Getty

COLLABORATORS

	TITLE : Getty	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Getty

1.1 Getty

```
                    GETTY V1.3
        written 1996 by Michael Schettler


                Introduction
                 What is this thing, anyway?


                Copyright
                 What to do and what not to do


                Requirements
                 Ahhh, let's see


                Getting started
                 Ok, let's do it!


                Commands
                 Oh oh, i knew it, it's very complex :(


                Options
                 What, more things to change ....


                Switches
                 Ahhhhhhhhh, still more things ...


                Technical infos
                 Some tips and tricks


                Getting registered
```

1.2 Introduction

Introduction
============

Getty is a CLI only tool to monitor the serial port. It basicly does the same thing as Matt Dillons Getty, but in a more elegant, amiga like, way.

My implementation of the Getty is as following:

* The first Getty is started as a server, the following Getty's are started as independant client processes of the master.

  The only thing that was really annoying in Matt's Getty was that if you had e.g. five serial lines, you had to start five Getty's ---> you had to have the memory for five Getty's, because Getty was loaded five times into memory, although it was the same code.

* I added a more comfortable commandline parsing, where much more actions can be taken (see
  commands
).

* Several actions can be taken, if a certain user is logging in.

  For example

  - an external programm can be started, like a Mailbox or something similar

  - a remote shell can be opened, where the user can access your computer

The access the user has can be limited (e.g. forbid access to certain paths or commands) using a seperate
access file

```
                  If  you  have  the  MultiUser  filesystem  by  Geert  ↩
                     Uytterhoeven
  installed, the access control is even better!

      - all the actions the user is taking can be  echoed  to  a  seperate
  snoop file

  * Everything is definable (e.g. you can
                define
                 the command to open a new
    shell or you can define the behavior of the serial line ....)

  * It's programmed in 100% assembler, resulting in highly optimized code

This is a shareware version which means that to get  all  things  working,
you have to
                register
                .
```

1.3 Copyright"

```
COPYRIGHT
=========

This software is copyrighted by Michael Schettler. That means that you are
NOT  ALLOWED  to  modify  the  program(s)  and  documentation  in any way.
Especially you MUST NOT REMOVE the  documentation  or  any  supplied  text
file.

You are NOT allowed to use this software or any part of it in an  comercial
way.  This  also includes any fonts, images or samples. You are NOT allowed
to decompile or resource any part of it.


DISTRIBUTION
============

This package is  freely  distributable.  That  means  you  are  allowed  to
redistribute this package as long as you follow these points:

a. Any re-distribution has to include all files in this archive, including
   this "COPYRIGHT" notice, without any modifications. You are NOT allowed
   to add any files to the archive.

b. This package may be freely distributed via  BBSs, InterNet/UseNet, soft-
   ware libraries such as Fred Fish's and Aminet  CD-ROM, and other similar
   electronic channels.

c. Disk magazines and services that charge extra for file transfers may NOT
   distribute it without written permission by the author.


DISCLAIMER
==========
```

By using this product, you accept the FULL responsibility for any damage or loss that might occur through its use or the inability to use it. The author of the software can NOT be held responsible.

RETURN SERVICE
==============

This software is Sharware, that means if you use this software you have to register as a user.

"Free distributable" only says that you do not have to pay for copying or redistributing the software. You are allowed to test this product for 30 days. If you like it and decide to use the product regularly, please register.

Remember: A program worth using is a program worth buying!

1.4 Requirements

Requirements
============

Well, what you want me to say ....

- needs Kickstart 2.04 or upwards to run

- needs FIFO.library
  (C) by Matt Dillon

- MultiUser.Library would be good but isn't mandatory
  (C) by Geert Uytterhoeven

- OwnDevUnit.Library would be good but isn't mandatory either

- really needs a modem ;)

- needs 1 MB of your precious memory (naa, just kidding ...)

1.5 Things about the shareware version

Well, what you what me to say ..... life's a b#!@? and i have to make a livin'.

If you don't have a valid keyfile, following things are disabled in the shareware version:

* loading a config file
  - only one config file can be loaded. The supplied config settings are used (supplied in 'Getty.config')

```
  * making use of some access file features
    - restricting the access of the user. Only one path  and  one  command
      can be defined
    - it's only possible to execute a programm and to start a shell.  It's
      not possible to execute a programm using the shell

  * setting the timeout used for the remote shell stuff
    - changing the default timeout (3 minutes)
    - after the remote shell time is over, the user gets  kicked  off  the
      line

  * running more than one client (line)

If you think, that this programm is a good programm worth using, then  why
don't you consider to register?

The shareware fee is only 20 DM and i think that's enough for me  and  not
to much for you.

You can reach me at

      schettler@informatik.fh-wuerzburg.de

    which is my account at the university, where  i  study  computer
    science (more often checked)

      or at

      twd@incubus.franken.de

    which is my account at my favourite bbs (*less* often checked)

and mail me your request for getting registered. After that,  we  can  get
together and discuss some things.
```

1.6 Technical Information

```
                Technical information
====================

Here are some hints on getting the best out of Getty:

    - It's better to start the first Getty via the 'run' command to  avoid
      the locking of the shell you started it from.

      If you accidently started Getty without using 'run' you can kill  it
      by sending a CTRL-C signal to the cli process.


    - Supplying filenames you can specify  so  called
                meta-characters
                  to
      insert the actual setting

      Example of an
```

```
          access file
          :

    USER      test4
    SHELL     "BBS DEVICE %D UNIT %U BAUD %B SHARED USER %u"
    SNOOP     "RAM:%u.snoop"
```

If the client is running with the default settings, and the user 'test4' has successfully logged in, following line is sent to the FIFO Shell:

'BBS DEVICE serial.device UNIT 0 BAUD 19200 SHARED USER test4'

1.7 Thank you to all those people

The following people should be mentioned, because if they weren't, the whole project would have got more complex

Thank you to

- Matthew Dillon

  This guy has been and always will be a great 'gift' to the amiga community. Without him and without his great FIFO library this project would not have been possible.

- Geert Uytterhoeven

  For developing the MultiUser library. Great!

- Arne 'Illegal' Hinrichsen

  a computer freak who helped me testing and improving Getty.

- Marc 'Nepomuk' Heuler

  a friend who gave me some good hints ....

- Helge 'Camy' Prösch

  a friend who installed Getty to see how it screws up his computer and for still keep going on with testing :)

- the following people who have mailed me suggestions on improving Getty

  Jean-Marc Xiume

- and finally to my girlfriend Tanja for the patience she has/had

```
     while/during the development of this (great) tool

   and maybe to you for beta-testing ;)
```

1.8 Programm History

```
                Because Getty is shipped with a load of tools, here's a load of ↩
                   histories


                 Getty
                   The thrilling story of a programmers dream


                 Term
                   and how he managed to not going insane while


                 FTP
                   he tried to implement this great piece of


                 GUI
                   software ...
```

1.9 Programm History: Getty

```
                History
=======

 01.12.96   V1.3
 ~~~~~~~~~~~~~~~

   * Fixed some awefull memory bugs!

     When writing log messages i allocated memory  for  strings  *without*
     the terminating null byte, but i copied the null byte to the memory

     --> crashed on an A3000, because the pointer to the next memory chunk
   was overwritten, which meant a guru 800000b :(

     Since i'm using an A2000 and my memory structure is a  bit  different
     than the structure of an A3000, i've never noticed this bug.

     Now Getty should work on all systems (no more Enforcer  and  Mungwall
     hits, if you know what i mean).

     (thanks to Arne 'Illegal' Hinrichsen for testing on A3000!)

 ------------------------------------------------------------------------
```

19.11.96 V1.2a
~~~~~~~~~~~~~~~~

* Silly me! The config functions did'nt set the serial bauds. The default baudrate of 19200 was *always* used!
(thanks to Arne 'Illegal' Hinrichsen)

* Added more failure checking while doing fifo stuff.

18.11.96 V1.2
~~~~~~~~~~~~~~~

* Fixed and improved the TRAP handling

Hint to programmers: If memory gets freed, you should'nt access it anymore :)

* Added a new commandline argument

If you'd like Getty to lock the serial device, set the ODU=OWNDEVUNIT switch to YES.

02.11.96 V1.1
~~~~~~~~~~~~~~~

* Well, definitly removed the FIFO bug (see V1.07ß)

* Added a new commandline argument (thanks to Jean-Marc Xiume)

If you specify TRAP as argument, you tell Getty not to act as frontend (e.g. wait on some actions on the serial port). Instead Getty skips the waiting part and starts immediatly at the "Login:" prompt. After the user has logged in, Getty performs the defined actions and exits again.

Example: Use a programm similar to Getty to monitor the serial port (e.g AVM). If this frontend detects something on the port, it calls Getty to handle the actions. After Getty has managed the actions, it returns to the calling frontend.

It's also possible to connect two computers using a nullmodem cable and to start Getty on one computer (now who wants to do that?).

23.09.96 V1.0
~~~~~~~~~~~~~~~

* Some cleanups

* First Aminet release.

08.08.96 V1.08ß
~~~~~~~~~~~~~~~~~~

* Improved output of the info text. If Getty was started in CLI 1 and the user wanted some infos in CLI 2, the output was echoed to CLI 1 and not like wanted to CLI 2.

* Improved internal structures (again, grmpf!)

* Improved the commandline arguments PR and PG. They now have a number template instead of switch template which means that you have to specify a timeout value (see
PR
and
PG
for more details.)

* Added support functions for the GUI stuff.

26.06.96 V1.07ß
~~~~~~~~~~~~~~~~~~

* Removed some Enforcer hits :(

* Fixed some FIFO bugs (finally!)

- removed the 'hanging line' bug

- and finally all CTRL-? codes comming from the remote shell are passed thru to the programm running in the shell (that's what happens if you code in assembler and don't watch what value is in which register ;)

* Improved the .access file handling. First the local access file in the users home directory (defined in the .passwd file) is searched. If no file is found, the global access file (defined in the .config file) is searched for the user.

Example:

```
homedirectory of user    = Users:TestUser
global access file path  = Getty:Config/Getty.access
```

If TestUser logs in, Users:TestUser/.access is searched for the access settings of the user. If no entry is found, Getty.access is searched.

* Fixed a bug in the MultiUser functions which kept Getty hanging after the 3rd login of the same user.

```
13.06.96   V1.06ß
~~~~~~~~~~~~~~~~~~
```

  * Added the external port concept to get a link to a GettyTerm.

```
11.06.96   V1.05ß
~~~~~~~~~~~~~~~~~
```

  * Reworked the keyfile structure. Now multiple programmes can access one keyfile (Getty, GettyTerm, ...)

  * Done some minor cleanups and improvements

```
26.05.96   V1.04ß
~~~~~~~~~~~~~~~~~
```

  * Fixed a serious bug. Getty crashed if the serial device couldn't be loaded.

```
22.05.96   V1.03ß
~~~~~~~~~~~~~~~~~
```

  * Fixed a serious bug. Getty was only starting *one* line, e.g if you've started the first Getty on serial unit 0 and wanted to start a second Getty on serial unit 1, Getty tried to open the unit 0 again, what lead to a deadlock!

  * Totally rewrote the logfile functions. Now an independant process is started to handle the stuff. This results in a dramaticly speed increasement.

    Also if a logfile is already existing, the data will be appended to it.

  * Improved/changed/added new meta characters, which are now case sENsitIV! (oops ...)

```
       %D - Device name           e.g. 'serial.device'
       %U - Device unit                '0'
       %B - Device baud                '19200'
    CHANGED  %P - Port (Task) name          'GETTY-1'
    NEW      %T - Time                      'hh:mm:ss (Mon, 12.02.96)'
    CHANGED  %u - Name of user              'username'
    CHANGED  %h - Home of user              'USERS:username'
```

    And, finally, you can use meta characters in the log/snoop/output filename. (This time i didn't forget to add this feature! ;)

  * Done some minor cleanups and improvements

19.05.96 V1.02ß
~~~~~~~~~~~~~~~~

* Fixed a bug which was introduced due to changing the syntax of the access file. If mode EXEC was selected, Getty didn't output to snoop, it did it to NIL:!

* Added a function to pre-parse the command line (mode SHELL, EXEC) now you can say EXEC "run BBS DEVICE %D UNIT %U" which will be parsed to the correct device and unit settings

* Added a better error detection in accompaning files. If the access entry of a user isn't correct (e.g. the mode isn't specified), the user is informed to contact the sysop and he wont get any access to your computer.

* Done some minor cleanups

12.05.96 V1.01ß
~~~~~~~~~~~~~~~~

* Reworked the internal data structures. Now every structure has its own .i file and everything is much more comfortable to handle

* Renamed the MR=MODEMRESET command to MI=MODEMINIT

* Added a ME=MODEMEXIT argument. This command is sent to the modem after the process has received an 'EXIT' command.
(thanks to 'illegal')

* Added a CTRL-C routine to simulate a 'EXIT ALL' command
(thanks to Arne 'Illegal' Hinrichsen)

* Reworked the parsing of the access file resulting in a *NEW* syntax

* Added an action field to the INFO output. Possible actions are

- INIT the device is beeing initialized
- EXIT the device is beeing shut down
- WAIT getty is waiting on some action
- LOGIN a login is handled
- LOGINFAX a fax login is handled
- LOGINUSR a data login is handled
- EXEC an external programm is running
- SHELL a shell is beeing initialized
- SHELL-E an external programm is running using the shell
- SHELL-R a remote shell is active
- OWNDEV getty got a request from the OwnDevUnit library

08.05.96 V1.0ß
~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~

```
   * First release to beta testers
```

1.10 Programm History: Term

```
                Although the GUI functions are moved to Term,  there  is  a   ↩
                   seperate
                GUI
                history section.

History
=======

 01.12.96   V1.3ß
 ~~~~~~~~~~~~~~~~

   * Fixed some awefull memory bugs (see
                Getty V1.3
                 for details).

 ---------------------------------------------------------------------------

 28.11.96   V1.2ß
 ~~~~~~~~~~~~~~~~

   Internal release to beta testers

 ---------------------------------------------------------------------------

 25.11.96   V1.1ß
 ~~~~~~~~~~~~~~~~

   * Major change: moved the complete Getty-Gui interface into  Term.  Now
     it's possible to control Getty via Term menu entry
                Interface
                .

 ---------------------------------------------------------------------------

 18.11.96   V1.04ß
 ~~~~~~~~~~~~~~~~~

   * Minor cleanups

   * Added 'Serial Flags' to the settings menu.

 ---------------------------------------------------------------------------

 08.08.96   V1.03ß
 ~~~~~~~~~~~~~~~~~

   * Updated to use new structures

   * First Aminet release
```

```
 ------------------------------------------------------------------------

 19.06.96   V1.02ß
 ~~~~~~~~~~~~~~~~~

   * Added conversion functions for console and serial stuff

 ------------------------------------------------------------------------

 13.06.96   V1.01ß
 ~~~~~~~~~~~~~~~~~

   * Added the external port concept to get a link to a getty client.

   * Added keyfile handling. Now the user needs a valid Getty  keyfile  in
     order to run a *fully* featured Term.

 ------------------------------------------------------------------------

  11.06.96   V1.0ß
  ~~~~~~~~~~~~~~~~

   * first release to beta testers.

 ------------------------------------------------------------------------

  12.05.96   V0.9
  ~~~~~~~~~~~~~~~

   * started programming.
```

1.11 Programm History: FTP

```
History
=======


 01.12.96   V1.1ß
 ~~~~~~~~~~~~~~~~

   * Removed the StdIO  functions  and  replaced  them  with  real  serial
     functions  and  some  magic -> you wont see the stuff you typed while
     playing with FTP in the shell. Now FTP is more reliable!

 ------------------------------------------------------------------------

 13.08.96   V1.0ß
 ~~~~~~~~~~~~~~~~

   * First Aminet release. Although the up/downloading of files is still a
     bit buggy.

 ------------------------------------------------------------------------
```

1.12 Programm History: GUI

```
                Remember, since version 1.1ß the GUI functions are moved to
                Term
                !

History
=======

 01.12.96   V1.2ß
 ~~~~~~~~~~~~~~~~

   * Improved the
                interface
                 a bit.

     Added a history listview to show the messages of the  server.  Almost
     95 percent of the functions are implemented now.

 ------------------------------------------------------------------------

 25.11.96   V1.1ß
 ~~~~~~~~~~~~~~~~

   * Moved all functions to Getty-Term. The concept  of  an  external  GUI
     programm has been dropped (see Tools/Term/Getty menu/Interface link " ↩
         TOOLS_TERM_M4_1"}).

 ------------------------------------------------------------------------

 05.08.96   V1.0ß
 ~~~~~~~~~~~~~~~~

   * First Aminet release (although the  version  should  have  been  0.1ß
     because only the update functions for the "what's happening" listview
     are implemented!)

 ------------------------------------------------------------------------
```

1.13 Example usage

```
                I've tried to make Getty as versatile as possible. Here are some   ↩
                   examples
of how you can use Getty:


                 As frontend for a BBS or a remote shell

                 As backend for an other serial frontend
```

1.14 Example usage as frontend

```
     Under construction!
```

1.15 Example usage as backend

```
     Under construction!
```

1.16 Accompaning Tools

```
                Accompaning Tools
================

There a several tools in the Tools  directory  to  make  it  easier  using
Getty.


                Crypt
                 a tool to encrypt a password

                Chat
                 a simple chat programm

                Ftp
                 a tool to mark and transfer files

                More
                 a text viewer

                Term
                 a fully featured ANSI aware terminal with Getty GUI

                Transfer
                 a SZ/RZ like programm, if you know what i mean ;)
```

1.17 Accompaning Tools: CRYPT

```
                This tool is used to encrypt a password.

Syntax

     CRYPT <USERNAME>

where

     USERNAME   is the name of the user.

You can use this tool to get the encrypted password in case you don't have
```

```
MultiUser.library installed.

After invoking the progamm you get prompted for a password. After entering
a  password  string  you  have  to  enter  the  string again to verify the
password. After that, the programm prints the  encrypted  password,  which
has to be put in the
                .passwd
                 file.
```

1.18 Accompaning Tools: CHAT

```
This tool is used to chat with the system operator.

Syntax

     CHAT [USERNAME]

where

     USERNAME   is the name of the user who wants to chat.


A window will be opened on the system console which consists of the remote
part (upper half) and the local part (lower half).

You may clear the input field by pressing  CTRL-A  or  exit  the  chat  by
pressing CTRL-C.
```

1.19 Accompaning Tools: FTP

```
This tool is used to navigate around a file area and to select  files  for
downloading.

   ************************************************************
   *******               Still under development             *******
   *******  functions marked with !! aren't implemented yet  *******
   *******   or some functions may still contain some bugs   *******
   ************************************************************

Syntax

     FTP P=PORT/K,HELP/K/S

where

     P=PORT     is the name of the client port, where FTP is running on
    (e.g. "GETTY-1")


Following commands are available:
```

```
     N                  - move to the next page of the actual directory

     P                  - move to the previous page of the actual directory

     M  [NAME | NUM]    - add/remove a file to/from the  download  list.  A
       '*'  as  argument  will  clear the list and a '?'
       will toggle the display of the directory/download
       list

 !!  V  [NAME | NUM]    - view the contents of a file

     CD [NAME | NUM]    - change the actual directory. A '/' and '..' means
       the   previous  directory,  '~'  means  the  home
       directory of the user

 !!  DL [NAME | NUM]    - download a file immediatly. If no  arguments  are
       given, all marked files are downloaded

 !!  UL                 - upload a file to the user's home directory

     H                  - show this text

     X                  - exit the programm
```

If you just enter a number, the list will be refreshed starting with the entry at the specified position.

1.20 Accompaning Tools: MORE

This tool is used to display the contents of a text file.

Syntax

```
     MORE {FILENAME}
```

where

```
     FILENAME   is the name of the file to display
```

1.21 Accompaning Tools: TERM

This is a fully featured ANSI aware terminal programm with up-/ ↩
downloading
and link-to-server-and-see-what's-happening capabilities.

```
   ************************************************************
   *******               Still under development             *******
   *******  functions marked with !! aren't implemented yet  *******
   *******   or some functions may still contain some bugs   *******
   ************************************************************
```

```
Syntax

     TERM


Menu structure

     Project

                 About

                 New Terminal

                 Zip Terminal

                 Exit Terminal
                      Settings

                 Serial active

                 Serial Device

                 Serial Unit

                 Serial Baud

                 Serial Flags
                     Environment
 !!
                 Convert Table
                  !!
                 Convert Ser->Ser
                  !!
                 Convert Ser->Con

                 Hardware Handshake

                 Local Echo

                 Sizebar

                 Terminal Font

                 Terminal Width

                 Terminal Height

                 Open Settings

                 Save Settings
                     Save Settings as
     Transfer

                 Upload File

                 Download File
```

```
Upload Directory

Download Directory

XPR Library
 !!
XPR Options
     Getty

Interface

Install Link

Remove Link
```

1.22 Accompaning Tools: TERM - Project Menu

```
Menu: Project
Item: About

      Display the about message (aha ;)
```

1.23 Accompaning Tools: TERM - Project Menu

```
Menu: Project
Item: New Terminal

      Opens a new terminal window. The terminal is opened using the actual
      settings. Each terminal can have different settings.

      *** DISABLED IN SHAREWARE VERSION ***
```

1.24 Accompaning Tools: TERM - Project Menu

```
Menu: Project
Item: Zip Terminal

      Shrinks terminal window to minimal size.
```

1.25 Accompaning Tools: TERM - Project Menu

```
Menu: Project
Item: Exit Terminal

      Well, closes the actual terminal window. If all terminals  (windows)
      are closed, the programm returns to the cli.
```

1.26 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Serial Active

      If this item is checked, the terminal is activated.
```

1.27 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Serial Device

      Set the name of the serial device.
```

1.28 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Serial Unit

      Set the serial unit of the terminal.
```

1.29 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Serial Baud

      Set the baudrate of the terminal.

      *** DISABLED IN SHAREWARE VERSION ***
```

1.30 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Serial Flags

      Set the serial flags of the terminal.
```

1.31 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Convert Table

      NOT IMPLEMENTED
```

1.32 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Convert Ser->Con

      NOT IMPLEMENTED
```

1.33 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Convert Con->Ser

      NOT IMPLEMENTED
```

1.34 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Hardware Handshake

      Tell the serial device to use serial handshaking.
```

1.35 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Local Echo

      Echo all chars written to the serial device to the window.
```

1.36 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Sizebar

      Put a sizebar to the actual window.
```

1.37 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Terminal Font

      Select the font to use in the window.
```

1.38 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Terminal Width

      Set the width of the terminal window, specified in characters.
```

1.39 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Terminal Height

      Set the height of the terminal window, specified in characters.
```

1.40 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Open Settings

      Read the settings from disk.

      DISABLED IN SHAREWARE VERSION
```

1.41 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Save Settings

      Save the settings to disk.

      DISABLED IN SHAREWARE VERSION, WILL DESTROY SETTINGS FILE IF USED!
```

1.42 Accompaning Tools: TERM - Settings Menu

```
Menu: Settings
Item: Save Settings as

      Save the settings to disk.

      DISABLED IN SHAREWARE VERSION, WILL DESTROY SETTINGS FILE IF USED!
```

1.43 Accompaning Tools: TERM - Transfer Menu

```
Menu: Transfer
Item: Upload File

      Upload (send) a file using the specified XPR library.

      *** DISABLED IN SHAREWARE VERSION ***
```

1.44 Accompaning Tools: TERM - Transfer Menu

```
Menu: Transfer
Item: Download File

      Download (receive) a file using the specified XPR library.

      *** DISABLED IN SHAREWARE VERSION ***
```

1.45 Accompaning Tools: TERM - Transfer Menu

```
Menu: Transfer
Item: Upload Directory

      Set the upload directory.
```

1.46 Accompaning Tools: TERM - Transfer Menu

```
Menu: Transfer
Item: Download Directory

      Set the download directory.
```

1.47 Accompaning Tools: TERM - Transfer Menu

```
Menu: Transfer
Item: XPR Library

      Set the XPR library used for sending/receiving files.
```

1.48 Accompaning Tools: TERM - Transfer Menu

```
Menu: Transfer
Item: XPR Options

      NOT IMPLEMENTED
```

1.49 Accompaning Tools: TERM - Getty Menu

```
                Menu: Getty
Item: Interface (Getty GUI functions)

      Opens a window where you can snoop/control a running Getty server.

      On the left side of  the  window  there's  a  listview  showing  all
      running clients with some infos:

    CLIENT   the name of a running client

    USER     the name of a user currently logged into the client (as
       in the
                .passwd
                 file defined)

    ACTION   what the client is currently doing

       INIT        the client is initializing
       EXIT        the client is terminating
       WAIT        the client is waiting on some action
       LOGIN       the client has detected serial data
       LOGINFAX    the client has detected a fax connect
       LOGINUSR    the client has detected a data connect
       EXEC        the client executes a programm *without*  a
             shell
       SHELL       the client executes  a  programm  *with*  a
             shell
       SHELL-E     the client handles  a  remote  shell  *and*
             executes a programm in it
       SHELL-R     the client *just* handles a remote shell
       OWNDEV      some other programm has  control  over  the
             serial device

    TIME     the remaining time the user has online.  If  the  users
       justs   sits  there  doing  nothing,  a  shell  timeout
       (doing-nothing-timeout, see
                TIMEOUTSHELL
                 option)   may
       occur  before  a  time-is-up  timeout (see
                access file
                ,
       command TIME) occurs.


      On the right side of  the  window  there  are  several  gadgets  for
      controlling a selected client.

    INIT     initialize a new client (see
                Commands/Init
                )

    EXIT     exits a running client (see
                Commands/Exit
                )
```

```
TRAP     initialize a new client in trap mode (see
            Commands/Trap
            )

SHOW     show the settings of the client (see
            Commands/Show
            )

ABORT    abort a running client (see
            Commands/Abort
            )

SNOOP    ** UNTIL NOW, NOT IMPLEMENTED **

PREFS    ** UNTIL NOW, NOT IMPLEMENTED **

INFO     shows client infos (see
            Commands/Info
            )
```

After you have selected a command, the command template is put in the COMMAND gadget. Now you have to select a client to send the command to from the CLIENT listview. After that, press SEND and the command will be transmitted to the server.

The results of the action is displayed in the HISTORY listview.

On the bottom of the window there are three gadgets:

```
SEND     sends the command given in the COMMAND gadget to the
   server

CLEAR    clears the contents of the COMMAND gadget

COMMAND  a valid client command
```

and also a HISTORY listview to show the last messages from the server. The text is shown in a kind of compressed form, where all unneeded linefeeds are removed. You should'nt have too much trouble reading and understanding the output ;)

1.50 Accompaning Tools: TERM - Getty Menu

```
Menu: Getty
Item: Install Link
```

*** DISABLED IN SHAREWARE VERSION ***

Installs a link to a running getty client. After installing the link, all output of the client will also be sent to the linked terminal.

```
Example:    1. Start Getty.
2. Start the terminal programm  (make  sure  the  device
   settings are the same as the ones Getty uses!)
3. Select  'New Terminal' from the project menu.
4. Now move to the menu of the second  window  (the  one
   which  just has opened) and select 'Install Link' and
   choose 'GETTY-1' from  the  listview  (or  any  other
   Getty will do)
5. After that, move to the first window again and  do  a
   connect  (if  you  are  using nullmodem.device simply
   type 'ATD')
6. After logging in, see what happens ...
```

1.51 Accompaning Tools: TERM - Getty Menu

```
Menu: Getty
Item: Remove Link

      Removes a link to a Getty client.
```

1.52 Accompaning Tools: TRANSFER

```
This tool is used to receive/transfer files using the XPR standard.

   ************************************************************
   *******               Still under development             *******
   *******  functions marked with !! aren't implemented yet  *******
   *******   or some functions may still contain some bugs   *******
   ************************************************************

Syntax

     TRANSFER S=SEND/K/S,R=RECEIVE/K/S,F=FILE/K,D=DEVICE/K,U=UNIT/K/N,
        B=BAUD/K/N,L=LIBRARY/K,GUI/K/S,HELP/K/S

where

     S=SEND/K/S             - send a file

     R=RECEIVE/K/S          - receive a file

     F=FILE/K               - file to send/receive

     D=DEVICE/K             - specify the serial device name
            (default is 'serial.device')

     U=UNIT/K/N             - specify the serial device unit
            (default is '0')

     B=BAUD/K/N             - specify the serial device baud
            (default is '19200')
```

```
     L=LIBRARY/K            - specify the XPR Library
            (default is 'xprzmodem.library')

     GUI/K/S                - display a window where the actual infos  are
            displayed

     HELP/K/S               - show this text


The file transfer is implemented using the metaxpr.library.
```

1.53 Syntax of the command line

```
                Syntax of the command line
==========================

This is what you get, when you invoke Getty with a '?'

HELP/K/S,INIT/K/S,EXIT/K,TRAP/K/S,INFO/K/S,SHOW/K,ABORT/K,UPDATE/K,
D=DEVICE/K,U=UNIT/K/N,B=BAUD/K/N,F=FLAGS/K/N,M=MODE/K/N,K=KEYFILE/K,
C=CFGFILE/K,P=PWDFILE/K,A=ACCFILE/K,PR=PATCHREQS/K/S,PG=PATCHGURU/K/S,
LF=LOGFILE/K,SF=SNOOPFILE/K,HF=HEADERFILE/K,SC=SHELLCOMMAND/K,
SI=SHELLINIT/K,LL=LOGLEVEL/K/N,RL=RETRIESLOGIN/K/N,TL=TIMEOUTLOGIN/K/N,
TS=TIMEOUTSHELL/K/N,MI=MODEMINIT/K,ME=MODEMEXIT/K,MC=MODEMCOMMAND/K,
PE=PASSWDENCRYPT/K,BA=BAUDADJUST/K,ODU=OWNDEVUNIT/K,USW=USE7WIRE/K,
ICD=IGNORECD/K,IDTR=IGNOREDTR/K,ICON=IGNORECONNECT/K,ALL/K/S,QUIET/K/S

Wow ...... But now, lets explain this template a little bit.


Getty expects the following syntax:


                [run]
                 Getty <command> <option>

where
                command
                 means the thing what you want Getty to do and
                option
                  means
further data.

See
                technical information!
```

1.54 Command-Template: Commands

```
                Commands
========
```

```
These are the  commands  to  tell  Getty  what  to  do.  Please  read  the
explanations  in the order they are given! Also read the section about the

                special commands
                , which can only be given at initial startup.



                 HELP
                  shows a quickhelp page


                 INIT
                  starts a new Getty client


                 EXIT
                  exits a running Getty client


                 TRAP
                  traps a running serial connection


                 INFO
                  lists all running clients


                 SHOW
                  shows the settings of a specified client


                 ABORT
                  aborts the current action of the specified client


                 UPDATE
                  updates the settings of an *already* runnig client
```

1.55 Command-Template: Command HELP

```
HELP
----

  Shows a quick help reference.

  Example:

     SHELL> getty help

     GETTY 1.0 (13.5.96) by Michael Schettler
     Status: Command overview ...
```

```
   GETTY Commands:
INIT/K/S               - Start a new GETTY process (default)
       If DEVICE and UNIT isn't given, the
       default settings are used.
EXIT/K                 - Exits a running GETTY
       You have to specify the process name
       of the GETTY you want to be killed.
       If 'all' is specified, all processes will
       be removed!
TRAP/K/S               - Traps a running serial connection
INFO/K/S               - Report all running processes
SHOW/K/S               - Report settings of specified GETTY
HELP/K/S               - Show this text
ABORT/K                - Abort current action of specified GETTY
       If 'all' is specified, all processes will
       be aborted!
   *  QUIET/K/S             - No info text, please
   *  PR=PATCHREQS/K/S      - Patch AutoRequesters
   *  PG=PATCHGURUS/K/S     - Patch Gurus


   GETTY Parameter:
   +  D=DEVICE/K            - Specify the serial device name
       (default is 'serial.device')
   +  U=UNIT/K/N            - Specify the serial device unit
       (default is '0')
   +  B=BAUD/K/N            - Specify the serial device baud
       (default is '19200')
   +  F=FLAGS/K/N           - Specify the serial device flags
       (default is '176')
       (= RAD_BOOGIE+SHARED+XDISABLED)
   +  M=MODE/K/N            - Serial mode (0= fax disabled, 1= fax only with
       19200 baud, 2= heavydrop)
   +  K=KEYFILE/K           - Name of the keyfile
       (default is 'Getty:Config/Getty.key')
   +  P=PWDFILE/K            - Name of the passwordfile
       (default is 'Getty:Config/Getty.passwd')
   +  A=ACCFILE/K            - Name of the accessfile
       (default is 'Getty:Config/Getty.access')
C=CONFIG/K             - Specify the configuration name
       (default is 'Getty:Config/Getty.config')


   Config commands:
UPD=UPDATE/K           - Update the following config values
   +  LF=LOGFILE/K          - Name of the logfile
       (default is 'RAM:Getty.logfile')
   +  HF=HEADERFILE/K       - Name of the file displayed at login
       (default is 'Getty:Config/Getty.header')
   +  SF=SNOOPFILE/K        - Name of the file to echo the shell output to
       (default is 'RAM:Getty.snoop')
   +  SC=SHELLCOMMAND/K      - Command to execute if a FIFO shell is opened.
       (default is 'EXECUTE >NIL: S:Shell-Remote')
   +  LL=LOGLEVEL/K/N        - Logfile level
   +  RL=RETRIESLOGIN/K/N    - Retries for login
   +  TL=TIMEOUTLOGIN/K/N    - Timeout for login
   +  TS=TIMEOUTSHELL/K/N    - Timeout for shell
```

```
    +  MI=MODEMINIT/K          - String to init (reset) the modem
    +  ME=MODEMEXIT/K          - Command to send to the modem at exit
    +  MC=MODEMCOMMAND/K       - Command(s) to send to the modem after init
    +- BA=BAUDADJUST/K         - Adjust bauds to connect bauds
    *- ODU=OWNDEVUNIT/K        - Use the OwnDevUnit.library to lock the serial  ↩
       device
    +- USW=USE7WIRE/K          - Enable 7-wire line
    +- ICD=IGNORECD/K          - Ignore carrier detect
    +- IDTR=IGNOREDTR/K        - Ignore dataterminal ready
    +- ICON=IGNORECONNECT/K    - Ignore connect message

    The parameters listed above can also be changed using UPDATE

    NOTE: All entries marked with an '*' should  only  be  given  at  the
    initial startup of GETTY!
    All entries marked with an '+' can also be set via config.
    All entries marked with an '-' may be set via 'ON' or 'OFF'.
    The rest of the entries may be given via commandline.
```

1.56 Command-Template: Command INIT

```
                INIT
----

  Start a new Getty client.

  The idea behind the new concept is to start Getty the first  time  as  a
  server  and  then  automaticly  address  the  server if the programm was
  invoked again. The user wont notice if Getty is  running  in  server  or
  client mode, but your memory will.

  Starting Getty for the first time it will initialize  some  data  needed
  and then create the first client named 'GETTY-1'.

  Clients  are  started  as  DOS  processes,  which  means,  they  run  as
  independant tasks in your system communicating with the server.

  Each client creates a so called update port, which has the same name  as
  the  client.  Also  a  message  port for the device specific messages is
  created.

  Example:

     SHELL>
                [run]
                 getty init

     will start a new Getty. If  this  is  the  first  Getty  started,  it
     installs  itself  as  a  server  and  creates  the  first  client. If
     everything went ok, you will see the  following  output  (unless  you
     have specified
                QUIET
                )

   GETTY 1.0 (13.5.96) by Michael Schettler
```

```
   Status: Device 'serial.device', unit 0 initialized.

     Since we didn't give Getty any
               options
               ,  the
               default
                 values  where
     taken to init everything.

     Now if you start a Xoper like tool to snoop the system, you will  see
     something similar like this:

   process    GETTY                            (1)
   process    GETTY-1                          (2)
   port       GETTY                            (3)
   port       GETTY-1                          (4)
   port       GETTY-1-serial.device-0          (5)

   1) the server process
   2) the first client process
   3) the message port of the server
   4) the message port of the client
   5) the message port for the device specific stuff of the client

     Starting Getty again with different  parameters  it  simply  sends  a
     'create  client'  message with the
               supplied parameters
                to the already
     running server.

     Getty can be aborted anytime by simply sending a ctrl-c signal to the
     first  Getty  started.  Getty will act like it received an 'exit all'
     command and removes itself from the system.
```

1.57 Command-Template: Command EXIT

```
EXIT <clientname>
-----------------

  Exit a running Getty client.

  You have to specifiy the name of the client which is  to  be  terminated
  (e.g.  GETTY-1).  If  no  more  clients  are running, the main server is
  removed from the system.

  Example:

     SHELL> getty exit GETTY-1

     GETTY 1.0 (13.5.96) by Michael Schettler
     Status: Device 'serial.device', unit 0 exited.

  This  message  is  sent  from  the  client,  which  has   recently
  terminated
```

```
   GETTY 1.0 (13.5.96) by Michael Schettler
   Status: All processes exited.
```

And this message is from the server which has terminated.

```
   Specifying a client which isn't running will result in

   GETTY 1.0 (13.5.96) by Michael Schettler
   Status: Unable to find process!

   Specifying 'ALL' as client name will terminate all clients.
```

1.58 Command-Template: Command TRAP

```
TRAP
----
```

Specifying TRAP as an argument, you tell Getty not to act as frontend (e.g. not to wait on actions on the serial port). Instead Getty skips the waiting part and starts immediatly at the "Login:" prompt. After the user has logged in, Getty performs the defined actions and exits again.

Useful if you don't like using Getty as a frontend (Getty can't do everything .... there are other good programms around which do a better job on some things).

Here's a quick example how this part works:

> Use a programm similar to Getty to monitor the serial port. If this frontend detects something on the port, it calls Getty to handle the actions. After Getty has managed the actions, it returns to the calling frontend.

1.59 Command-Template: Command INFO

```
                INFO
----
```

List all running clients

If you want to know whats going on in your system (concerning Getty ;) this is a good method to do so.

After invoking the server with this command it prints a table containing all the running Getty clients listed by name, serial device/unit and actual user online.

Example:

```
SHELL> getty info

GETTY 1.0 (13.5.96) Michael Schettler
Status: Reporting running processes

Port        Device                  Unit    User         Action
-------------------------------------------------------------------
GETTY-1     serial.device           0       test         LOGINUSR
(1)         (2)                     (3)     (4)          (5)

Programm is running in
           demo mode
                (6)


(1) this is the name of the client

(2) the device name

(3) the device unit

(4) the user currently online or an empty text

(5) the thing the client is doing at the moment, e.g.

INIT       the device is beeing initialized
EXIT       the device is beeing shut down
WAIT       the client is waiting on some action
LOGIN      a login is handled
LOGINFAX   a fax login is handled
LOGINUSR   a user login is handled
EXEC       an external programm is running
SHELL      a shell is beeing initialized
SHELL-E    an external programm is running using the shell
SHELL-R    a remote shell is active
OWNDEV     the client got a request from the OwnDevUnit library

(6) the mode the client is running in.

 Remember: If you don't have a valid  keyfile, some  features  are
     disabled (see
              registration
              )
```

1.60 Command-Template: Command SHOW

```
               SHOW <clientname>
-----------------

  Shows the settings of a specified client.

  This one is a big one. The server displays the actual settings of the
  specified client.
```

```
  Example:

     SHELL> getty show getty-1

     GETTY 1.0 (13.5.96) Michael Schettler
     Status: Reporting settings of GETTY-1


                 DEVICE
                      = serial.device
                 UNIT
                  = 0,
                 BAUD
                  = 19200

                 FLAGS
                  = 176,
                 MODE
                  = 0


                 MODEMINIT
                   = ATZ
                 MODEMEXIT
                   = ATZ

                 MODEMCMD
                    = ATM1S0=1

                 MODEMCMD
                    = +FCLASS=0+FCR=1+FAA=1

KEYFILE    = Getty:Config/Getty.key

                 KEYUSER
                     =


                 CONFIG
                      = Getty:Config/Getty.config
                 HEADER
                     = Getty:Config/Getty.header

                 PASSWD
                      = Getty:Config/Getty.passwd
                 ACCESS
                     = Getty:Config/Getty.access

                 SNOOP
                       = Ram:Getty.snoop


                 LOGFILE
                     = Ram:Getty.log
                 LOGLEVEL
                   = 2
```

```
                SHELLCMD
                   = C:NEWSHELL

                SHELLINIT
                  = C:EXECUTE >NIL: S:Remote-Startup


                TIMEOUTLOGIN
                 = 30,
                TIMEOUTSHELL
                 = 3

PATCHREQS = off,
                BAUDADJUST
                 = off,
                IGNOREDTR
                 = off,
                IGNORECONNECT
                 = off
PATCHGURU = off,
                USE7WIRE
                   = off,
                IGNORECD
                  =  on,
                PASSWDENCRYPT
                 =  on
```

As you can see, all the settings of the specified client are reported.

1.61 Command-Template: Command ABORT

```
ABORT <clientname>
------------------
```

Abort the current action of the specified client.

This command *tries* to abort the client and *tries* to set him to the phone-line-wait-state

Particularly this means if someone is logging in, the line is dropped, the user is kicked out and Getty returns to waiting on a phone-ring.

Example:

```
SHELL> getty abort getty-1

GETTY 1.0 (13.5.96) by Michael Schettler
Status: Aborting action of GETTY-1
```

This is a message from the server

```
     GETTY 1.0 (13.5.96) by Michael Schettler
     Status: Aborted Device 'serial.device', unit 0.

  And this message comes from the client.

NOTE: Sending a ctrl-d signal to the process has the same effect.
```

1.62 Command-Template: Command UPDATE

```
                UPDATE <clientname>
-------------------

  Specifying this argument you tell the server to update the settings of
  an already running client.

  Simply specify the things you would like to have changed via the
                command
                  line.

  NOTE: This command is NOT used to reload the config file  of  a  client.
  It's used to modify the settings in the structure of the client.
```

1.63 Default settings

```
                These are the default settings Getty uses, if an argument isn't  ↩
                   specified:


                 D=DEVICE
                  = serial.device

                 U=UNIT
                  = 0

                 B=BAUD
                  = 19200

                 F=FLAGS
                  = 176

                 M=MODE
                  = 0

                 K=KEYFILE
                  = GETTY:Config/Getty.key

                 P=PWDFILE
                  = GETTY:Config/Getty.passwd

                 A=ACCFILE
                  = GETTY:Config/Getty.access
```

```
C=CFGFILE
 = GETTY:Config/Getty.config

LF=LOGFILE
 = RAM:Getty.log

LL=LOGLEVEL
 = 2

SF=SNOOPFILE
 = ""

HF=HEADERFILE
 = GETTY:Config/Getty.header

SC=SHELLCOMMAND
 = C:NewShell

SI=SHELLINIT
 = C:Execute >NIL: S:Remote-Startup

RL=RETRIESLOGIN
 = 3

TL=TIMEOUTLOGIN
 = 30

TS=TIMEOUTSHELL
 = 600

MI=MODEMINIT
 = ATZ

ME=MODEMEXIT
 = ATZ

MC=MODEMCOMMAND
 = ATM1S0=1
= +FCLASS=0+FCR=1+FAA=1


PE=PASSWDENCRYPT
 = OFF

BA=BAUDADJUST
 = OFF

ODU=OWNDEVUNIT
 = OFF

USW=USE7WIRE
 = OFF

ICD=IGNORECD
 = OFF
```

```
                IDTR=IGNOREDTR
                 = OFF

                ICON=IGNORECONNECT
                 = OFF
```

1.64 Parsing of meta characters in filenames

```
The meta characters can be used in filenames to insert the actual settings
the client is running with.

     possible meta characters        example output

     %D - Device name                'serial.device'
     %U - Device unit                '0'
     %B - Device baud                '19200'
     %P - Port (Task) name           'GETTY-1'
     %T - Time                       'hh:mm:ss (Mon, 12.02.96)'
     %u - Name of user               'username'
     %h - Home of user               'USERS:username'

Example:

     A filename like 'MailBox DEVICE %D UNIT %U USER %u' will be
     translated to 'MailBox DEVICE serial.device UNIT 0 USER username'
```

1.65 Command-Template: Options

```
                Options
=======

The options specify the settings and the
                modes
                 Getty should work with.
They can be set via the
                command line
                 or via
                config file
                .


                 D=DEVICE

                 U=UNIT

                 B=BAUD

                 F=FLAGS

                 M=MODE
```

```
K=KEYFILE

P=PWDFILE

A=ACCFILE

C=CFGFILE

LF=LOGFILE

LL=LOGLEVEL

SF=SNOOPFILE

HF=HEADERFILE

SC=SHELLCOMMAND

SI=SHELLINIT

RL=RETRIESLOGIN

TL=TIMEOUTLOGIN

TS=TIMEOUTSHELL

MI=MODEMINIT

ME=MODEMEXIT

MC=MODEMCOMMAND
```

1.66 Command-Template: Option DEVICE

```
D=DEVICE <serial.device>
-------------------------

  Set the serial device to be monitored. If no device  is  specified,  the
  default is used.

  Default: 'serial.device'
```

1.67 Command-Template: Option UNIT

```
U=UNIT <serial unit number>
---------------------------

  Set the unit of the serial device. If  no  unit  number  is  given,  the
  default will be used.

  Default is unit 0
```

1.68 Command-Template: Option BAUD

```
B=BAUD <bauds>
--------------

  Set the baud-rate of the serial device

  This value can range from 300 bauds upto the maximum  bauds  your  modem
  can handle.

  Default is 19200 bauds
```

1.69 Command-Template: Option FLAGS

```
F=FLAGS <flags value>
---------------------

  Sets the serial flags as specified in 'includes/devices/serial.i'

  Here is a quick overview of the definitions taken from the include file:

      Symbol             Bit  Value   Comment
      ------------------------------------------------------------
      SER_XDISABLED      7     128    xOn-xOff feature disabled bit
      SER_EOFMODE        6      64    EOF mode enabled bit
      SER_SHARED         5      32    non-exclusive access
      SER_RAD_BOOGIE     4      16    high-speed mode active
      SER_QUEUEDBRK      3       8    queue this Break ioRqst
      SER_7WIRE          2       4    RS232 7-wire protocol
      SER_PARTY_ODD      1       2    use-odd-parity bit
      SER_PARTY_ON       0       1    parity-enabled bit

  Default is 176 (which is 128+32+16)
```

1.70 Command-Template: Option MODE

```
M=MODE <value>
--------------

  Sets the mode Getty handles the incomming calls.

  following values are recognized

     0   means    fax handling is disabled
     1   means    fax handling only with 19200 baud
     2   means    to drop the serial line the hard way (close  the  device
      and open it again)

  Default is 0
```

1.71 Command-Template: Option KEYFILE

```
                K=KEYFILE <filename>
--------------------

  Ahhh, the most important setting. The name of the keyfile.

  Default is 'Getty:Config/Getty.key'

  (Remember to
                register
                !)
```

1.72 Command-Template: Option PWDFILE

```
                P=PWDFILE <filename>
--------------------

  Specifies the password file Getty should use.


  Syntax of the password file
  ---------------------------

     For each user, the password file must contain a line like this:

      <UserID>|<Password>|<UID>|<GID>|<UserName>|<HomeDir>|<Programm>

     with

      <UserID>      User Login ID                          (case sensitive!)
      <Password>    Encrypted password, or empty or "*"    (case sensitive!)
      <UID>         User number (1 - 65535)
      <GID>         Primary group number (0 - 65535)
      <UserName>    Full user name
      <HomeDir>     Home directory
      <Programm>    Programm to be executed

     Comment lines start with a ";" in the FIRST colum.


  Example of a password file (this one is my MultiUser passwd file):
  ------------------------------------------------------------------

     guest||1|0|Guest|T:|
     twd|[MnVI\KZtoa|4100|65535|Michael Schettler|dh0:|
     ROOT|_ggYCfiaMVd|65535|65535|System.Administrator|dh0:|
     helge|eDCZgTpoiYj|4097|2048|Helge Prösch|Users:helge|
     test|jSaEsHXtsm]|4099|2048|Test-User|t:|

     lets take the user test (see last line of password file) and  explain
     a little
```

```
    test|jSaEsHXtsm]|4099|2048|Test-User|RAM:|C:Info
    (1)  (2)          (3)  (4)  (5)       (6)  (7)

    (1) this is the name the user has to log in with
    (2) the encrypted password of the user (the password is 'test')
    (3) the user id (e.g. used by MultiUser)
    (4) the group id (e.g. used by MultiUser)
    (5) the user name
    (6) the directory the user is in after login
    (7) finally the programm which is executed

    NOTE: Getty only uses the first and second entry of  the  user  data,
    the  rest  is  ignored. This syntax is kept to be compatible to
    the MultiUser.Library, which means an entry like this

        test|jSaEsHXtsm]|||||
        (1)  (2)

    is ok. Getty gets it's vital user data from the
               .access file
               .

  Default is 'Getty:Config/Getty.passwd'
```

1.73 Command-Template: Option ACCFILE

```
               A=ACCFILE <filename>
--------------------

  Specifies the access file, Getty should use


  Syntax of the access file
  -------------------------

     For each user, the Access File must contain entries like this:

  USER     <UserID>
  MODE     "
               Filename Arguments
               "
  [TIME    <Timeout>]
  [SNOOP   "
               Filename
               "]
  [PATH    "<UserPath>"]
  [CMD     "<UserCommand>"]

     where arguments in [] can be left away


     Syntax:

  USER <UserID>      LoginID, *same* as specified  in  the  password
```

```
       file!

MODE               Can either be

       EXEC   <Filename>  <Filename> will be executed

       SHELL  <Filename>  If a FIFO  shell  should  be
              opened and <Filename> should
              be  executed.  Supplying  an
              empty string ("") will cause
              Getty  to  open   a   remote
              shell.

       LOCKED             If the user  has  no  access
              (not allowed to login)

TIME <Timeout>     The amount of seconds the user  has  access  to
       the  FIFO  shell  (only remote shell!). If this
       argument isn't  specified,  the  default  value
       of 180 is used.

SNOOP [Filename]   If the output should be echoed to a  snoop-file
       defined  in  the config, select an empty string
       ("") as filename.
       If  [Filename]  is  given,  the  global  config
       setting is overwritten.

PATH <Path>        Path the user has *NO* access to

CMD <Command>      Commands the user has *NO* access to


   IMPORTANT: STARTING A PROGRAMM IN EXEC MODE, GETTY HAS  *NO*  CONTROL
  OVER  WHAT'S  GOING ON. IF THE RUNNING PROGRAMM CRASHES OR
  THE SERIAL LINE IS DROPPED, GETTY HANGS.

  YOU HAVE NO WAY TO STOP GETTY IN SUCH SITUATIONS!

  I'll do my best to find a solution to  this  problem  ....
  although it's a bit tricky and might require some MAGIC :)


Example of a access file (this one is my access test-file):
------------------------------------------------------------

   USER      test1                ;the user 'test1' has no access
   LOCKED

   USER      test2                ;this user has access to the remote
   SHELL     ""                   ;shell but only for 3 minutes (default)
   SNOOP     ""                   ;no snooping is done.

   USER      test3                ;this user also has access to the
   SHELL     ""                   ;remote shell ....
   TIME      30                   ;but for only 30 seconds!
   SNOOP     "RAM:.%u.snoop"      ;the output will also be echoed to
   PATH      RAM:                 ;"RAM:test3.snoop". The user has no
```

```
    CMD       List                  ;access to this command and path

    USER      test4                 ;if this user logs in, the "Programm"
    SHELL     "Programm"            ;is executed enabling output to the
    TIME      30                    ;shell and to "ram:test4.snoop"
    SNOOP     "RAM:.%u.snoop"       ;TIME has *no* meaning here!

    USER      test5                 ;if this user logs in, the "Programm"
    EXEC      "Programm"            ;is executed. No output is displayed,
           ;the programm is simply executed.

    Also see the supplied access file for even more examples.

    NOTE1: Comments start with a ";"  (aha, i knew it ...)

    NOTE2: Keep in mind that each  userblock  *has*  to  start  with  the
     keyword  USER.  Please keep the order of the 'commands' in the
     access file as listed above!

    NOTE3: If no PATH or CMD entry is given,  the  programm  is  executed
     normally  and  the  user  has  *full*  access. (unless you are
     running MultiUser.Library)

    NOTE4: The filename can contain
               meta-characters
                which will  be  parse
     before the filename is used.
```

1.74 Command-Template: Option CFGFILE

```
               C=CFGFILE <filename>
-------------------

  Specify the config file


  Syntax of the config file
  -------------------------

     The config file can contain all the arguments  Getty  understands  in
     the
               command line
               .


  Example of a config file (this one is my config test-file):
  -----------------------------------------------------------


               keyfile
                       Getty:Config/Getty.keyfile

               pwdfile
                       Getty:Config/Getty.passwd
```

```
            accfile
                    Getty:Config/Getty.access

;-------------------------------------------------------------------
;the serial settings


            device
                     serial.device

            unit
                       0

            baud
                       19200

            flags
                      176

            mode
                       0

;-------------------------------------------------------------------


            logfile
                    ram:Getty.log              ;the logfile. each Getty

            loglevel
                   3                          ;can have it's own logfile
             ;or all Getty's share the
             ;same (see example)


            headerfile
                 Getty:Config/Getty.header ;this textfile is displayed
            ;if a user logs in


            snoopfile
                  ""                          ;normally you should supply
             ;a filename to write the
             ;snoop-stuff to, but if you
             ;specify it like this, no
             ;snooping is done


            shellcommand
               "C:NEWSHELL"               ;used for remote-shell

            shellinit
                  "C:EXECUTE >NIL: S:Remote-Startup"

;-------------------------------------------------------------------
```

```
            retrieslogin
               3

            timeoutlogin
               30

            timeoutshell
               360

;---------------------------------------------------------------

passwdencrypt  off

            owndevunit
                 on

            baudadjust
                 off

            use7wire
                   off

            ignorecd
                   on                          ;enable this if you are
             ;using an amiga 1000 (i do!)

            ignoredtr
                  off

            ignoreconnect
              off

;---------------------------------------------------------------


            modeminit
                 "ATZ"

            modemexit
                 "ATZ"

            modemcommand
              "ATM1S0=1"

            modemcommand
              "+FCLASS=0+FCR=1+FAA=1"
```

1.75 Command-Template: Option LOGFILE

```
            LF=LOGFILE <filename>
----------------------
```

Specify the output of the logging.

```
Example of a log file
---------------------

Each line documents a recently happened event in the form

   client   time      event
   ------------------------------------------------------------

   GETTY-1  21:57:57  ------ SESSION BEGIN (Wed, 01.05.1996) ------
a new client has started

   GETTY-1  21:57:57  Open serial device 'serial.device', unit 0
   GETTY-1  21:57:58  Device 'serial.device', unit 0 initialized.
   GETTY-1  21:57:58  Init Modem
the actions the client has taken until now. At  this  moment,  the
client is just sitting there and waits on something to happen.

   GETTY-1  21:58:25  Serial Data Detected
a phone ring has been detected. at this moment, the  client  picks
up  the  phone  and answers the ring (if 'RING' is caught from the
serial, else it checks, if a carrier is detected.)

   GETTY-1  21:58:30  Connect at 14400 baud
just some info for you, the sysop. The client displays the
              header
                text and prompt the user for a login.

   GETTY-1  21:58:38  Login, User='
              helge
              '
the user has successfully logged in.

   GETTY-1  21:58:39  Shell initialized for ''
according to the
              access file
              , the actions are taken.  A  ''  means
that no file is executetd but a remote shell is started (i know, i
have to work on this one)

Now the shell logging starts  were  *everything*  that  the  shell
produces is written to the
              log file
                   ------- Begin of Shell-Logging -----

   Neuer Shell-Prozeß 10

   Following commands are available:

 EXIT      - quit shell and logoff

   system:hdtools: Datei ist lesegeschützt     ** that's MultiUser :) **
   10> list
   .lastlogin                    28 ----rwed 02-Feb-96  01:10:48
   C                            Dir ----rwed 01-Jan-96  22:08:33
   S                            Dir ----rwed 01-Jan-96  22:08:33
   T                            Dir ----rwed 01-Jan-96  22:08:33
```

```
    Data                          Dir ----rwed 01-Jan-96  22:08:34
    Projects                      Dir ----rwed 01-Jan-96  22:08:34
    .profile                       74 ----rwed 01-Jan-96  22:08:34
    2 files - 5 directories - 14 blocks used
    10> exit
    Prozeß 10 endet

    -------- End of Shell-Logging ------

    GETTY-1  21:59:39  Shell returned.
  ok, the shell terminated.

    GETTY-1  21:59:39  Disconnecting
  well, then drop the line.

    GETTY-1  21:59:44  Init Modem
  now return to the start and wait, wait, wait .....

    GETTY-1  21:59:50  Device 'serial.device', unit 0 exited.
  ok, the sysop (me) decided to quit the game.

    GETTY-1  21:59:54  ------ SESSION END (Wed, 01.05.1996) ------
  end of the show, the client steps aside and returns.
```

1.76 Command-Template: Option LOGLEVEL

```
LL=LOGLEVEL <level>
-------------------

  This value specifies the level of the log messages.

  Specify

     0    if you want no log messages at all,
     1    you want error messages to be logged, and
     2    to log the things the user does if he/she is using the
    remote shell.

  Remember, the higher the value, the more info you will get.

  Default is 2
```

1.77 Command-Template: Option SNOOPFILE

```
                SF=SNOOPFILE <filename>
-----------------------

  Specify the name of the file to write the shell output  to  (see
                access
                  file, option SNOOP).
```

1.78 Command-Template: Option HEADERFILE

```
HF=HEADERFILE <filename>
------------------------
```

Supply the name of the file which is displayed if a data connection is accomplished.

Example of a header file

```
>>  Please identify!
>>
```

will show exactly the text marked with '>>' when a user is trying to do a login.

1.79 Command-Template: Option SHELLCOMMAND

```
                SC=SHELLCOMMAND <command>
-------------------------
```

Specify the command to start the remote shell (see access).

Default is 'C:NEWSHELL'

1.80 Command-Template: Option SHELLINIT

```
                SI=SHELLINIT <command>
-------------------------
```

Specify the command to initialize the remote shell (see access).

Default is 'C:EXECUTE >NIL: S:REMOTE-STARTUP'

1.81 Command-Template: Option RETRIESLOGIN

```
RL=RETRIESLOGIN <number>
------------------------
```

Set the number of retries, the user has during the login procedure.

Default is 3

1.82 Command-Template: Option TIMEOUTLOGIN

```
TL=TIMEOUTLOGIN <value>
----------------------
```

Set the number of seconds the user has to login.

Default is 30

1.83 Command-Template: Option TIMEOUTSHELL

```
TS=TIMEOUTSHELL <number>
------------------------
```

Specifies the time the user has to type something until he gets kicked out automaticly because nothing happened.

Default is 180 seconds.

1.84 Command-Template: Option MODEMINIT

```
MI=MODEMINIT <string>
---------------------
```

Set the string which is sent to initialize (reset) the modem

Default is 'ATZ'

1.85 Command-Template: Option MODEMEXIT

```
ME=MODEMEXIT <string>
---------------------
```

Set the string which is sent to the modem at exit

Default is 'ATZ'

1.86 Command-Template: Option MODEMCOMMAND

```
MC=MODEMCOMMAND <string>
------------------------
```

Set the string(s) which is/are sent to initialize the modem.

Defaults are 'ATM1S0=1' and '+FCLASS=0+FCR=1+FAA=1'

1.87 Command-Template: Switches

```
                Switches
========
```

The following switches simply modify the behavior and handling of some modem stuff. Specifying 'ON' enables them, where 'OFF' disables them.

PE=PASSWDENCRYPT

BA=BAUDADJUST

USW=USE7WIRE

ICD=IGNORECD

ODU=OWNDEVUNIT

IDTR=IGNOREDTR

ICON=IGNORECONNECT

1.88 Command-Template: Switch PASSWDENCRYPT

```
                PE=PASSWDENCRYPT <ON | OFF>
----------------------------
```

Specifies if Getty should encrypt the passwords given in the password file. Encryption is done via a MultiUser compatible algorithm.

Example of a valid password file entry:

```
   username|eDCZgTpoiYj|0|1|crypted password|T:|
      ^^^^^^^^^^^
      MultiUser crypted password


   username|password|0|1|crypted password|T:|
      ^^^^^^^^
      plain password entry, *not* crypted.
```

IMPORTANT: You have to specify if the passwords are crypted or not or your users will never get access to your computer.

See

```
        tools/CRYPT
          Default is off (*not* crypted!)
```

1.89 Command-Template: Switch BAUDADJUST

```
BA=BAUDADJUST <ON | OFF>
------------------------
```

Specifies if Getty should adapt to the bauds of the user.

Example:

Getty is running with 19200 bauds and a user is calling us with 9600 bauds.

If baudadjust is set to ON, Getty adapts to 9600 bauds.

1.90 Command-Template: Switch OWNDEVUNIT

```
ODU=OWNDEVUNIT <ON | OFF>
-------------------------
```

If this switch is set to ON Getty will lock the serial device using the OwnDevUnit.library.

NOTE: This option can only be specified at the initial startup of the server.

1.91 Command-Template: Switch USE7WIRE

```
USW=USE7WIRE <ON | OFF>
-----------------------
```

Tell the serial device that we are running with a 7-wire serial cable.

1.92 Command-Template: Switch IGNORECD

```
ICD=IGNORECD <ON | OFF>
-----------------------
```

Getty checks every second if the carrier is still there. If you don't mind that the caller has acciedently slipped from the chair and left the line still open, set this switch to 'OFF'

1.93 Command-Template: Switch IGNOREDTR

```
IDTR=IGNOREDTR <ON | OFF>
-------------------------
```

If Getty drops the line it check the DTR signal. If the signal isn't

there, it keeps looping more often. Selecting 'OFF' will force Getty to ignore the DTR signal.

1.94 Command-Template: Switch IGNORECONNECT

```
ICON=IGNORECONNECT <ON | OFF>
-----------------------------
```

Everytime the modem detects a connect it sends a connect message like 'CONNECT 19200' to the serial device. If you would like Getty to ignore this message, set this switch to 'ON'

1.95 Command-Template: Specialties

```
                Special Commands
================

These are the options which wont fit into the other categories ...


                 PR=PATCHREQS

                 PG=PATCHGURU

                 QUIET
                Remember: These options can only be specified at the initial ↩
                   startup of
    the server.
```

1.96 Command-Template: Specialty PATCHREQS

```
PR=PATCHREQS <number>
---------------------
```

Supplying this argument you tell the server to patch the EasyRequest() function call to automaticly do a 'Cancel' after the defined seconds on any requesters apearing while Getty is running.

Imagine following situation:

> You are not at home. Your best friend is calling to remotly access your computer. He has access to the remote shell and accidently types following line:
>
> 'list freddy:' instead of 'list freddy'.

Now, your computer is running wild because he wants to display the contents of the volume 'freddy', which isn't mounted to the system. 'OK' the computer thinks, 'lets tell the user i need a volume called freddy'.

He pops a requester like 'Please insert volume freddy:' and waits for the user to insert the requested volume. But the user (YOU) isn't at home. While the computer sits waiting and your friend sits waiting Getty pops in, DisplayBeep()'s the screen for about the defined seconds and 'hits' cancel on the requester. The computer is happy because he can continue to search for freddy, your friend is happy because your computer isn't hanging and life goes on ...

NOTE: This option is only available while starting Getty for the first time!

If you define a timeout of 0 seconds, the requester is canceled immediatly!

1.97 Command-Template: Specialty PATCHGURU

```
PG=PATCHGURU <number>
---------------------
```

Imagine the situation that your computer thinks 'Well, it's time for another guru.' Imagine also that you are, at this time, not at home. Who do you think should tell the computer to stop the guru stuff he's doing at the moment?

Well, Getty does.

It automaticly cancels the guru after the defined timeout (but only, if this feature is enabled) and the computer does a reset.

NOTE: If the computer does a reset, your current environment (the programms which are running) are wiped. So keep in mind that you have to think about a solution to restore this environment using a special startup-sequence to get Getty running again!

NOTE: This option is only available while starting Getty for the first time! If you define a timeout of 0 seconds, the computer reboots immediatly.

1.98 Command-Template: Specialty QUIET

```
QUIET
-----
```

Tell Getty (the server) to shut up and not to display any infos.