XModule

COLLABORATORS

	TITLE : XModule		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 3, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

XModule

1.1 XModule.guide

This manual documents XModule version 3.0 released on 06 Jan ↩
95.

   Copyright (C) 1993-1995 by Bernardo Innocenti, Freely distributable,
all rights reserved.

Overview
            An overview of XModule.

Distribution
        Licence & Disclaimer.

Getting Started
   How to run XModule.

User Interface
     General notes on the user interface.

Panels
            Description of all windows.

Format Conversion
 Brief description of module conversion with XModule.

Optimization
        How to save memory and disk space with modules.

ARexx Interface
    Every program should have one nowadays, doesn't it?

Known Bugs
            Notes on things that are still not working...

1.2 XModule.guide/Overview

Overview
****

   The Amiga computer has standard file formats for almost everything.
Graphics, text, animation, hypertext and many other data types are
easily imported into any program that supports the given standards, but
this isn't true for music. Too bad the music interchange file format
defined by Electronic Arts (the old IFF SMUS) was too poor to be usable
in high-quality music sequencers. Therefore, every music editor
created its own proprietary module format, so that, for instance, you
cannot load into ProTracker music written with Oktalyzer.

   XModule (pronounced Cross Module) born as music module conversion
utility that tries to solve this problem. The project started in early
1993, as a friend of mine (Fabio), who wrote a lot of (good) music on
Oktalyzer, needed ProTracker to obtain certain effects. He had an old
PD conversion program called Tracker that worked only in particular
conditions and was really bogus. So I decided to contact the author
and he let me have the source for Lit30000 (about $20).

   The original program has been completely rewritten from scratch, as
the source was poorly coded. Now XModule sports many other features,
like module optimization, a nice user interface and complex module
editing functions.

   This is why I like to call XModule a module processor. XModule is to
music processing what ADPro and ImageFX are to image processing (I am
not very modest :-).

1.3 XModule.guide/Distribution

```
                 Distribution
************
```

   XModule is a Freeware program. This means that you (probably) have
got it free and you should not ask any money if you decide to copy it
for someone else. If, instead, someone sold you XModule for a price
higher than the media used to store it on (i.e. $1 for a DD disk), tell
him he should try making money honestly.

   I'm not asking you to pay for XModule because I never paid for a
ShareWare program, so I felt that I somehow had to pay my debt and I
decided to distribute XModule as free software, a little gift to all
the Amiga community. Feel free to distribute XModule to your friends
and enemies, but, if you do so, you must keep the documentation and the
other distribution files together with the executable. The source code
is not required, but I'm distributing it to let others take advantage
of my work, so, if possible, try to include the source too.

   Fred Fish, SAN, ADS and Aminet are especially granted permission to
include XModule into their PD libraries. Other PD libraries are
welcome as well.

   If you intend to include XModule or parts of it in a commercial
package, please at least send me a free copy of the product. I would
be honored to see one of my programs being part of a commercial package.

   You can modify or improve XModule (see
                 Programmers
                 ), only be so kind
to return the new source code to me, so I can put the new features in
the next release. And, please, do not strip my name from the
documentation, the program source, or the executable.

   This product is provided as is, without warranties of any kind: the
author of this program cannot be held liable for any defects in the
executable nor in the documentation or in any other files contained in
this package. Any damage directly or indirectly caused by the
use/misuse of XModule is the sole responsibility of the user her/him
self. In other words: XModule never formatted my hard disk nor killed
my dog, so I suppose it won't cause any damage to your system either,
but, just in case, do not blame me if something nasty happens.

1.4 XModule.guide/Getting Started

```
                 Getting Started
***************
```

   XModule requires Kickstart version 2.0 or greater. The reason is
that I hate programming on 1.3 and I won't do it any more. If you still
haven't upgraded to 2.0, you'll miss the chance to run XModule and many

other wonderful programs. So what are you waiting for?

   XModule also takes advantage of some 2.1 and 3.0 features, such as
DataTypes, AmigaGuide, Locale and many enhancements for the user
interface.

   You can invoke XModule either from the Shell or Workbench. XModule
accepts both command line and ToolTypes arguments. Workbench and Shell
parameters are the same. The full AmigaDOS template is:

     XModule FROM/M,PUBSCREEN/K,PORTNAME/K,SETTINGS/K,
       CX_POPUP/T,CX_POPKEY/K,CX_PRIORITY/K/N,
       ICONXPOS/K/N,ICONYPOS/K/N,ICONNAME/K

The FROM keyword specifies modules to load at startup time. Standard
AmigaDOS wildcards can be used to match more than one module. From
Workbench, you multi-select module icons instead of specifying a
ToolType.

   Specifying the PUBSCREEN keyword, followed by a public screen name,
allows you to open XModule on an existing public screen. The given
name should match exactly (case matters), or the system won't be able to
find the requested screen. If the requested screen does not exist, a
new public screen will be opened, cloning the resolution and colors of
the default public screen (usually the Workbench screen). Specifying
an empty string (eg.: "") you will automatically pick the default
public screen. Note that this behavior is not standard: most
commodities will abort if the requested public screen does not exist.
The default is to use or create a public screen called XModule.

   PORTNAME will change the name of XModule's ARexx port. If a port
with this name already exists, a number (eg: .1) will be appended to
the name, until a unique name is found. You can disable the ARexx port
providing an empty string (eg.: "") for PORTNAME. The port name
defaults to XMODULE. For more information about the ARexx port, consult
the
                 ARexx Interface
                  chapter.

   SETTINGS causes XModule to use the given settings file instead of
looking for the default preferences file.

   CX_POPUP controls wether the user interface should show as soon as
XModule starts. The default is to always show the interface. When used
from the shell, CX_POPUP acts as a toggle, turning off the user
interface. From Workbench, you specify CX_POPUP=YES or CX_POPUP=NO to
show and hide the user interface, respectively. See
                 Commodities
                  .

   CX_POPKEY specifies the hotkey that pops up XModule's user interface.
The given string should be a valid Commodities input description. You
can disable XModule's Commodities broker specifying an empty string
(eg.: "") for CX_POPUP. The default is ctrl alt x. See
                 Commodities
                  .

   CX_PRIORITY sets the Commodities priority for the XModule broker.
An high priority causes the input to be processed by {No Value For
"prgnmae"} before other commodities with a lower priority. The priority
ranges from +127 to −128. The default priority is 0. See
                  Commodities
                      .

   ICONXPOS specifies the X position of XModule's AppIcon. The default
is to let Workbench choose a suitable position for the icon.

   ICONYPOS specifies the Y position of XModule's AppIcon. The default
is to let Workbench choose a suitable position for the icon.

   ICONNAME specifies the name of XModule's AppIcon. The default name
is XModule.

1.5 XModule.guide/User Interface

                  User Interface
****

   XModule's GUI follows Commodore's 2.0 style guidelines. Under
Kickstart 3.0 and above, XModule takes advantage of new OS capabilities
to enhance some aspects of the GUI.

                  Windows Layout

                  Shortcut Keys

                  On-Line Help

                  Localization

                  Commodities

                  Default Icons

1.6 XModule.guide/Windows Layout

Windows Layout
==============

   Gadget and menu layout is font sensitive, but using some fancy fonts
could result in a somewhat weird look. If you are using a very big
font and a window becomes too big to fit in the screen, topaz/8 will be
used as a fallback for rendering that window.

When you close a window, it will remember its position and size when
you open it again. The same is true for window zooming.

1.7 XModule.guide/Shortcut Keys

                    Shortcut Keys
=============

   In all windows, the following shortcut keys are active:

ESC
     Close the active window. Does not work with ToolBox window, as it
     would make it too easy to accidentally exit the program.

HELP
     Bring up AmigaGuide on-line help. See
                 On-Line Help
                     .

TAB
     Cycle forward through string and numeric gadgets within the active
     window. SHIFT-TAB will cycle backwards. TAB also activates the
     first string or numeric gadget if none is selected.

RETURN
     Activate the first string or numeric gadget in the active window.

Cursor UP/DOWN
     Move up and down in a viewlist whenever the active window contains
     one. SHIFT-Cursor moves 5 items up/down. ALT-Cursor moves to the
     top or to the bottom of the list. This shortcut requires at least
     Kickstart 3.0.

   In addition, gadgets with underscored characters in labels, can be
operated hitting the highlighted key. SHIFT-key acts the opposite of
the unshifted key (sliders are decremented by one, cycle gadgets will
cycle backwards, etc). Keyboard shortcuts will not work for all gadget
kinds under Kickstart 2.0.

1.8 XModule.guide/On-Line Help

On-Line Help
============

   This feature requires amigaguide.library, which is distributed with
Workbench version 2.1 and up, and XModule.guide located in the current
directory or in the same directory of XModule's executable or in the
HELP:language/ directory, where language is any of your Locale
preferred languages. AmigaGuide(TM) is also available as a freely

distributable package for 1.3-2.0 users.

   To get help on a gadget, press HELP while the mouse pointer is over
the gadget. To get help on a string gadget, activate it and press
HELP. Hitting HELP while the mouse isn't over a gadget, will give you
help on the active window.

   To get help on a menu item, hilight the item and press HELP while
keeping the right mouse button pressed.

1.9 XModule.guide/Localization

```
Localization
============
```

   Localization requires locale.library, which is distributed with
Workbench version 2.1 and up, and the catalog file XModule.catalog
located in the same directory of XModule's executable or in the
LOCALE:catalogs/language/ directory, where langage is is any of your
Locale preferred languages.

1.10 XModule.guide/Commodities

```
Commodities support
===================
```

   XModule is a Commodity and can be controlled by the Commodities
Exchange program.

   XModule will pop up when its hotkey is typed; the main window will
activate and the screen will move to the front. If XModule's interface
is hidden, it will be revealed.

   The user interface can be hidden with the Exchange program. When
the user interface is hidden, the only ways you have to still operate
XModule is through the ARexx port or the AppIcon.

   You can also control XModule by sending the following signals to its
process:

*CTRL-C
     Quit XModule.

*CTRL-D
     Disable the Commodities broker (equivalent to selecting Inactive
     with Commodities Exchange).

*CTRL-E
     Enable the Commodities broker (equivalent to selecting Active with
     Commodities Exchange).

```
*CTRL-F
     Deiconify XModule (equivalent to double-clicking on the AppIcon).
```

1.11 XModule.guide/Default Icons

```
Default Icons
=============
```

   When XModule creates an icon for a file, it looks for the user
preferred icon for the type of data being saved.

   The template icons can be placed in the same drawer where XModule
resides or in the ENV:sys/ directory. In the latter case, you should
store them in ENVARC:sys/ to have them copied to ENV:sys/ every time
the system boots up.

   Template icons should be only of type Project. When XModule does
not find the template icon, it uses the default Project icon instead.

   Whenever the default icon does not have its own default tool, it
will be set to XModule's executable.

   The following default icons are supported:

```
-def_Module
-def_Instrument
```

1.12 XModule.guide/Panels

```
                Panels
******
```

   Sorry, XModule's documentation is still incomplete. The next release
will hopefully be more accurate.

   But wait! Do not commit suicide for this reason: XModule is very
easy to use, and you will be able to operate it even without this
chapter.

```
                ToolBox
```

1.13 XModule.guide/ToolBox

```
                         ToolBox Window
==============

Gadgets
-------

Patterns (P)
     This option has not been implemented yet.

Instruments (I)
     Opens or activates the Instruments information window.

Sequence (Q)
     Opens or activates the Sequence editor.

Songs (S)
     Opens or activates the Song information window.

Optimization (O)
     Opens or activates the
                 Optimization
                  window.

Play (P)
     This option has not been implemented yet.

Project Menu
------------

New (Amiga-N)
     Creates an empty song.

Open (Amiga-O)
     Opens a module over the current one.

Save (Amiga-S)
     Saves the current song to the same path it was loaded from.

Save As (Amiga-A)
     Opens a file requester and saves the current song to the selected
     path.

Clear Module (Amiga-C)
     Opens or activates the Clear Module panel.

Join Modules (Amiga-J)
     This option has not been implemented yet.

About (Amiga-?)
     Brings up a requester showing various information on XModule.

Iconify (Amiga-I)
     Closes all open windows and screens and puts an AppIcon on the
     Workbench.  You can De-Iconify XModule by double clicking on the
     icon.
```

Quit (Amiga-Q)
     Quits XModule.

1.14 XModule.guide/Format Conversion

                   Format Conversion
****

    XModule can load and save modules created by other music editors.
The following is a list of all supported module formats:

                 XModule

                 NoiseTracker

                 ProTracker

                 SoundTracker

                 StarTrekker

                 UNIC-Tracker

                 FastTracker

                 Oktalyzer

                 MED

                 OctaMED

                 ScreamTracker

                 MIDI File
                    As a matter of fact, there are things that simply cannot be ←
                       done
because of the big differences between different module formats. Below
is a list of the internal limitations of each module format. When you
save to a particular format and the module exceeds one of these values,
or makes use of features not implemented, some data will be stripped
away to produce at least a partial conversion.

Program	Instr	MaxPatt	PattLen	SongLen	InstrLen	Tracks
Oktalyzer	36	128	128	128	128K	4
Sound Tracker	15	64	64	128	64K	4
Noise Tracker	31	64	64	128	64K	4
StarTrekker	31	64	64	128	64K	4/8
Pro Tracker	31	100	64	128	64K	4
MED/OctaMED	63	256	256	256	No Limit	1-64

```
   Where:
   #Instr   is the maximum number of instruments,
   MaxPatt  is the maximum number of patterns,
   PattLen  is the maximum number of lines per pattern,
   SongLen  is the maximum number of positions in a song,
   InstrLen is the maximum size of an instrument in bytes,
   Tracks   is the maximum number of tracks.
```

   Effects are the hardest thing to convert. While Speed and Volume
are almost the same between different music editors, some effects (such
as Oktalyzer's H and L) cannot be converted because there are no
equivalents in other music editors. Therefore, you will have to modify
the score manually if you intend to convert a module while keeping such
effects. In addition, some effects behave in different ways even if
they claim to do the same thing.

   Last but not least, some music editors support synthetic or hybrid
instruments. XModule does not convert such instruments, because I can
hardly think of a way to implement a sample to synth conversion (any
ideas?). I used to like synthetic music, but it seems that all those
great composers on the good old C64 have vanished... Why have you
musicians become so lazy? 8-(

1.15   XModule.guide/NoiseTracker

```
NoiseTracker
============
```

   Noise/ProTracker 31 instruments module. This is absolutely the most
common module format; almost any music editor and module player (even
the crap ones for the PeeCee) understand this format. Noise/ProTracker
modules can be recognized by looking for the ID M.K. at offset 1080
($438 in hex) in the file. M.K. are the initials of Mahoney & Kaktus,
the two guys who designed this format.

   Unfortunately, there is an infinite variety of standard NoiseTracker
modules with fancy IDs. In fact, most tracker clones put their own ID
instead of the standard one even if there is no reason to do it. If
you find such a module, try replacing the ID with an hex editor and you
will be able to load it most of the times.

1.16   XModule.guide/ProTracker

```
ProTracker 100 Patterns
=======================
```

   XModule will automatically switch to ProTracker 100 whenever the
module being saved exceeds the 64 patterns limit of NoiseTracker

modules. Modules with more than 64 patterns have the ID M!K! instead
of the standard Noise/ProTracker ID (M.K.).

   Note: You must load such a module with ProTracker 2.3 or better in
order to play/edit it. Older players will either refuse to load it or
they'll turn the module into scratch dance.

   ProTracker 3.0 has a completely new file format, but this isn't
supported yet, as I've never seen a module of this kind yet.

1.17 XModule.guide/SoundTracker

```
SoundTracker 15 Instruments
===========================
```

   This format is pretty useless as SoundTracker is becoming really
rare. Do not save SoundTracker modules unless you really need to,
because most players/editors will stop supporting this format very soon.
There is no way to identify a SoundTracker module, therefore XModule
will ask you to confirm a SoundTracker module when the file being
loaded matches no other known module format.

1.18 XModule.guide/StarTrekker

```
StarTrekker
===========
```

   StarTrekker is another NoiseTracker clone, which is capable of
playing modules with 4 or 8 channels and supports MIDI devices. The
format of 4 channels modules is the same of NoiseTracker, except for
the FLT4 ID. Modules with 8 channels have the ID FLT8, and keep the
extra data in a second file, whose structure I don't know. If you want
support for 8 channels StarTrekker modules, please send me some
information on its format.

1.19 XModule.guide/UNIC-Tracker

```
UNIC-Tracker
============
```

   Actually, I've never seen UnicTracker, but I sometimes ripp Unic
modules with ExoticRipper, so I decided to give support for this weird
Tracker clone too. The format is very similar to that of ProTracker
modules, except for the EMW3 ID.

1.20   XModule.guide/FastTracker

```
FastTracker
===========
```

   These modules are Standard Noise/ProTracker modules except that they
have 6 or 8 tracks. This format comes from a damn PeeCee tracker,
which I've never seen. Anyway, adding this format was a matter of five
minutes. XModule does not suppor FastTracker 2.0 modules (those with
extension .xm) yet.

1.21   XModule.guide/Oktalyzer

```
Oktalyzer 1.1
=============
```

   Oktalyzer is an old editor which can play up to 8 channels by mixing
instruments in real time. Oktalyzer modules start with the ID
OCTASONGGMOD. Development of Oktalyzer has been discontinued, as the
author seems to be programming on Windoze now. It's really a shame, as
many musicians found that Oktalyzer was one of the most powerful
trackers for the Amiga.

1.22   XModule.guide/MED

```
MED
===
```

   Not all features of MED are supported by XModule. In particular,
synthetic and hybrid instruments are ignored.

1.23   XModule.guide/OctaMED

```
OctaMED
=======
```

1.24   XModule.guide/ScreamTracker

```
ScreamTracker
=============
```

   ScreamTracker 3.01 is a damn PeeCee program made by those Future Crew
guys who are trying to turn clones into Amigas. ScreamTracker has the

amazing feature of playing 32 tracks, 16 of which are for 8 bit digital
samples. The other 16 tracks play ADLib FM synths (not supported by
XModule.

1.25 XModule.guide/MIDI File

MIDI File
=========

   MIDI is the industry standard musical score format (the same way
Windoze is the industry standard graphic interface :-). The MIDI
concept is very different from tracker modules. There are no sequence
and patterns, but a long stream of events (notes) that have to be sent
to some musical device after a specific amount of time is elapsed.
There are no instruments. Instead, each note is associated with a
preset number that matches a particular instrument.

   XModule's MIDI file saver is based on Andrew Scott's MidiMod program.

1.26 XModule.guide/Optimization

                Optimization
****

   Module optimization is achieved by removing all unused data in such
a way that the module will sound the same as before. XModule can
process a module to reduce its size as much as possible. It is
generally a good idea to optimize a module before distributing it, but
never while you are still composing it, because you could easily loose
some data that you didn't want to discard. Do not expect to gain a lot
of bytes optimizing modules: saving 20% of the total size is usually a
very good result.

* Unused patterns
     Some formats (e.g.: Sound/Noise/ProTracker) store empty patterns
     in a module if a pattern with a greater number is used. For
     instance, a song using patterns 1, 2 and 5 will contain patterns 3
     and 4 as well.

     Warning: If a song contains a part that has not been finished yet
     and has not already been inserted in the position table, you will
     lose it.

     Warning: When a song is saved to Noise/ProTracker, patterns beyond
     the last used patterns are discarded regardless of this switch.
     This is due to a limitation of Noise/ProTracker module format.

* Pattern Cutting
     When a pattern is breaked with the Pattern Break effect, it is cut
     to the line containing the break. When saving to a format that

does not support variable length patterns (eg.:
           SoundTracker
           ), the
pattern is grown again, so this optimization won't help with such
formats.

* Unused instruments
    A composer may decide to use an instrument and then change his
    mind, but forget to clear the instrument. In this case the module
    will contain an instrument that is never played. These instruments
    are simply stripped away.

* Sample data after a loop
    The part of an instrument following a loop is never played and is
    therefore discarded.

* Instruments zero tails
    Long zero tails eat up memory and produce no sound. This
    optimization is performed only on loopless instruments, as a zero
    sequence inside a loop does make difference. XModule leaves at
    least two zero bytes to avoid the nasty click produced by the
    speaker dropping its volume to zero too quickly.

1.27 XModule.guide/ARexx Interface

ARexx Interface
****

   Sorry, sorry! XModule's documentation is still incomplete. The
next release will hopefully be more accurate (it's already three
releases I'm promising it :-).

   Anyway, XModule's ARexx port is a little brain dead at the moment,
so you're not missing a great deal.

1.28 XModule.guide/Known Bugs

              Known Bugs
****

   Murphy's laws state that nothing can be perfect, not even computer
programs. XModule is far from being perfect and I need your help to
fix as many bugs as possible. If you discover a bug which isn't
reported in the list below (note: there is no list at the moment; this
is a beta release, remember?), please notify me.

   When you report a bug, state exactly what happens and indicate how
to reproduce the conditions that generated the problem. Reports like
'XModule sometimes crashes converting my favorite module' does not help

at all. Include your name if you want to be credited in future XModule
releases.

    If the program refuses to convert a module, mail it to me on a disk
or upload it to a BBS that I can reach and I'll try to fix the problem
as soon as possible. Glass Globe BBS, whose address can be found in
the
                  Author Info
                   chapter, will do. If you want the disk back with the
fixed version of XModule, send me a self-addressed envelope with enough
italian stamps or enough money for a reply.

    XModule has been tested on several machines running Kickstart 2.0,
3.0 and 3.1. It should work on any 68K family processor (was tested on
68000, 68020 and 68040) and is Enforcer and Mungwall proof. I also
tested XModule with virtual memory programs such as VMM40 and found that
everything seems to work correctly.

1.29 XModule.guide/Future Plans

Future plans
****

* TeX
    Another format I'd like to add is AmigaTeX. If you know TeX,
    you'll surely be wondering what a typesetting program can be used
    for in music. Well, AmigaTeX is at the moment is the best
    implementation available on any computer of the TeX typesetting
    environment, and its distribution includes MusicTeX, a powerful
    music language which can output notes in a great standard staff
    notation. I know no other program on the Amiga that can print
    notes of such fine quality (if you do, let me know!), so I think
    that many musicians would appreciate such an option.

* Other module formats
    Some formats I'd like to give support for in XModule as soon as
    possible:

        - FastTracker II (.XM)

        - ScreamTracker (.S3M)

        - MultiTracker (.MTM)

        - OctaMED MMD1 and MMD2

        - Face The Music (.FTM)

        - Quadra Composer

    If you have documentation regarding a format you would like to see
    in the next release, please send it to me together with a couple
    of modules and, if possible and legal, the program that created

them. I'll try to implement the new format only if it is somewhat
easy: it is very hard to convert, say, Future Composer to
SoundTracker. Please, do not send a module without at least a
text file describing its structure, as I'm not willing to spend my
nights hex-dumping modules :-).

* External Loaders/Savers
    Add support for 3rd party loaders and savers in XModule.
    Currently, I'm thinking about the best implementation of the
    external module interface.

* Full Module Editor
    I'd like to turn XModule into a fully featured editing environment
    for any kind of module, but this would require an enormous amount
    of effort, and time is never enough...

1.30 XModule.guide/Programmers

                    Programmer notes
****

   XModule has been written in C and compiled with SAS C 6.51, Other
compilers may require some changes in the source code.

   The program is written in respect of the Amiga multitasking
operative system and complies with Commodore programming guidelines.
All allocated resources will be (hopefully) freed before program
termination. I wrote XModule with the goal in mind of keeping the
executable fast and small.

   This manual has been typeset using MakeInfo, a GNU PD utility ported
to the Amiga by Reinhard Spisser and Sebastiano Vigna. I found that
the time I saved writing one file for three was nearly compensated by
the time spent learning how to operate MakeInfo.

   Besides, English is not my first language (as you might have
guessed) and I bet this document is full of grammar mistakes.

   If you intend to translate, improve or somewhat edit this document,
please work on the supplied TexInfo file and then process it with the
utilities supplied in the MakeGuide distribution. You can find
MakeGuide in several PD libraries such as Aminet.

   Do not hesitate sending criticisms to my work, and, even better,
advices on how to improve the program and its documentation. See

                    Author Info
                        .

1.31   XModule.guide/XModule Format

```
XModule Module Format
*********************
```

   Before going into the details of the XModule module format, I'd like
to explain why I felt the need to invent yet another module format.
There were plenty of them, but none had the flexibility I was looking
for. The only one exception is in my opinion OctaMED's MMD. But the
MMD file format is very complicated, hard to read and poorly
documented. When I designed it, I wanted XMOD to be as flexible, easy,
efficient and expandable as possible. If you are developing a music
program, please consider supporting XMOD. If you have troubles with
the following format description, contact me and I'll be glad to help
you.

   XModule modules are standard IFF files. They consist of a sequence
of chunks (also called proprieties) each storing a particular type of
information related to the module. Text chunks, such as NAME and AUTH
chunks are variable sized and NOT null terminated. When you read such
a chunk, you should take care to clip the string to your maximum buffer
size. NAME and AUTH fields are optional and should be assumed empty
when missing.

   As with every IFF file, you should not depend on the order and size
of any chunk. However, as usual with most IFF formats, header chunks
must always precede related BODY chunks. XModule requires this for
PHDR chunks because it isn't possible to load a pattern without knowing
its length and number of tracks. Some chunks may grow in size in
future versions of XModule.

```
     FORM XMOD

       [NAME]
         The name chunk contains the original name used to save the module.

       [ANNO]
         XModule stores its copyright information here.

       MHDR
         Module header.  Contains a ModuleHeader structure as defined in
         <XModuleClass.h>.

       FORM SONG
         The song FORM contains all data relative to one song.  Any
         number of songs can be stored into one module.

         [NAME]
           Name of the song.

         [AUTH]
           Author of the song.

         SHDR
           Song Header.  Contains a SongHeader structure as defined in
           <XModuleClass.h>.
```

```
SEQN
  Song Sequence.  The contents of this chunk are unsigned bytes
  representing the pattern to play at a particular position.
  The size of this chunk should match the Length field of the
  SongHeader.
  When a song contains multiple sections, one SEQN chunk will
  be stored for each section (not implemented).

FORM PATT
  Each occurrence of this FORM contains one pattern.  Patterns
  are numbered starting from 0.  Each pattern has its own
  length and number of tracks.

  [NAME]
    Name of this Pattern.

  PHDR
    Pattern Header.  Contains a PatternHeader structure as
    defined in <XModuleClass.h>.

  BODY
    The BODY chunk is stored as an array of PatternNote
    structures.  Tracks are stored one after the other.
    There are NumTracks*Length Note structures in the BODY
    chunk.

END FORM PATT

FORM 8SVX
  Each SONG contains one or more occurences of this FORM,
  one for each sample.  XModule currently saves and loads
  only the NAME, VHDR and BODY chunks.  Be prepared to
  decode Fibonacci Delta Encoded BODY chunks. An
  additional non-standard chunk is saved:

  INST
    XModule instrument info.  Contains an InstrumentInfo
    structure  as defined in <XModuleClass.h>.

  When the BODY chunk is missing, the sample must be
  loaded from the instrument library using the contents
  of the NAME chunk as the file name.

  See the IFF 8SVX documentation for more details.

END FORM 8SVX

END FORM SONG

END FORM XMOD
```

1.32 XModule.guide/Credits

```
Credits
*******
```

* Alastair Booker, for suggestions and help with the FastTracker
  format.

  It is not your fault if Intel systems are so unstable! :-)))

* Fabio Barzagli, for beta testing XModule and for giving me all his
  awesome modules!

  Keep on making good music, Fabio!

* Massimo Brogioni, for being my boss and for spreading all those
  bogus XModule releases all around the world.

  Hey boss, have you got a spare 5Gig hard disk?

* Alessandro Lombardi, for 2.x beta testing.

  Stop deleting files!

* Erno Tuomainen, for his wonderful finnish BBS, for his kindness
  and for all the support he gave me.

  Are there any more top-rated modules to leech?

* Pauli Porkka, for his kindness, for letting me win a free DASMP
  key, for the sources and for all the e-mails.

  Will you support my format in the next DASMP release?

* Woody (Nicola Ferruzzi), for S3M and XM format descriptions.

  I'll send you a postcard from Japan as soon as I go there. :-)

* The Amiga, for being the best computer in the world. :^)

  Commodore: You've put really a lot of efford trying to kill the
  Amiga, but you won't succeed!

1.33 XModule.guide/Author Info

```
How to contact the author
*************************
```

   If you want to make me aware of your suggestions, bug reports, ideas
or you want to send me a gift, a good module or, why not, some money
:-), reach me in any of the following ways:

* Mail

          Bernardo Innocenti
          Via Ventiquattro Maggio, 14

```
        50129 - Firenze
        ITALY

* FAX

        +39-55-8877771
        Attn:  Bernardo Innocenti
        Subj:  Amiga/XModule

* E-Mail

        FidoNet:  Bernardo Innocenti 2:332/118.4
        AmigaNet: Bernardo Innocenti 39:102/201.4
        UseNet:   bernie@shock.nervous.com
```

   I will try to reply to all my E-Mail, but I'm too lazy to reply
old-style mail. Your suggestions will be taken into account unless
they are 'I want XModule to do ice tea'-style.

1.34 XModule.guide/Concept Index

```
            Concept Index
*************
```

UNIC-Tracker