
  // argv[1] is url
  // argv[2] is name of subjects list
  Set_String_ subjects;
  fill_string_set_with_tcl_list (subjects, argv[2]);
  (*ssdbm_ptr).set_subjects_of_resource(argv[1], subjects);
  return TCL_OK;
  };


int ssdbm_new_resource
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[])
  // argv[1] is url
  // argv[2] is name
  // argv[3] is a list of subject ids
  {
  Resource resource;
  resource.url=argv[1];
```

```
  resource.name=argv[2];
  Set_String_ subject_ids;
  fill_string_set_with_tcl_list (subject_ids, argv[3]);
  (*ssdbm_ptr).new_resource (resource, subject_ids);
  return TCL_OK;
  };

int ssdbm_update_resource
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[])
  {
  // argv[1] is url
  // argv[2] is resource name
  Resource resource;
  resource.url=argv[1];
  resource.name=argv[2];
  (*ssdbm_ptr).update_resource (resource);
  return TCL_OK;
  };

int ssdbm_exists_resource
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[])
  // argv[1] is a url
  // argv[2] is name of status variable, fill with "yes" or "no"
  {
  if (((*ssdbm_ptr).exists_resource(argv[1]))==RESOURCE_ALREADY_EXISTS)
     Tcl_SetVar(interp, argv[2], "yes", 0);
   else Tcl_SetVar(interp, argv[2], "no", 0);
  return TCL_OK;
  };

int ssdbm_delete_resource
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[])
  // argv[1] is a url
  {
  (*ssdbm_ptr).delete_resource_from_all_subjects(argv[1]);
  return TCL_OK;
  };

int ssdbm_destructor
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[])
  // no args
  {
  (*ssdbm_ptr).~SSDBM();
  return TCL_OK;
  };

int ssdbm_resources_list
    (ClientData clientData, Tcl_Interp* interp, int argc, char*  argv[])
  // argv[1] is the name of the urls list to fill
  // argv[2] is the name of the resources (names) list to fill
  {
  Set_String_ urls;
  Set_String_ resource_names;
  (*ssdbm_ptr).urls_list(urls);
  String_Set_to_tcl_list (interp, urls, argv[1]);
  Resource resource;
  urls.next_member_returns_first();
  while (urls.can_use_get_next_member())
    {
    (*ssdbm_ptr).get_resource (urls.get_next_member(), resource);
    resource_names.add_member(resource.name);
    };
  String_Set_to_tcl_list (interp, resource_names, argv[2]);
  return TCL_OK;
  };


void register_new_tcl_commands (Tcl_Interp* interp)
  {
  Tcl_CreateCommand (interp, "ssdbm_resources_in_intersection",
                             ssdbm_resources_in_intersection,  (ClientData *) NULL,
                             (Tcl_CmdDeleteProc *) NULL);
  Tcl_CreateCommand (interp, "ssdbm_subjects_list",
                             ssdbm_subjects_list, (ClientData  *) NULL,
```

```
                        (Tcl_CmdDeleteProc *) NULL);
  Tcl_CreateCommand (interp, "ssdbm_update_resource",
                        ssdbm_update_resource,  (ClientData *) NULL,
                        (Tcl_CmdDeleteProc *) NULL);
  Tcl_CreateCommand (interp, "ssdbm_set_subjects_of_resource",
                        ssdbm_set_subjects_of_resource,  (ClientData *) NULL,
                        (Tcl_CmdDeleteProc *) NULL);
  Tcl_CreateCommand (interp, "ssdbm_subjects_of_resource",
                        ssdbm_subjects_of_resource,  (ClientData *) NULL,
                        (Tcl_CmdDeleteProc *) NULL);
  Tcl_CreateCommand (interp, "ssdbm_new_resource",
                        ssdbm_new_resource, (ClientData  *) NULL,
                        (Tcl_CmdDeleteProc *) NULL);
  Tcl_CreateCommand (interp, "ssdbm_exists_resource",
                        ssdbm_exists_resource,  (ClientData *) NULL,
                        (Tcl_CmdDeleteProc *) NULL);
  Tcl_CreateCommand (interp, "ssdbm_destructor",
                        ssdbm_destructor, (ClientData *)  NULL,
                        (Tcl_CmdDeleteProc *) NULL);
  Tcl_CreateCommand (interp, "ssdbm_delete_resource",
                        ssdbm_delete_resource,  (ClientData *) NULL,
                        (Tcl_CmdDeleteProc *) NULL);
  Tcl_CreateCommand (interp, "ssdbm_resources_list",
                        ssdbm_resources_list, (ClientData  *) NULL,
                        (Tcl_CmdDeleteProc *) NULL);
  };


void fill_string_set_with_tcl_list (Set_String_& intersection, char* list)
  {
  char* ptr = list;
  while ((*ptr)!='\0')
    {
    //skip whitespace
    String tmp;
    while ((*ptr=='\t')||(*ptr=='\x020')) ptr++; // \x020 is SPACE  char
    while ((*ptr!='\0')&&(*ptr!='\t')&&(*ptr!='\x020'))
                {
                tmp += *ptr;
                ptr++;
                };
    if (tmp.length()>0)
                intersection.add_member(tmp);
    };
  };



void Resource_to_tcl_list_element (Resource resource, String string)
  {
  string = "{";
  string += resource.url;
  string += " ";
  string += resource.name;
  string += "}";
  };

void String_Set_to_tcl_list
                (Tcl_Interp* interp, Set_String_& set, char*  listname)
  {
  Tcl_SetVar(interp, listname, "" , 0); //empty list
  String temp;
  while (set.can_use_get_next_member())
    {
    temp = set.get_next_member();
    Tcl_SetVar(interp, listname, "{", TCL_APPEND_VALUE);
    Tcl_SetVar(interp, listname, temp, TCL_APPEND_VALUE);
    Tcl_SetVar(interp, listname, "}", TCL_APPEND_VALUE);
    Tcl_SetVar(interp, listname, " ", TCL_APPEND_VALUE);
    };
  };

void subject_id_set_to_tcl_list
                (Tcl_Interp* interp, Set_String_& set, char*  listname)
```

```
  {
  Tcl_SetVar(interp, listname, "" , 0); //empty list
  String temp;
  while (set.can_use_get_next_member())
   {
   temp = set.get_next_member();
   Tcl_SetVar(interp, listname, temp, TCL_APPEND_VALUE);
   Tcl_SetVar(interp, listname, " ", TCL_APPEND_VALUE);
   };
  };
```

25 ssdbm.h

```
#ifndef _ssdbm_h
#define _ssdbm_h 1


#include <unistd.h>
#include <iostream.h>
#include <stdlib.h>
#include <gdbm.h>   /* gnu database managmer */
#include "ss.h"
#include "functions.h"
#include <errno.h>
#include <BitSet.h>


const String subject_file_extention = ".subject";
const String record_file_extention = ".record";
const int gdbm_file_block_size = 1024;
const int gdbm_file_mode = 00664;



enum SS_Error { OK,
                FATAL_SERVER_ERROR,
                BAD_REPLY_FILE,
                UNRECOGNISED_REQUEST,
                DATABASE_NOT_FOUND,
                RESOURCE_NOT_FOUND,
                SUBJECT_NOT_FOUND,
                ILEGAL_REQUEST,
                RESOURCE_ALREADY_EXISTS,
                SUBJECT_ALREADY_EXISTS,
                NO_SUCH_NODE
        };

enum SS_Update_Mode { ACTIVE, PASSIVE };



class SSDBM
  {
  public:
    // status set to 0 for OK, else -1
    SSDBM (SS_Error& status, const String& name,
                        int flags = GDBM_WRCREAT);
    //NB see gbdm.h for values of flag
    ~SSDBM ();
    void get_errors(int& gdbm_errors_, int& file_errors_)
                {gdbm_errors_=gdbm_errno; file_errors_=errno;};

    // all these functions return negative error values, or 0 = OK
    // NB /*psconst*/ means psuedo const, set members are  unchanged, but
    /* Set object is changed (as its iterator is used), so it can't  be
       const */

    SS_Error new_resource (const Resource& resource,
```

```
        /*psconst*/ Set_String_& subject_ids);
    SS_Error update_resource (const Resource& resource);
    SS_Error delete_resource_from_all_subjects (const String&  url);
    SS_Error get_resource (const String& url, Resource& resource);

    SS_Error new_subject (const String& subject_id,
                          const String& parent_id);
    SS_Error add_subject (const String& subject_id,
                          const String& parent_id);
    SS_Error add_subject (const String& subject_id,
                          /*psconst*/ Set_String_&  parent_ids);
    SS_Error delete_subject (const String& subject_id);
    SS_Error remove_subject
                (const String& subject_id,const String&  child_id);

    SS_Error add_resource_to_subjects (const String& url,
                           Set_String_& subject_ids);
    SS_Error remove_resource_from_subjects (const String& url,
                                            / *psconst*/ Set_String_& subject_ids);
    SS_Error urls_in_intersection
                          (/*psconst*/ Set_String_&  subject_ids,
                           Set_String_& urls);
    SS_Error subjects_of_resource
                          (const String& url, Set_String_&  subject_ids);
    SS_Error set_subjects_of_resource
                          (const String& url, Set_String_&  subject_ids);
    SS_Error resources_in_intersection
                          (/*psconst*/ Set_String_&  subject_ids,
                           Set_Resource_& resources);
    SS_Error parents_of_subject (const String& subject_id,
                                 Set_String_&  parent_ids);
    SS_Error children_of_subject (const String& subject_id,
                                  Set_String_&  child_ids);
    SS_Error intersectors_of_subject (const String& subject_id,
                                      Set_String_&  intersector_ids);
    SS_Error urls_list (Set_String_& urls);
    SS_Error subjects_list (Set_String_& subjects);
    SS_Error exists_resource (const String& url);

    void text_dump (String& dump);
    void reorganise();

  protected:
    GDBM_FILE records_file;
    GDBM_FILE subjects_file;
    datum data_datum;
    datum key_datum;
    SS_Error load_record (const String& url, Record& record);
    SS_Error load_subject (const String& subject_id, Subject&  subject);
    SS_Error save_record (/*psconst*/ Record& record, int flag);
    SS_Error save_subject (/*psconst*/ Subject& subject, int  flag);
    SS_Error update_records_after_additions_to_subject
                (const Subject& original_subject, const  Subject& new_subject,
                 const SS_Update_Mode mode);
    SS_Error update_records_after_deletions_from_subject
                (const Subject& original_subject, const  Subject& new_subject,
                                                  const SS_Update_Mode mode);
    SS_Error update_subjects_after_additions_to_record
                (const Record& original_record, const Record&  new_record,
                                                const SS_Update_Mode mode);
    SS_Error update_subjects_after_deletions_from_record
                (const Record& original_record, const Record&  new_record,
                                                      SS_Update_Mode mode);

    SS_Error tidy();
  };


class SS_Map_Manager : public SSDBM
  {
  public:
    SS_Map_Manager::SS_Map_Manager (SS_Error& status, const  String& name,
                                                  int flags=GDBM_WRCREAT)
                          : SSDBM(status, name, flags) {};
```

```
  SS_Error resources_in_node
                (Node_id node_id, Set_Resource_& resources);
  SS_Error neighbor_nodes
                (Node_id node_id, Set_Node_id_&  nearby_nodes);
  SS_Error possible_unit_distance_nodes
                (Node_id node_id, Set_Node_id_&  unit_distance_nodes);

  SS_Error intersectors_of_node
                (Node_id node_id, Set_String_& intersectors);
  SS_Error subjects_close_to_node
                (Node_id node_id, Set_String_& close_subjects);

  void test()
                {
                Set_String_ a;
                a.add_member (" pig ");
                a.add_member (" cow ");
                a.add_member (" hen ");
                a.add_member (" horse ");
                a.add_member (" bull ");
                a.add_member (" sheep ");
                a.add_member (" dog ");
                a.add_member (" goose ");
                Node_id b;
                b.add_member (" hen ");
                b.add_member (" horse ");
                b.add_member (" bull ");

                Set_Node_id_ c;
                generate_node_id_combinations
                        (b,a,0,c);
                cout <<"\nresults 0: \n";
                c.screen_dump();

                Set_Node_id_ d;
                generate_node_id_combinations
                        (b,a,1,d);
                cout <<"\nresults 1: \n";
                d.screen_dump();

                Set_Node_id_ e;
                generate_node_id_combinations
                        (b,a,2,e);
                cout <<"\nresults 2: \n";
                e.screen_dump();

                Set_Node_id_ f;
                generate_node_id_combinations
                        (b,a,3,f);
                cout <<"\nresults 3: \n";
                f.screen_dump();

                Set_Node_id_ g;
                generate_node_id_combinations
                        (b,a,4,g);
                cout <<"\nresults 4: \n";
                g.screen_dump();



                };

protected:
  void generate_node_id_combinations
                (Node_id& node_id, Set_String_& subject_ids, int  distance,
                                                Set_Node_id_& generated_node_ids);
  void generate_bit_combinations
                        (unsigned u, unsigned d, unsigned  s,
                                                        BitSet* result);
};
```

```
#endif
```

26 ssdbm.cxx

```
#include "ssdbm.h"
#include <BitSet.h>
#include "functions.h"


SS_Error SSDBM::add_resource_to_subjects (const String& url,
                                          Set_String_& subject_ids)
  {
  SS_Error error;
  Record original_rec, new_rec;
  error = load_record(url, original_rec);
  if (error==OK)
    {
    new_rec=original_rec;
    new_rec.subject_ids+=subject_ids;
    error = update_subjects_after_additions_to_record
                                                  (original_rec, new_rec, PASSIVE);
    if (error==OK) save_record (new_rec, GDBM_REPLACE);
    };
  return error;
  };

SS_Error SSDBM::remove_resource_from_subjects (const String& url,
                                                          Set_String_& subject_ids)
  {
  SS_Error error;
  Record original_rec, new_rec;
  error = load_record(url, original_rec);
  if (error==OK)
    {
    new_rec=original_rec;
    new_rec.subject_ids-=subject_ids;
    if (new_rec.subject_ids.no_of_members()!=0)
      {
      error = update_subjects_after_deletions_from_record
                                                  (original_rec, new_rec, PASSIVE);
      if (error==OK) save_record (new_rec, GDBM_REPLACE);
      }
     else error=ILEGAL_REQUEST;
    };
  return error;
  };



SS_Error SSDBM::new_resource (const Resource& resource,
                                                  Set_String_& subject_ids)
  {
//cout << "in SSDBM::new_resource\n";
  SS_Error error;
  Record empty_rec;
  Record new_rec (resource,subject_ids);
  error = update_subjects_after_additions_to_record
                                                          (empty_rec, new_rec, PASSIVE);
  if (error==OK) error=save_record (new_rec, GDBM_INSERT);
  return error;
  };


SS_Error SSDBM::update_resource (const Resource& resource)
  {
  Record rec;
```

```
  SS_Error error = load_record(resource.url, rec);
  if (error==OK)
    {
    rec.resource=resource;
    error = save_record(rec, GDBM_REPLACE);
    };
  return error;
  };


SS_Error SSDBM::new_subject (const String& subject_id,
                                                const String& parent_id)
  {
  SS_Error error;
  Subject sub_parent;
  Subject sub;
  sub.subject_id=subject_id;

  if (parent_id.length()!=0)
     {
     error = load_subject(parent_id, sub_parent);
     sub.parent_subjects.add_member(parent_id);
     sub_parent.child_subjects.add_member(subject_id);
     }
   else error = OK;
  if (error == OK)
    {
    error = save_subject(sub, GDBM_INSERT);
    if ((error==OK)&&(parent_id.length()!=0))
    {
    error = save_subject(sub_parent, GDBM_REPLACE);
    };
    };
  return error;
  };

SS_Error SSDBM::add_subject
                          (const String& subject_id, const  String& parent_id)
  {
  SS_Error error;
  Subject sub_parent;
  Subject sub;
  error = load_subject(subject_id, sub);
  if (error==OK)
    {
    error = load_subject(parent_id, sub_parent);
    if (error==OK)
    {
    sub.parent_subjects.add_member(parent_id);
    sub_parent.child_subjects.add_member(subject_id);
    error = save_subject(sub, GDBM_REPLACE);
    if (error==OK) error=save_subject(sub_parent, GDBM_REPLACE);
    };
    };
  return error;
  };

SS_Error SSDBM::add_subject
                (const String& subject_id,  /*psconst*/  Set_String_& parent_ids)
  {
  SS_Error error, temp_error;
  error=OK;
  parent_ids.next_member_returns_first();
  while (parent_ids.can_use_get_next_member())
    {
    temp_error=add_subject(subject_id,  parent_ids.get_next_member());
    if (temp_error!=OK) error=temp_error;
    };
  return error;
  };

SS_Error SSDBM::remove_subject
                          (const String& subject_id,const  String& child_id)
  {
```

```
  SS_Error error;
  Subject sub_child;
  Subject sub;
  error = load_subject(subject_id, sub);
  if (error==OK)
    {
    error = load_subject(child_id, sub_child);
    if (error==OK)
    {
    sub.parent_subjects.delete_member(child_id);
    sub_child.parent_subjects.delete_member(subject_id);
    error = save_subject(sub, GDBM_REPLACE);
    if (error==OK) error=save_subject(sub_child, GDBM_REPLACE);
    };
    };
  return error;
  };


SS_Error SSDBM::delete_resource_from_all_subjects
                                    (const String& url)
  {
  Record rec_original, rec_new;
  SS_Error error = load_record(url, rec_original);
  if (error==OK)
    {
    rec_new.resource.url=rec_original.resource.url;
    //ie rec_new has the same name as rec_original, but it's in no  subjects
    error=update_subjects_after_deletions_from_record
                                              (rec_original, rec_new, ACTIVE);
    if (error==OK)
      {
      key_datum.dptr=url;
      key_datum.dsize=url.length()+1;
      gdbm_delete(records_file, key_datum);
      };
    };
  return error;
  };


SS_Error SSDBM::subjects_of_resource
                          (const String& url, Set_String_&  subject_ids)
  {
  SS_Error error;
  Record record;
  error=load_record(url, record);
  if (error==OK) subject_ids=record.subject_ids;
  return error;
  };


SS_Error SSDBM::set_subjects_of_resource
                          (const String& url, Set_String_&  subject_ids)
 //this function sets the subjects of the resource, reguardless of the  resources
 //previous subjects, all updates are ACTIVE.
  {
  SS_Error error;
  Record original_rec, intermediate_rec, new_rec;
  error = load_record(url, original_rec);
  if (error==OK)
    {
    intermediate_rec=original_rec;
    intermediate_rec.subject_ids+=subject_ids;
    error = update_subjects_after_additions_to_record
                                      (original_rec,  intermediate_rec, ACTIVE);
    if (error==OK)
      {
      new_rec=original_rec;
      new_rec.subject_ids=subject_ids;
      error = update_subjects_after_deletions_from_record
                                      (intermediate_rec,  new_rec, ACTIVE);
      if (error==OK) save_record (new_rec, GDBM_REPLACE);
      };
```

```
  };
return error;
};

SS_Error SSDBM::urls_in_intersection
                        (/*psconst*/ Set_String_&  subject_ids,
                         Set_String_& urls)
{
SS_Error error;
Subject temp_sub;
subject_ids.next_member_returns_first();
for (int n=1; n<=subject_ids.no_of_members(); n++)
  {
  error = load_subject(subject_ids.get_next_member(), temp_sub);
  if (error!=OK) break;
  if (n==1) urls=temp_sub.urls;
    else    urls*=temp_sub.urls;
  };
return error;
};

SS_Error SSDBM::resources_in_intersection
        (/*psconst*/ Set_String_& subject_ids, Set_Resource_&  resources)
{
SS_Error error;
Set_String_ urls;
Record rec;
error = urls_in_intersection(subject_ids, urls);
if (error==OK)
  {
  urls.next_member_returns_first();
  while (urls.can_use_get_next_member())
  {
  error = load_record(urls.get_next_member(), rec);
  if (error!=OK) break;
  resources.add_member(rec.resource);
  };
  };
return error;
};

SS_Error SSDBM::parents_of_subject
                        (const String& subject_id,  Set_String_& parent_ids)
{
SS_Error error;
Subject sub;
error=load_subject(subject_id,sub);
if (error==OK) parent_ids=sub.parent_subjects;
return error;
};

SS_Error SSDBM::children_of_subject
                        (const String& subject_id,  Set_String_& child_ids)
{
SS_Error error;
Subject sub;
error=load_subject(subject_id,sub);
if (error==OK) child_ids=sub.child_subjects;
return error;
};


SS_Error SSDBM::intersectors_of_subject
                        (const String& subject_id,  Set_String_& intersector_ids)
{
SS_Error error;
Subject sub;
error=load_subject(subject_id,sub);
if (error==OK)
  {
  // now must get all the subjects of all the records of sub
  Record temp_rec;
  sub.urls.next_member_returns_first();
  while (sub.urls.can_use_get_next_member())
```

```
      {
      error=load_record(sub.urls.get_next_member(),temp_rec);
      if (error==OK)
        intersector_ids+=temp_rec.subject_ids;
       else break;
      };
    };
  return error;
  };

SS_Error SSDBM::delete_subject (const String& subject_id)
 //NB this operation is ileagal if it leaves subjectless records
  {
  SS_Error error;
  Subject original;
  Subject empty_original;
  error = load_subject(subject_id, original);
  if (error==OK)
    {
    empty_original.subject_id=subject_id;
    error = update_records_after_deletions_from_subject
                                    (original,  empty_original, PASSIVE);
    if (error==OK)
    {
    // temp needed here, otherwise simplified() would return a  tempary ob.
    String temp = simplified(subject_id);
    key_datum.dptr=temp;
    key_datum.dsize=temp.length()+1;
    gdbm_delete(subjects_file, key_datum);

    //now make all children excommunicate this parent
    Subject child;
    original.child_subjects.next_member_returns_first();
    while (original.child_subjects.can_use_get_next_member())
      {
      error = load_subject
                          (original.child_subjects.get_next_member(), child);
      if (error!=OK) break;
      child.parent_subjects.delete_member(subject_id);
      error = save_subject (child, GDBM_REPLACE);
      if (error!=OK) break;
      };
    };
    };
  return error;
  };


SS_Error SSDBM::update_subjects_after_additions_to_record
                  (const Record& original_record, const Record&  new_record,
                   const SS_Update_Mode mode)
 // must make sure that i) new subjects exist, if mode ACTIVE make new
                                                                    //  subjects
 //                ii) new subjects in record made to reference record
  {
  Set_String_ new_subject_ids = new_record.subject_ids;
  new_subject_ids-=original_record.subject_ids;
  SS_Error error=OK;
  Subject* subjects=new Subject[new_subject_ids.no_of_members()];
  new_subject_ids.next_member_returns_first();
  String temp_subject_id;
  int n;
  for (n=0; n<new_subject_ids.no_of_members(); n++)
    {
    temp_subject_id=new_subject_ids.get_next_member();
    error = load_subject(temp_subject_id, subjects[n]);
    if (error==OK)
      subjects[n].urls.add_member(new_record.resource.url);
     else
       if ((mode==ACTIVE)&&(error==SUBJECT_NOT_FOUND))
                {
                error = new_subject (temp_subject_id, "");
                subjects[n].subject_id=temp_subject_id;
                 subjects[n].urls.add_member(new_record.resource.url);
```

```
                };
    if (error!=OK) break;
    };
  if (error==OK)
    for (int n=0; n<new_subject_ids.no_of_members(); n++)
                save_subject(subjects[n], GDBM_REPLACE);
  delete[] subjects;
  return error;
  };

SS_Error SSDBM::update_subjects_after_deletions_from_record
                (const Record& original_record, const Record&  new_record,
                                                        const SS_Update_Mode mode)
 // must make sure that references are removed from the subjects
 // if mode ACTIVE, chuck away subjects that are empty & unrelated
  {
  Set_String_ stale_references = original_record.subject_ids;
        stale_references -= new_record.subject_ids;
  Subject sub;
  SS_Error error=OK;
  stale_references.next_member_returns_first();
  while (stale_references.can_use_get_next_member())
    {
    error = load_subject(stale_references.get_next_member(), sub);
    if (error == OK)
                {
                 sub.urls.delete_member(original_record.resource.url);
                error = save_subject(sub, GDBM_REPLACE);
                }
       else break;
    };
  if ((error==OK)&&(mode==ACTIVE))
    {
    stale_references.next_member_returns_first();
    while (stale_references.can_use_get_next_member())
      {
      error = load_subject(stale_references.get_next_member(),  sub);
      if (error == OK)
                if (    (sub.urls.no_of_members()==0) &&
                                (sub.parent_subjects.no_of_members()==0) &&
                                (sub.child_subjects.no_of_members()==0) )
                  error = delete_subject (sub.subject_id);
      if (error!=OK) break;
      }
    }
  return error;
  };


SS_Error SSDBM::update_records_after_additions_to_subject
                (const Subject& original_subject, const  Subject& new_subject,
                                                        const SS_Update_Mode mode)
 //must add references to the new subject, in all the resource records that
 //have been added to new_subject
  {
 //Set_String_ new_references = ((new_subject.urls)(original_subject.urls));
  Set_String_ new_references=new_subject.urls;
                new_references-=original_subject.urls;

  // now add the new references
  // NB we don't assume that all the new references exist
  SS_Error error=OK;
  //Record records[new_references.no_of_members()];
  Record* records= new  Record[new_references.no_of_members()*sizeof(Record)];
  int n;
  new_references.next_member_returns_first();
  for (n=0; n<new_references.no_of_members(); n++)
    {
    error=load_record (new_references.get_next_member(),  records[n]);
    if (error==OK)
       records[n].subject_ids.add_member(original_subject.subject_id);
     else break;
    };
  if (error==OK)
```

```
    for (n=0; n<new_references.no_of_members(); n++)
      save_record (records[n], GDBM_REPLACE);
  delete[] records;
  return error;
  };

SS_Error SSDBM::update_records_after_deletions_from_subject
                (const Subject& original_subject, const  Subject& new_subject,
                                                        SS_Update_Mode mode)
 //must remove references to original_subject from all records that have been
 //removed from original_subject
  {
  SS_Error error = OK;
  int deletion_makes_a_subjectless_record=0;
 //Set_String_ stale_references = (original_subject.urls)(new_subject.urls);
  Set_String_ stale_references=original_subject.urls;
              stale_references-=new_subject.urls;

  // get rid of stale references
  if (mode=PASSIVE)  // ie must not end up with records that are in
    {                                       // no subjects
    int n;
    Record records[stale_references.no_of_members()];
    stale_references.next_member_returns_first();
    for (n=0; n<stale_references.no_of_members(); n++)
    {
    error=load_record (stale_references.get_next_member(),  records[n]);
    if (error!=OK) break;
     records[n].subject_ids.delete_member(original_subject.subject_id);
    if (records[n].subject_ids.no_of_members()==0)
              deletion_makes_a_subjectless_record=1;
    if (deletion_makes_a_subjectless_record==1) break;
    };
    if ((deletion_makes_a_subjectless_record==1)&&(error==OK))
     error=ILEGAL_REQUEST;
     else
    {
    for (n=0; n<stale_references.no_of_members(); n++)
              {
              error=save_record (records[n], GDBM_REPLACE);
              if (error!=OK) break;
              };
    };
    }
   else   // mode == ACTIVE, ie can delete records that end up in no
    {                                       // subjects
    Record rec;
    stale_references.next_member_returns_first();
    for (int n=0; n<stale_references.no_of_members(); n++)
    {
    load_record (stale_references.get_next_member(), rec);
    rec.subject_ids.delete_member(original_subject.subject_id);
    if (rec.subject_ids.no_of_members()==0)
              {
              key_datum.dptr=rec.resource.url;
              key_datum.dsize=rec.resource.url.length()+1;
              gdbm_delete(records_file, key_datum);
              }
      else  save_record (rec, GDBM_REPLACE);
    };
     };

  return error;
  };


SS_Error SSDBM::save_subject (Subject& subject, int flag)
  {
  SSByteStore data (100,80);
  SSByteStore key (20,4);

  key.store(simplified(subject.subject_id));
  key_datum.dptr = (char*)key.pointer();
  key_datum.dsize = key.used_size();
```

```
  data.store(subject);

  data_datum.dptr = (char*)data.pointer();
  data_datum.dsize = data.used_size();

  int ret =  (gdbm_store(subjects_file, key_datum, data_datum, flag));
  if (ret==0) return OK;
  if (ret==1) return SUBJECT_ALREADY_EXISTS;
  else return FATAL_SERVER_ERROR;

}; //end ::save_subject


SS_Error SSDBM::load_subject (const String& subject_id,
                                        Subject&  subject)
  {
  SS_Error ret=OK;
  SSByteStore data (0,80);
  SSByteStore key (20,4);

  key.store(simplified(subject_id));
  key_datum.dptr = (char*) key.pointer();
  key_datum.dsize = key.used_size();

  data_datum = gdbm_fetch (subjects_file, key_datum);
  if (data_datum.dptr==NULL)
    {
    ret=SUBJECT_NOT_FOUND;
    }
    else
    {
    data.take_over(data_datum.dptr, data_datum.dsize);
    data.get(subject);
    };
  return ret;

}; //end ::load_subject



SS_Error  SSDBM::exists_resource (const String& url)
  {
  SS_Error ret;
  SSByteStore key (20,4);

  key.store(url);
  key_datum.dptr = (char*) key.pointer();
  key_datum.dsize = key.used_size();

  data_datum = gdbm_fetch (records_file, key_datum);
  if (data_datum.dptr==NULL)
    ret=RESOURCE_NOT_FOUND;
    else
    ret=RESOURCE_ALREADY_EXISTS;
  return ret;
  };

SS_Error SSDBM::save_record (Record& record, int flag)
  {
  SSByteStore data (100,80);
  SSByteStore key (20,4);

  key.store(record.resource.url);
  key_datum.dptr=(char*)key.pointer();
  key_datum.dsize=key.used_size();

  data.store(record);

  data_datum.dptr=(char*)data.pointer();
  data_datum.dsize=data.used_size();

  int ret = (gdbm_store (records_file, key_datum, data_datum, flag));
  if (ret==0) return OK;
```

```
  if (ret==1) return RESOURCE_ALREADY_EXISTS;
  else return FATAL_SERVER_ERROR;
  };

SS_Error SSDBM::get_resource (const String& url, Resource& resource)
  {
  SS_Error error;
  Record rec;
  error = load_record (url, rec);
  if (error == OK) resource=rec.resource;
  return error;
  };


SS_Error SSDBM::load_record (const String& url, Record& record)
  {
  SS_Error ret=OK;
  SSByteStore data (0,80);
  SSByteStore key (20,4);

  key.store(url);
  key_datum.dptr = (char*) key.pointer();
  key_datum.dsize = key.used_size();

  data_datum = gdbm_fetch (records_file, key_datum);
  if (data_datum.dptr==NULL)
    {
    ret=RESOURCE_NOT_FOUND;
    }
   else
    {
    record.resource.url = url;
    data.take_over(data_datum.dptr, data_datum.dsize);
    data.get(record);
    };

  return ret;
  };


SS_Error SSDBM::urls_list (Set_String_& urls)
  {
  datum temp_key;
  key_datum=gdbm_firstkey(records_file);
  while (key_datum.dptr!=NULL)
    {
    urls.add_member(key_datum.dptr, NO_TEST_FOR_DUPLICATE);
    temp_key=key_datum;
    key_datum=gdbm_nextkey(records_file,temp_key);
    free (temp_key.dptr);
    };
  return OK;
  };


SS_Error SSDBM::subjects_list (Set_String_& subjects)
  {
  datum temp_key;
  key_datum=gdbm_firstkey(subjects_file);
  while (key_datum.dptr!=NULL)
    {
    subjects.add_member(key_datum.dptr, NO_TEST_FOR_DUPLICATE);
    temp_key=key_datum;
    key_datum=gdbm_nextkey(subjects_file,temp_key);
    free (temp_key.dptr);
    };
  return OK;
  };


void SSDBM::reorganise()
  {
  gdbm_reorganize(records_file);
  gdbm_reorganize(subjects_file);
```

```
  };


void SSDBM::text_dump (String& dump)
 // fill dump with entire subject space contents
 // format is ssci commands that reproduce the subject space, ie
 //     newsubject <subject_id>
 //     addsubject <child_id> <parent_id>
 //     newresource <url> <name> <subject> <subject> ...
 // section separator is |= at start of an otherwise empty line
 //
  {
  Set_String_ subjects;
  Set_String_ urls;
  subjects_list(subjects);
  urls_list(urls);

  subjects.next_member_returns_first();
  while (subjects.can_use_get_next_member())
    {dump+= "newsubject ";
     dump+="\""; dump+=subjects.get_next_member(); dump+="\"\n";
    };

  String tmp;
  Set_String_ parents;
  subjects.next_member_returns_first();
  while (subjects.can_use_get_next_member())
    {
    subjects.get_next_member(tmp);
    parents_of_subject(tmp,parents);
    if (parents.no_of_members()!=0)
      {
      dump+="addsubject ";
      dump+="\""; dump+=tmp; dump+="\" ";
      parents.next_member_returns_first();
      while (parents.can_use_get_next_member())
                {dump+="\""; dump+=parents.get_next_member();  dump+="\" ";};
      dump += '\n';
      };
    };

  Record record;
  urls.next_member_returns_first();
  while (urls.can_use_get_next_member())
    {
    load_record(urls.get_next_member(), record);
    dump+="newresource ";
    dump+="\""; dump+=record.resource.url; dump+="\" ";
    dump+="\""; dump+=record.resource.name; dump+="\" ";
    record.subject_ids.next_member_returns_first();
    while(record.subject_ids.can_use_get_next_member())
     {dump+="\""; dump+=record.subject_ids.get_next_member();  dump+="\" ";};
    dump+="\n";
    };

  };


SSDBM::SSDBM (SS_Error& status, const String& name,
                                        int  flags=GDBM_WRCREAT)
  {
  String temp;
  temp = name+record_file_extention;
  records_file=gdbm_open (temp.chars(),
                gdbm_file_block_size, flags, gdbm_file_mode, 0);
  //NB IGNOR THESE WARNINGS ON LINE ABOVE & BELOW
  //#warning ignore warnings around this line
  temp = name+subject_file_extention;
  subjects_file=gdbm_open (temp.chars(),
                gdbm_file_block_size, flags, gdbm_file_mode, 0);

  if ((records_file!=NULL)&&(subjects_file!=NULL))
     status=OK;
    else status=DATABASE_NOT_FOUND;
```

```
  }; //end ::SSDBM



SSDBM::~SSDBM ()
  {
  //gdbm_sync(records_file);
  //gdbm_sync(subjects_file);
  if ((records_file!=NULL)&&(subjects_file!=NULL))
      reorganise();
  if (records_file!=NULL)
      gdbm_close(records_file);
  if (subjects_file!=NULL)
      gdbm_close(subjects_file);
  }; //end ::~subject_database


SS_Error SS_Map_Manager::resources_in_node
                (Node_id node_id, Set_Resource_& resources)
  {
  SS_Error error;
  Set_Resource_ temp_res;
  Set_String_ S;
  error = intersectors_of_node(node_id, S);
  S-=node_id;
  if (error==OK)
     error = resources_in_intersection(node_id, resources);
  while ( (error==OK)&&(S.can_use_get_next_member()) )
    {
    Set_String_ temp_subs;
    temp_subs.add_member(S.get_next_member());
    error = resources_in_intersection(temp_subs, temp_res);
    resources-=temp_res;
    };
  if (error==OK) if (resources.no_of_members()==0) error = NO_SUCH_NODE;
  return error;
  };

SS_Error SS_Map_Manager::intersectors_of_node
    (Node_id node_id, Set_String_& intersectors)
  {
  SS_Error error;
  Set_String_ temp;
  error = intersectors_of_subject(node_id.get_first_member(), intersectors);
  if (error!=OK) return error;
  while (node_id.can_use_get_next_member())
    {
    error = intersectors_of_subject(node_id.get_next_member(),  temp);
    if (error!=OK) return error;
    intersectors*=temp;
    };
  return OK;
  };

SS_Error SS_Map_Manager::subjects_close_to_node
    (Node_id node_id, Set_String_& close_subjects)
  {
cout<<"\nsubjects_close_to_node\n";
  SS_Error error;
  Set_String_ temp;
  error = intersectors_of_subject(node_id.get_first_member(),  close_subjects);
  if (error!=OK) return error;
  while (node_id.can_use_get_next_member())
    {
    error = intersectors_of_subject(node_id.get_next_member(),  temp);
    if (error!=OK) return error;
    close_subjects+=temp;
    };
cout<<"\nsubjects_close_to_node return\n";
  return OK;
  };

SS_Error SS_Map_Manager::possible_unit_distance_nodes
```

```
                (Node_id node_id, Set_Node_id_&  possible_unit_distance_nodes)
 {
 SS_Error error;
 //find specialised_nodes
 Set_String_ subject_ids_for_adding_to_node_id;
 Set_String_ intersectors;
 error = intersectors_of_node
            (node_id, intersectors);
 if (error!=OK) return error;
 subject_ids_for_adding_to_node_id = intersectors;
 subject_ids_for_adding_to_node_id -= node_id;
 subject_ids_for_adding_to_node_id.next_member_returns_first();
 while (subject_ids_for_adding_to_node_id.can_use_get_next_member())
  {
  Node_id temp = node_id;
   temp.add_member(subject_ids_for_adding_to_node_id.get_next_member());
  possible_unit_distance_nodes.add_member(temp);
  };
 //find generalised_nodes
 Set_String_ subject_ids_for_removing_from_node_id = node_id;
 while (subject_ids_for_removing_from_node_id.can_use_get_next_member())
  {
  Node_id temp = node_id;
  temp.delete_member
            (subject_ids_for_removing_from_node_id.get_next_member());
  possible_unit_distance_nodes.add_member(temp);
  };
 return error;
 };


SS_Error SS_Map_Manager::neighbor_nodes
                        (Node_id node_id, Set_Node_id_&  neighbor_nodes)
 {
 SS_Error error;
 Set_String_ subject_ids;
 // error=subjects_close_to_node(node_id, subject_ids);
 error=intersectors_of_node(node_id, subject_ids);
 if (error!=OK) return error;

 for (int distance=1; ( distance<=2 )
                        &&(neighbor_nodes.no_of_members()<=10);
     distance++)
  {
  Set_Node_id_ possible_node_ids;
  generate_node_id_combinations(node_id, subject_ids, distance,
                                                    possible_node_ids);
  possible_node_ids.next_member_returns_first();
  while (possible_node_ids.can_use_get_next_member())
   {
   Node_id  possible_node_id=possible_node_ids.get_next_member();
   Set_Resource_ dummy;
   error = resources_in_node (possible_node_id, dummy);
   if (error==OK)
            neighbor_nodes.add_member(possible_node_id);
    else if (error==NO_SUCH_NODE)
            error = OK;
           else /*return error*/ error=OK;
   };
  };
 return error;
 };

void SS_Map_Manager::generate_node_id_combinations
            (Node_id& node_id, Set_String_& subject_ids, int  distance,
                                        Set_Node_id_& generated_node_ids)
 // Take the subjects in subject_ids, and use them as to generate combinations
 // of subjects that define node_ids that are the specified distance away from
 // node_id.  Does not check for null node_ids.

 {
 int i;  // i will be used in loops to move through the index
```

```
 //make index array of subject_ids;
  int last_index=subject_ids.no_of_members();
  String index[last_index+1];                  //first index is 0
  subject_ids.next_member_returns_first();
  for (i=1; i<=last_index; i++)
    index[i]=subject_ids.get_next_member();

 //make bitset from node_id;
  BitSet node_bs;
  node_bs.clear();
  for (i=0; i<=last_index; i++)
    if (node_id.has_member(index[i]))
                node_bs.set(i);

 //generate cobination vectors
  int no_of_generators = nCr(subject_ids.no_of_members(),distance);
  BitSet generators[no_of_generators];
   generate_bit_combinations(subject_ids.no_of_members(),distance,1,generators );
 //generate node_ids
  for (i=0; i<no_of_generators; i++)
    {
    BitSet result_bs;
    Node_id result_id;
    xor(node_bs, generators[i], result_bs);
    for (int b=1; b<=subject_ids.no_of_members(); b++)
    if (result_bs.test(b)) result_id.add_member(index[b]);
    generated_node_ids.add_member(result_id);
    };
  };


void SS_Map_Manager::generate_bit_combinations
                          (unsigned u, unsigned d, unsigned  s,
                                                          BitSet* result)
    // u = bitset size
    // d = bits to place
    // s = start placing this bit
    // result = ptr to BitSet=nCr(u,d)] in effect,
    // call by reference, done this way so caller can use stack  space
    // for array
  {
  unsigned result_counter=0;
  BitSet mask1;
  // inside the for loop, is a recursive call
  if (d>0)  // test for stopping condition of recursive loop
    {
   for (unsigned p=s; (p+d-1<=u); p++)
    {
    mask1.clear();
    mask1.set(p);
    unsigned  mask2s_size;
    mask2s_size = nCr(u-p,d-1);
    BitSet mask2s[mask2s_size];
    generate_bit_combinations(u, d-1, p+1, mask2s);
    for (unsigned mask2s_counter=0; mask2s_counter<mask2s_size;
                                                          mask2s_counter++)
      xor(mask2s[mask2s_counter],mask1,
                          result[result_counter++]);
    };

   };
  };
```

27 ssindexer.cxx

```
#include <iostream.h>
#include <gdbm.h>
```

```
#include <stdio.h>
#include "ssdbm.h"
#include "ss.h"


void generate_index(SSDBM ssdbm);


int main(int argc, char* argv[])
  {
  if (argc!=2) cout <<"arg 1 = database name \n";
  else
    {
    SS_Error error;
    int file_mode;
    SSDBM ssdbm(error, argv[1], GDBM_READER);
    if (error==OK)
      {
     generate_index(ssdbm);
      }
     else
     {cerr<<argv[0]<< " error opening subject space - can only open  subject space by one
program at a time\n";};
   };
  };


void generate_index(SSDBM ssdbm)
  {
  Set_String_ urls;
  ssdbm.urls_list(urls);
  String url;
  Resource resource;
  Set_String_ subjects;
  urls.next_member_returns_first();
  while (urls.can_use_get_next_member())
    {
    urls.get_next_member(url);
    ssdbm.get_resource(url, resource);
    ssdbm.subjects_of_resource(url, subjects);
    cout << "<A href=\"" << url << "\"> ";
    cout << resource.name << "</A>";
    cout << "  : ";
    subjects.next_member_returns_first();
    while (subjects.can_use_get_next_member())
                cout << subjects.get_next_member() << " ";
    cout << "<P>\n";
    };
  };
```

28 test-ss.dump

```
newsubject "cern"
newsubject "manual"
newsubject "guide"
newsubject "simon"
newsubject "web"
newsubject "www"
newsubject "home"
newsubject "ncsa"
newsubject "mosaic"
newresource "http://info.cern.ch/hypertext/WWW/TheProject.html" "CERN : The
World Wide Web project" "cern" "home" "web" "www"
newresource "http://www.rpi.edu/Internet/Guides/decemj/text.html" "Internet  Web
Text" "web" "www" "home" "guide"
newresource "http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/
NCSAMosaicHome.html" "NCSA Mosaic Home Page" "home" "web" "www" "mosaic"
"ncsa"
```

```
newresource
"http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/mosaicdocs.html" "NCSA
Mosaic for X Documentation" "mosaic" "manual"
newresource "http://web.elec.qmw.ac.uk:12121/simon.html" "simon manual "  "simon"
"manual"
newresource "http://web.elec.qmw.ac.uk:12121/" "SIMON homepage" "simon"  "web"
"www" "home"
```

29 text-dump-0.3-format-ss.cxx

```
#define _ByteStore_platform_dependant

#include <iostream.h>
#include <gdbm.h>
#include <stdio.h>
#include "ssdbm.h"
#include "ss.h"
#include "ssci-lang.h"


int main(int argc, char* argv[])
  {
  if (argc!=2) cout <<"arg 1 = database name \n";
  else
    {
    SS_Error error;
    SSDBM ssdbm(error, argv[1], GDBM_READER);
    if (error==OK)
      {
    String dump;
    ssdbm.text_dump(dump);
    cout << dump;
      }
     else
    {cerr<<argv[0]<< " error opening subject space - can only open  subject space by one
program at a time\n";};
   };
  return 0;
  };
```

30 upgrade-from-0.5-to-0.6

```
#! /bin/sh

#This script upgrades any binary or source simon.0.5 to simon.0.6


cat simple-simon.tcl | sed -e 's/pack .button_bar_3 -fill x -expand true/pack
.button_bar_3 -fill x -side bottom/' > simple-simon.tcl.0.6

rm simple-simon.tcl
mv simple-simon.tcl.0.6 simple-simon.tcl

echo "This directory should now contain simon version 0.6"
```