The Astronomical Instrumentation Software Specification

Aaron Smith

February 25, 2004

Rev: 1.0-pre1

The Astronomical Instrumentation Software Specification is intended to provide guidelines for the development of software for astronomical instruments. It is hoped that with this document instrumentation designers will be able to better unify the existing hardware to a point that hardware dependence is not a requirement for software.

1 Introduction to the Specification

1.1 What is in the specification

The specification takes advantage of the fact that software for astronomical instrumentation has to perform some rudimentary functions, and has to supply the results of these functions so that they are meaningful to the observer. The Specification outlines the data structures, and the functions to operate on those data structures. In order to quantify these structures the conventions of ANSI C will be used, however implementations are not limited to the C language. The specification also requires for each unique class of instrument a driver that can be made available, via static link or dynamic link, to any software program the user desires to build on top of it. This is a strict requirement of the specification.

1.2 Typing

1.2.1 Types

The aiss relies on a few types being defined. In this section will explain what those types are, they are not incredibly strict, be careful to document any deviations from this standard.

data_format type
values allowed: UINT8, UINT16, UINT32, FP32, FP64, INT8, INT16, INT32
This data type is meant as a flag in the Image data structure to allow programs to correctly interpret the values of data coming from an imaging device.

STR20
This is nothing more than the old programmers trick, it is a quick way of specifying a array of 20+1 characters.

2 Data Structures

2.1 The Image data Structure (a_Image)

The image data object is designed to interpolate well into the FITS standard as well and provide any useful information to the system.
The Image structure must contain the flowing:

fits_head
A string of 2880 characters. This contains the header for the image to be written to a FITS file. You should write this using the fits_head() function.

bitpix
The number of bits per pixel

pix_format
A variable of the type **data_format** which specifies the exact the type of data contained in payload.

rows
The integer number of rows in the two dimensional image

columns
The integer number of columns in the image

payload
This is the image data itself in the format specified by the *pix_format* variable, it is recommended you use a void pointer to specify an address for data start.

flags
A string of flag characters that is used to indicate information about the image, this information can be arbitrary, however the first ten characters of flag are reserved for the driver and the second ten are reserved for the program. **It is strongly recommended that any thing your program or driver places in the flag string be documented to the fullest extent possible.**

2.2 The pointing_object structure (a_pointing_object)

The pointing object structure is intended to contain data needed for pointing a telescope however this can generally be used to store information about the current position of the telescope, most programs should have two of the structures, one describing the telescopes position the other describing where the telescope should be pointed.

alt
A floating point number describing the altitude of the telescope in degrees if this is not reported or not used it should be set to -1.

az
A floating point number describing the telescopes azimuth position in degrees. If this is not in use its value should be set to -1.

ra
The right ascension of the telescope in decimal hours. If this data value is not in use it should be set to -1.

dec
The declination of the telescope in decimal degrees. If this data value is not being used it should be set to -1.

equinox
The equinox of the object coordinates in decimal years. This value is mandatory when using *ra* and/or *dec* however if using *alt* or *az* this value is ignored.

object_name
A character string that is the name of the object if not know set to NULL. This is not mandatory but the implementor is encouraged to use this string.

ppm
A floating point number indicating the objects proper motion in arcseconds per year.

epoch
The epoch of the coordinates, for precession and proper motion calculation.

2.3 The instrument state structure (a_state)

The instrument state structure contains all information about the instrument, in its current form. Some of this data may in fact change during the course of observing, some is read from the config at program initialization. The a_state structure contains:

name
A **STR20** array containing the name of the instrument.Read from "aiss.conf" or "aiss.ini".

observatory
A character pointer which in the name of your observatory read from "aiss.conf" or "aiss.ini".

data_type_default
This is the default type that data coming from your imager has, the variable is of type **data_format**.

cds_rows
The integer number of rows in a correlated double sampling mode. This value does not necessarily have to be used, if it is not used then it should be set to -1.

cds_column
The integer number of columns in a correlated double sampling mode. This value does not necessarily have to be used, if it is not used then it should be set to -1.

p_rows
The integer number of rows on your array, this value is mandatory for all imaging devices.

p_columns
The physical number of columns, this number is mandatory for all imaging devices.

default_sub

Whether or not correlated double sampled images will be subtracted by default. This can be either **boolean** or integer.

flags
A character string of driver specific flags.

numbfw
The integer number of filter wheels in your instrument.

numbfilters
The integer number of filters you have in your system.

filternames
This is an array of **STR20** variables of length *numbfilters*. The index of this corresponds to the order in which filters are input into "aiss.conf" or "aiss.ini".

2.4 Bulk Image statistics structure (a_Stat)

This header contains bulk image statistics.
m1,m2,m3,m4
These variables contain the median values for each quadrant of the image.

mu1,mu2,mu3,mu4
These floating point data values contain the mean for each quadrant in the image.

3 Functions

the function definitions are the bread and butter of this specification, what the functions look like is irrelevant so long as the definitions are **strictly** adhered to.

3.1 Driver Access functions

int driver_init(void)*
This function should assign all necessary variable values, and prepare the instrument for use.

int shutter()
This function will open the shutter on the instrument, this is not strictly necessary. The function returns 0 on success and -1 on failure. If the function is not implemented simply Print an error message and return 0.

int expose(int time, a_image)*
This is the entrance point to grab an image from an imaging device. *time* is the exposure time in milliseconds. *a_image** is a pointer to an Image data struct. (defined in section 2.1) The function returns 0 on success and -1 when there is a failure.

int tcs_get(a_pointing_object)*
This is the function for filling a pointing_object with data from the Telescope.Return 0 on success and -1 on an error.

int tcs_send(a_pointing_object)*
This is the function to send new coordinates to the TCS for goto. Returns 0 on success, -1 on failure.

*int tcs_com(char *)*
This is reserved for sending individual commands and command sequences to the TCS. The driver maintainer should provide this interface, but it should not be relied upon in applications without some sort of translation layer. Primarily used for scripting. Returns 0 on success, -1 on failure.

int fw_home(int fw)

This function will home the filter wheel designated by *fw* in the global a_State struct. it is not strictly necessary, if it is deprecated in your driver make sure to document that fact.

int fw_setpos(int fw,int pos)
This is the function to move the filter wheel *fw* to a position *pos*

3.2 Functions for Data Output

*int fits(a_image *data, char* fname)*
This function will convert an a_Image structure into a fits file titled *fname*.

int fits_head(a_Image,a_pointing_object *)*
This function writes a full fits file from the data contained in the a_Image file and places the information for the image subject in the header from the a_pointing object.

*int tocfitsio(a_Image *, char *fname)*
This function creates a fits file from an aiss image file using cfitsio.

*int fromcfitsio(char *fname, a_Image *data)*
This function fills an aiss image struct with data read from a FITS file using cfitsio.

3.3 Data operation Functions

*int image_init(a_image *)*
This function initializes an image for use in an application, it takes information from the global a_State array. Returns 0 on success and -1 on failure.

a_Image im_add(a_Image *a, a_Image *b)*
This function will produce an addition of two aiss images, with the proper check that the images are matched in size.

a_Image im_sub(a_Image *a, a_Image *b)*

This function will produce an subtraction of two aiss images, with the proper check that the images are matched in size.

a_Image im_multi(a_Image *a, a_Image *b)*
This function will produce an multiplication of two aiss images, with the proper check that the images are matched in size.

a_Image im_divide(a_Image *a, a_Image *b)*
This function will produce an division of two aiss images, with the proper check that the images are matched in size.

4 Configuring the system

AISS relies on some semi permanent information for use in its regular applications. Here is a sample config file:

```
#This file is the aiss config it was generated by confgen do not edit.
NAME= FANNICAM
OBS_NAME= FMO
CDS_ROWS= 2048
CDS_COLUMN= 1024
PHYSICAL_ROWS= 1024
PHYSICAL_COLUMNS= 1024
BITPIX= 32
DEFAULT_SUB= T
DATA= 1
NUM_FW= 3
NUM_FILTERS= 20
FILTERW_1= 4
FILTERW_2= 12
FILTERW_3= 20
FW_1= Field Stop
FW_2= Block
FW_3= 3" Slit
FW_4= 1" Slit
FW_5= Open
FW_6= Block
FW_7= Open
FW_8= J
FW_9= H
FW_10= Ks
FW_11= FEII
FW_12= Br-Gamma
FW_13= Pa-Beta
FW_14= POL I
FW_15= POL II
FW_16= R=300 Grism
FW_17= R=30 Grism
FW_18= Open
```

```
FW_19= Block
FW_20= Open
```

5 The sequencing system

The sequencing system is primarily intended to help with the task of reducing data. Examples are image processing, track and stack processing, photometry, RBG or LRBG color combining. The system described in this section, has the goal of making these tasks easier or even automate them completely. All images are assumed to be fits images.

5.1 The Sequence file "*.seq"

The sequence file is generated by a program after a series of related images is taken. It is the application programmers choice when to break a sequence of images, but it is logical to allow that if a discrete set of images of the same exposure time are specified they should go into their own .seq file. The seq file contains a set of file names followed by the '/' delimiter. A program parsing the .seq file might simply count the number of delimiters in the file to determine the number of files. This leads to the generation of a sequence meta file or the processing of a sequence by a users program, or the sequence file is read once to determine the allocation of memory for image buffers, then the buffers are read into the reduction program.

5.2 The Sequence Meta File "*.smf"

The sequence meta file can do two things, one make the construction of an a_Image structure easy by specifying its components, or it can contain the data that is to be processed. The former makes the .smf file not strictly necessary, the second makes the .smf very large. It is up to the programmer what type of .smf is generated. Make no mistake one should not assume that the meta files are consistent across applications, they are used primarily to facilitate automated data reduction across a package of reduction tools.

6 Appendices

6.1 Sample Functions

These are sample implementations taken from the AISF or Astronomical Instrumentation Software framework.

6.1.1 aisf.h

The following is the header defining the implementation of AISS:

```
#ifndef AISF_H
#define AISF_H
#ifndef TRUE
#define TRUE 1
#endif
#ifndef FALSE
#define FALSE 0
#endif

enum Image_Data_Type{INT16,INT32,INT8,FP32,FP64,UNIT8,UINT16,UINT32};
typedef char STR20[21];

//image structur from section 2.1 of AISS
typedef struct
{
  char fits_head[2880];
  STR20 flags;
  int bitpix;
  enum Image_Data_Type pix_format;
  int rows;
  int columns;
  void *payload;
}a_Image;

//telescope pointing data structure AISS section 2.2
typedef struct
{
  float alt;
```

```
  float az;
  float ra;
  float dec;
  float equinox;
  char *object_name;
  float ppm;
  float epoch;
}a_point;


//a_state struct specified in section 2.3 of AISS
typedef struct
{
  STR20 name;
  STR20 observatory;
  enum Image_Data_Type d_type_default;
  int cds_rows;
  int cds_columns;
  int p_rows;
  int p_columns;
  int sub;
  int numfw;
  int *breaks;
  int numbfilters;
  int bitpix;
  STR20 *f_names;
} a_state;

typedef struct
{
  long m1;
  long m2;
  long m3;
  double m4;
  double mu1;
  double mu2;
  double mu3;
  double mu4;
```

```
}a_Stat;

//Global Data
int exp_timer;

//Global Structs
a_state aiss_sys;
a_point tcsin;
a_point tcsout;


//system function defs (Is this all there is in the API?
int aiss_sys_init(a_state *);
int image_init(a_Image *);
int driver_init(void *);
int expose(a_Image *data, int exptime);
int tcs_get(a_point*);
int tcs_send(a_point*);
int tcs_com(char *);
int fw_home(int);
int fits(a_Image *data,char * fname);
int fits_head(a_Image *data, a_point *);
int fits_read(a_Image *, char *);
#endif
```

6.1.2 aiss_sys_init

```
 int aiss_sys_init(a_state *data)
{
  FILE *config;
  char tmp;
  char crap;
  char tempbreaks[3];
  int i,a;
  config=fopen("/etc/aiss.conf","r");
  if(config==NULL)
    {
      printf("Error 001 Config file not found.\n");
```

```
    return -1;
  }

while(crap !='\n')
  crap=(char)fgetc(config);


fscanf(config,"NAME= %s", (char *) &data->name);
crap=(char)fgetc(config);

fscanf(config,"OBS_NAME= %s",(char *) data->observatory);
crap=(char)fgetc(config);

fscanf(config,"CDS_ROWS= %d",&data->cds_rows);
crap=(char)fgetc(config);

fscanf(config,"CDS_COLUMN= %d",&data->cds_columns);
crap=(char)fgetc(config);

fscanf(config,"PHYSICAL_ROWS= %d",&data->p_rows);
crap=(char)fgetc(config);

fscanf(config,"PHYSICAL_COLUMNS= %d",&data->p_columns);
crap=(char)fgetc(config);

fscanf(config,"BITPIX= %d",&data->bitpix);
crap=(char)fgetc(config);

fscanf(config,"DEFAULT_SUB= %c",&tmp);
crap=(char)fgetc(config);

if(tmp=='T')
  data->sub=TRUE;
else if (tmp=='F')
  data->sub=FALSE;
else
  {
    printf("Error parsing config, invalid subtraction def. ERR 002\n");
```

```
    return -1;
  }

fscanf(config,"DATA= %d", (int *)&data->d_type_default);
crap=(char)fgetc(config);

fscanf(config,"NUM_FW= %d",&data->numfw);
crap=(char)fgetc(config);

data->breaks=calloc(data->numfw,sizeof(int));

fscanf(config,"NUM_FILTERS= %d",&data->numbfilters);
crap=(char)fgetc(config);

for(i=1;i<=data->numfw;i++)
  {
    for (a=0;a<11;a++)
tmp= (char)fgetc(config);
    tempbreaks[0]=fgetc(config);
    tmp=(char)fgetc(config);
    if (tmp=='\n')
{
  data->breaks[i-1]=atoi(&tempbreaks[0]);
  continue;
}
    else
{
  tempbreaks[1]=(char)fgetc(config);
  data->breaks[i-1]=atoi(tempbreaks);
}
  }
fclose(config);
return 0;
}
```

6.1.3 image_init

```
int image_init (a_Image *data)
```

```
{
  if(aiss_sys.sub==FALSE)      // these conditions set the size of the image
    {
      data->rows=aiss_sys.cds_rows;
      data->columns=aiss_sys.cds_columns;
      data->flags[10]='F';
    }
  else if(aiss_sys.sub==TRUE)
    {
      data->rows=aiss_sys.p_rows;
      data->columns=aiss_sys.p_columns;
      data->flags[10]='T';
    }
  else
    {
      printf("It would appear that the system configuration has not been read e
      return 1;
    }                         //display if init !called first

  data->bitpix=aiss_sys.bitpix;                  //copy over number of bits per
  data->pix_format=aiss_sys.d_type_default;      //copy image data type default


  /*********************************************************************
   This switch system allocates the proper size for payload based on the data t
  **********************************************************************/

  switch(data->pix_format)
    {
    case INT16:
      {
data->payload=calloc(data->rows*data->columns,sizeof(short));
break;
      }
    case INT32:
      {
data->payload=calloc(data->rows*data->columns,sizeof(long));
break;
```

```
      }
    case INT8:
      {
data->payload=calloc(data->rows*data->columns,sizeof(char));
break;
      }
    case FP32:
      {
data->payload=calloc(data->rows*data->columns,sizeof(float));
break;
      }
    case FP64:
      {
data->payload=calloc(data->rows*data->columns,sizeof(double));
break;
      }
    default:
      {
printf("Image data type flag invalid.");
return 1;
      }
    }

return 0;
}
```

6.2 The FDL

```
                GNU Free Documentation License
                  Version 1.2, November 2002


 Copyright (C) 2000,2001,2002  Free Software Foundation, Inc.
     59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.


0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document "free" in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of "copyleft", which means that derivative
works of the document must themselves be free in the same sense.  It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does.  But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book.  We recommend this License
principally for works whose purpose is instruction or reference.


1. APPLICABILITY AND DEFINITIONS
```

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
```

```
    or any later version published by the Free Software Foundation;
    with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
    A copy of the license is included in the section entitled "GNU
    Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts,
replace the "with...Texts." line with this:

    with the Invariant Sections being LIST THEIR TITLES, with the
    Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other
combination of the three, merge those two alternatives to suit the
situation.

If your document contains nontrivial examples of program code, we
recommend releasing these examples in parallel under your choice of
free software license, such as the GNU General Public License,
to permit their use in free software.
```