TimeKeep

COLLABORATORS			
	TITLE : TimeKeep		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 1, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

TimeKeep

1.1 TimeKeep Documentation

```
                                          TimeKeep
                    v 94.Oct.18
```

Description

Shareware

Tutorial

Development and Testing

Gadgets

Installing

Client Name Codes

Starting

Invoice.rexx

Configuration

InvoiceText.rexx

SoFar.rexx

Error Messages

Future Enhancements

Programming Invoices

Known BUGS

Legalness

```
                Thank Yous

                New in this version

                Previous versions

                Locating The Author
                              TimeKeep is a time-clock utility
            for keeping track of time spent on projects.
```

1.2 Description

```
Description
-----------
```

TimeKeep is a time-clock utility for keeping track of time spent on projects for client billing on a per hour basis.

Clients and Projects can be defined and a time clock interface used to TimeIn and TimeOut on a selected project. TimeKeep will remember which project is currently in progress and will select it automatically when the program is re-started.

In addition, TimeKeep has the ability to keep track of per hour rates and expenses for individual projects. TimeKeep will create output files containing this information which can be used to create invoices. As an example, ARexx scripts are included which will create invoices based on the TimeKeep datafiles.

This software could also be used for keeping track of hours spent on software development. However, another utility is now available to better handle this type of TimeKeeping: TimeKeep-Dev.

1.3 Shareware

```
                Shareware
---------
```

A great deal of time has been spent in the development of TimeKeep.

This version is SAVE disabled. If you wish to receive a non-disabled version of TimeKeep, a Shareware fee of $15.00 (CANADIAN) or more is required. Shareware fees of $20.00 (CANADIAN) or more will receive the next version automatically, at no additional charge.
See:
```
                Locating The Author
              Your shareware payment helps to insure continued development of  ←
                  TimeKeep.
```
See:
```
                Future Enhancements
```

1.4   Development and Testing

```
Development and Testing
-----------------------
```

TimeKeep was developed on an Amiga 3000T/25MHz with 2 Meg CHIP / 10 Meg
FAST memory.

TimeKeep has been tested under Kickstart version 37.175, Workbench
version 38.12.

1.5   Installing

```
Installing
----------
```

The following files should be contained in the same directory:

```
                        TimeKeep
                        TimeKeep.info
```

```
Sample .time file
-----------------
```

TEST.time is provided for testing the ARexx programs included in this
distribution.

Unmodified, the included ARexx scripts will look for TEST.time in the
S:TimeKeep directory. The first time you run TimeKeep, it will create
this directory, if it does not already exist. TEST.time should be copied
into this directory.

1.6   Starting

```
Starting
--------
```

From Workbench, double-click on the TimeKeep icon.

1.7   Configuration

```
                Configuration
-------------
```

TimeKeep supports configuration through use of Workbench ToolTypes. If
any options are not included as a ToolType, the stated defaults will be
used.

ToolTypes can be typed in upper or lower case or any combination of the
two. To have a ToolType ignored surround it with brackets ( ).

For example:
                                (ToolType=Value)

Invalid settings and ToolTypes will be ignored. In this case, the default
values will be used.

The following configuration commands are supported:

  AutoClientNameCode=
    Description: If set to YES, TimeKeep will automatically create a
                 Client Name Code using its built-in function, when adding
                 a new client.

                 See
                 Client Name Codes
                 for more information.

    Valid values: YES, NO

    Default: YES

    Example: AutoClientNameCode=NO

  BackFillColor=
    Description: Sets the palette color to use when filling the background
                 A value of -1 will disable the fill. Colors are numbered
                 from 0 up.

                 For MagicWB users, a value of 4 is recommended.

                 TimeKeep's NewScreen is currently a 4 color screen,
                 taking its palette from the Workbench.

    Valid values: -1, 0...(nColors-1)

    Default: -1

    Example: BackFillColor=4

  Database=
    Description: This is the FULL PATH and FILENAME of the data file
                 created by TimeKeep. This file is where TimeKeep stores
                 Client and Project information.

    Default: "S:TimeKeep.data"

    Example: Database="CLIENTS:.INVOICES/.TimeKeep.data"

  DataPath=
    Description: This is the FULL PATH to where TimeKeep will create its

```
                       .time files.  These files contain TimeIn and TimeOut
                       information for generating invoices.

          Default: "S:TimeKeep"

          Example: DataPath="CLIENTS:.INVOICES/.TimeKeep"


     DefaultProject=

       Description: This is the text to be used as the name of the default
                    project.  When a new Client is added, a default project
                    is automatically created.

       Default: "Miscellaneous"

       Example: DefaultProject="General"


     Font=
       Description: This is the name of an 8 point font to be used within
                    TimeKeep.  Currently, only 8 point Proportional or
                    Non-Proportional fonts are supported.

                    TimeKeep will use the Default System Font, regardless of
                    size, for the window title bar, if opened on the Default
                    Public Screen.  If you use a really large Default System
                    Font, you may have problems with the error message
                    display.  If this is a problem for you, try using the
                    NewScreen option, and the Font option to set a smaller
                    font.

       Valid values: Any 8 point, Proportional or Non-Proportional font.

       Default: "topaz.font"

       Example: Font="Topaz2p"


     NewClientRate=
       Description: This is the amount to use as the default Charge Rate when
                    adding a new Client.

       Default: "0.00"

       Example: NewClientRate="20.00"


     NewExpenseAmount=
       Description: This is the amount to us as the default when adding a new
                    Expense item.

       Default: "0.00"

       Example: NewExpenseAmount="10.00"
```

```
  NewScreen=
    Description: If set to YES, TimeKeep will open on its own Public
                 Screen.

                 If set to NO, TimeKeep will open on the default Public
                 Screen.

    Valid values: YES, NO

    Default: NO

    Example: NewScreen=YES


 RoundMethod=
    Description: This selects one of two built-in rounding methods.

                 If set to <|>, then the TimeIn function will round down
                 to the nearest RoundMinutes value, and the TimeOut
                 function will round up to the nearest RoundMinutes value.

                 If set to <>, then the TimeIn and TimeOut functions will
                 round to the nearest RoundMinutes value.

    Valid values: <|>, <>

    Default: <|>

    Example: RoundMethod=<>


 RoundMinutes=
    Description: This sets the minutes value when rounding TimeIn and
                 TimeOut.

                 If set to 1, this will cause the TimeIn and TimeOut
                 functions to round to the nearest minute.  This value,
                 in effect, disables rounding.

    Valid values: 1...60

    Default: 15

    Example: RoundMinutes=10
```

1.8 Error Messages

```
Error Messages
--------------

Could Not Create Gadgets

- TimeKeep was not able to create its gadgets.

Could Not Lock Screen
```

– TimeKeep could not get a lock on a default public screen, or could not
  get visual information from it.   TimeKeep may not have been able to find
  Workbench or another public screen to open on.

Could Not Open Font

– TimeKeep was not able to locate, or open the requested font as an 8
  point font.

Could Not Open Screen

– TimeKeep was not able to open its own screen.

Could Not Open Window

– TimeKeep was not able to open its window.

Insufficient Memory

– TimeKeep could not allocate memory to create its structures and lists.

ERROR: Client Exists

– The Client Name already exists.   There can not be two clients with the
  same name.

ERROR: Default Project can not be edited

– The name of the Default Project can not be edited.   It can only be set
  globally using the DefaultProject Tooltype.

ERROR: Insufficient Memory

– There was not enough memory to perform the requested function.

ERROR: Invalid Client Code

– The Client Code already exists or has not been confirmed by pressing
  ENTER within the Code gadget.   There can not be two Clients with the
  same Client code.

ERROR: Project Exists

– The Project Name already exists.   There can not be two projects with the
  same name.

1.9 Tutorial

                    Tutorial

This tutorial will single-step you through the normal operation of
TimeKeep.

The assumption is made that you have not changed any of the defaults
through use of ToolTypes.

ADDING A NEW CLIENT
===================

· Start TimeKeep by Double-Clicking on its icon.

When you use TimeKeep for the very first time, the only gadget that will
be selectable is the ADD gadget, under the Client list requester. If
data had already been entered into TimeKeep, that data would now be
displayed.

· Left-click on the ADD gadget under the Client list requester.

The Client Name string gadget is auto-selected.

· Type 'Test Number 1' {ENTER}

If you wish to abort ADDing, at any time click on the DELETE gadget.

The code 'TN1' is automatically entered into the CODE: string gadget and
the gadget is auto-selected.

If the auto-created Code is satisfactory, just press {ENTER}. If you
wish to alter it, you may do so. If you have set AutoClientNameCode to
NO, the gadget will be empty, and you must enter a Code. We will change
the default Code...

· Type 'TN1' {ENTER}

The gadget should now contain 'TN1TN1'.

When adding a new client, TimeKeep will now automatically select the Rate
string gadget. It will contain the NewClientRate as set by the ToolType.
You may edit the value if you wish.

· Type {ENTER}

'Test Number 1' will now be added to the Client List and the default
Project 'Miscellaneous' will be added to the Project List. The default
Project Rate will be set to the same as the Client Rate. In most cases,
the two rates will be the same. If necessary, TimeKeep provides the
ability to alter the Rate of each Project.

ADDING A NEW PROJECT
====================

· Left-Click on the ADD gadget, under the Project List.

The Project Name string gadget is auto-selected.

· Type 'Project 2' {ENTER}

If you wish to abort ADDing, at any time click on the DELETE gadget.

'Project 2' will now be added to the Project List. The Project Rate will
be set to the same as the Client Rate. In most cases, the two rates will
be the same. If necessary, TimeKeep provides the ability to alter the
Rate of each Project.

KEEPING TRACK OF TIME
=====================

When you are ready to start/continue work on a project, all you have to
do is select the project (or add a new project) and TimeIn.

· Left-Click on the Time In gadget

The current time, rounded as set by the RoundMethod and RoundMinutes
ToolTypes, will appear in the Time In string gadget. If the Date and
Time are satisfactory, you may continue. If you wish to make changes
you may do so by selecting the Date or Time string gadget and make
changes followed by {ENTER}.

TIMING-OUT OF A PROJECT IN PROGRESS
===================================

When you have already started a project using Time In, that project will
be remembered the next time you start TimeKeep, and will be automatically
selected.

If we were Exiting and Re-starting TimeKeep, the 'Project 2' project
would be automatically selected since it now has a TimeIn.

· Left-Click on the Time Out gadget

The current time, rounded as set by the RoundMethod and RoundMinutes
ToolTypes, will appear in the Time Out string gadget. If the Time is
satisfactory, you may continue. If you wish to make changes you may
do so by selecting the Time Out string gadget and make changes followed
by {ENTER}.

TimeKeep does not allow time to be carried on over 24 hours. For this
reason, changing the information in the Date string will change the Date
for both Time In and Time Out.

ADDING AN EXPENSE
=================

· Left-Click on the ADD gadget in the Expenses box.

The program will automatically select the Expense Name string gadget.

· Type 'Expense 1' {ENTER}

The program will automatically enter the Expense Amount as set by the
NewExpenseAmount ToolType and select the Expense Amount string gadget.

· Enter an amount of '14.35' {ENTER}

TimeKeep will enable the SAVE gadget to allow you save the entered
expense.

· Left-Click on the SAVE gadget in the Expenses box.

The program will now update or create the .time file with the information
entered.

QUITTING
========

If you wish to save any changes you have made, you should click on SAVE
or SAVE & QUIT.

· Left-Click on the QUIT gadget.

END OF TUTORIAL
===============
For further information about the use of TimeKeep see the
                 Gadgets
                 section.

1.10 Gadgets

Gadgets

So Far:

  This gadget displays the dollar value of time spent on projects for the
  currently selected client. It is calculated as:

      (<TimeOut> - <TimeIn>) * <CurrentProjectRate> + <CurrentValue>

  The following information may also be appear:

      ??????   The amount is unknown at this time.
      ------   The amount has been altered, but not saved yet.

Code:

  This gadget displays the current Client Name Code.

  If the AutoClientNameCode tooltype is activated, pressing 'HELP' within
  this gadget will re-invoke the builtin feature. This can be useful if
  you accidentally change the contents of this gadget and are not
  familiar with the method used to determine the Client Name Code. If
  AutoClientNameCode is not activated, pressing 'HELP' will clear this

gadget.

1.11 Client Name Codes

```
                Client Name Codes
-----------------

Client Name Codes can be a maximum of 6 characters long.

TimeKeep has a built-in function to automatically create Client Name
Codes.  At present, it simply combines the first letter of each word in
the Client name.

If you have a method of creating Name Codes that you would like to see
supported, contact the author.  See:
                    Locating The Author
                This feature can be enabled or disabled using the  ←
                    AutoClientNameCode
ToolType.


Applicable Configuration Commands:


                AutoClientNameCode
```

1.12 Invoice.rexx

```
                Invoice.rexx
------------

The included utility INVOICE.REXX is a Final Writer script for generating
invoices and keeping track of invoice numbers.

Invoice.rexx takes TimeKeep's .time files as input and supports Time
and Expense information.


Modifications should be made to Invoice.rexx to set it up for your system.


Modifying Invoice.rexx
---------------------

LINE 12:
                    DataPath = "S:TimeKeep"

  This definition should be identical to the DataPath ToolType definition.
  It is released with the default DataPath used by TimeKeep.  So, a change
  here may not be necessary.
```

```
LINES 273, 275, 277:
                              'Type' "<Name>"
                              'Type' "<Address>"
                              'Type' "<Phone Number>"
```

  These lines enter the 'Payment To' information at the bottom of the
  invoice. Substitute your mailing information within the "".


```
FONT:
                  'Font' "AvantGarde_Bold"; 'FontSize' 24
```

  Throughout the program you will find the above font and fontsize
  commands. These can be modified to use whatever fonts and fontsizes
  you wish to use. There are currently 19 occurances of this command
  line.

Starting Invoice.rexx

To use the ARexx script with Final Writer, assign it to a Menu or Button
using the USER MENU or USER BUTTON STRIP preferences -OR- Use the
USER/OTHER/AREXX SCRIPT menu selection to execute it.

A sample .time file is included with this distribution.
See:
                  Installing
                  for more information.

Using Invoice.rexx

Invoice.rexx assumes a method of invoice numbering as follows: Invoice
numbers are Client dependent. The first invoice for each client is
numbered 1, and following invoices are incremented by 1. Invoice numbers
are created as <ClientNameCode>##. For example: SDT01.

When first started, Invoice.rexx will prompt the user for a Client Name
Code. Invoice.rexx will look in the directory defined by the 'DataPath'
variable for a <ClientNameCode>.time file. To try the TEST.time file,
enter TEST.

If Invoice.rexx finds a .time file for the entered Client Name Code, it
will then look for a <ClientNameCode>.invoice file. This file contains
the invoice number of the last created invoice. It will be created by
Invoice.rexx if it is not found and will, initially, contain a 1. This
is a simple text file, containing a number followed by a carriage return.

If Invoice.rexx can not find a .time file for the entered Client Name
Code it will inform you and exit.

Invoice.rexx will then proceed to create an invoice. This will take a
few seconds, because FinalWriter's ARexx support is not as good as it
could be, and lacks support for some very necessary features. Be
patient, it is worth it.

Invoice.rexx creates 2 or 3 sections within the invoice: WORK, EXPENSES,
and WORK HOURS. If no expenses are found in the data file, the EXPENSES
section will not be included. If no time information is found, the WORK
section will not be included.

Upon completing the invoice, Invoice.rexx will position the cursor after
the first project in the WORK section. This section is used for giving a
decription of the kind of work done on each project. This must be
entered by the user, as Invoice.rexx and TimeKeep do not keep track of
this information.

If Invoice.rexx Does Not Meet Your Needs

If Invoice.rexx does not fully, or even partially, meet your invoicing
needs and you are not able to modify Invoice.rexx suitably on your own,
registered users may contact the author. See:
                 Locating The Author
                 If you have programming experience, you can refer to the

                 Programming Invoices
                 section for programming information to assist in
developing your own Invoicing utility.

1.13 InvoiceText.rexx

                     InvoiceText.rexx

The included utility INVOICETEXT.REXX is an ARexx program for generating
invoices and keeping track of invoice numbers.

InvoiceText.rexx takes TimeKeep's .time files as input and supports Time
and Expense information.

A sample .time file is included with this distribution.
See:
                 Installing
                 for more information.

Modifications should be made to InvoiceText.rexx to set it up for your
system.

Modifying InvoiceText.rexx

LINE 12:
                     DataPath = "S:TimeKeep"

  This definition should be identical to the DataPath ToolType definition.
  It is released with the default DataPath used by TimeKeep. So, a change

here may not be necessary.

LINE 104: (note: this line has been divided to fit within this text)

```
         call open('Out', DataPath"/TimeKeep/Invoice.
              "Client.NameCode''Client.Invoice, 'W')
```

  This line opens the output file. To redirect the output to somewhere
  else, modify this line. I do not recommend any other changes to this
  line, if you are not an experienced ARexx programmer.

  For example, to redirect output to RAM: change this line to:

```
    call open('Out', "RAM:Invoice."Client.NameCode''Client.Invoice, 'W')
```

LINES 195, 196, 197:

```
            call writeln('Out', "<Name>")
            call writeln('Out', "<Address>")
            call writeln('Out', "<Phone Number>")
```

  These lines enter the 'Payment To' information at the bottom of the
  invoice. Substitute your mailing information within the "".

Starting InvoiceText.rexx

From a SHELL or CLI type:

```
                RX InvoiceText <ClientCode>
```


InvoiceText.rexx takes only one parameter, a ClientCode.

For example:

```
                RX InvoiceText TEST
```


Using InvoiceText.rexx

InvoiceText.rexx assumes a method of invoice numbering as follows:
Invoice numbers are Client dependent. The first invoice for each client
is numbered 1, and following invoices are incremented by 1. Invoice
numbers are created as <ClientNameCode>##. For example: SDT01.

InvoiceText.rexx accepts a Client Name Code from the command line.
InvoiceText.rexx will look in the directory defined by the 'DataPath'
variable for a <ClientNameCode>.time file. To try the TEST.time file,
enter TEST.

If InvoiceText.rexx finds a .time file for the entered Client Name Code,
it will then look for a <ClientNameCode>.invoice file. This file
contains the invoice number of the last created invoice. It will be
created by InvoiceText.rexx if it is not found and will, initially,

contain a 1. This is a simple text file, containing a number followed by
a carriage return.

If InvoiceText.rexx can not find a .time file for the entered Client Name
Code it will inform you and exit.

InvoiceText.rexx will then proceed to create an invoice. Without
modification, InvoiceText.rexx will output a file in RAM: with the name:

                        Invoice.<ClientNameCode><Invoice#>

InvoiceText.rexx creates 1 to 3 sections within the invoice: WORK,
EXPENSES, and WORK HOURS. If no expenses are found in the data file, the
EXPENSES section will not be included. If no time information is found,
the WORK and TIME sections will not be included.

Upon completing the invoice, InvoiceText.rexx will position the cursor
after the first project in the WORK section. This section is used for
giving a decription of the kind of work done on each project. This must
be entered by the user, as InvoiceText.rexx and TimeKeep do not keep
track of this information.

If InvoiceText.rexx Does Not Meet Your Needs

If InvoiceText.rexx does not fully, or even partially, meet your
invoicing needs and you are not able to modify InvoiceText.rexx suitably
on your own, registered users may contact the author.
See:
                Locating The Author
                If you have programming experience, you can refer to the

                Programming Invoices
                section for programming information to assist in
developing your own Invoicing utility.

1.14 SoFar.rexx

                SoFar.rexx

The included utility SoFar.REXX is a shell script for getting Expense,
Work and Total values for work in progress.

SoFar.rexx takes TimeKeep's .time files as input and supports Time
and Expense information.

A sample .time file is included with this distribution.
See:
                Installing
                for more information.

Modifications should be made to SoFar.rexx to set it up for your system.

```
Modifying SoFar.rexx
--------------------

LINE 10:
                        DataPath = "S:TimeKeep"

  This definition should be identical to the DataPath ToolType definition.
  It is released with the default DataPath used by TimeKeep.  So, a change
  here may not be necessary.


Using SoFar.rexx
----------------

From a SHELL or CLI type:

                        RX SoFar <ClientCode>


SoFar.rexx takes only one parameter, a ClientCode.

For example:

                        RX SoFar TEST
```

1.15 Programming Invoices

```
                  Programming Invoices
--------------------

If you write your own TimeKeep invoicing utility for Final Writer or any
other software, please send a copy to the author for inclusion with this
distribution.  See:
                  Locating The Author
                  .time File Format
------------------

This is the format of the .time data file created by TimeKeep.  See
TEST.time, included in this distribution, for an example.


LINE 1:
                  "<ClientNameCode>","<ClientName>"

  example:
                  "TEST1","This is Test #1"


LINE 2...n:

  "T","<Project>","<DateIn>","<TimeIn>","<TimeOut>","<Hours>","<Rate>"

                              - OR -
```

```
                    "E","<Project>","<Expense>","<Amount>"
```

  where:

  "T" = Time Information

```
    <Project> = Project name
    <DateIn>  = Time In date
    <TimeIn>  = Time In time
    <TimeOut> = Time Out time
    <Hours>   = TimeOut - TimeIn (hours)
    <Rate>    = Project hourly rate
```

  "E" = Expense Information

```
    <Project> = Project name
    <Expense> = Expense name
    <Amount>  = Expense amount
```

  examples:

```
    "T","Test Project","May 16","12:15 PM","1:00 PM","0.75","15.00"

            "E","Test Project2","This is an Expense","7.77"
```

1.16 Legalness

```
                Legalness
---------
```

TimeKeep is copyright ©1994 by Scotware\Software. With the exception of
the ARexx utilities, the files included with this distribution may not be
altered in any way. TimeKeep is released as Shareware - You are required
to pay a
                Shareware
                fee if you continue to use this software beyond a
trial period of 10 days.

The software concept and design remain the property of the author,
scott dhomas trenn and Scotware\Software.

Permission is granted for this program to appear in any public domain
software archives.

Permission is granted for this program to appear on any commercial disks
or CDROMs, provided that a single copy of the first product, containing
this version of this progam, is sent to the author.

The author is in no way liable for any damage resulting from the use of
this program.

1.17   Future Enhancements

```
Future Enhancements
-------------------
```

- Nettwork support, Multi-user
- Larger font support
- Definable Client Code function
- Busy pointer
- Select InProgress button gadget
- Cancel TimeIn/TimeOut button gadgets
- Enhance Invoice.rexx: File Requester, Move default Project to top of
  project list, etc.
- Built-in Invoicing???
- Runnable from CLI
- and MORE...

1.18   Known BUGS

```
Known BUGS
----------
```

- In Progress counter is not completely working.

1.19   Locating The Author

```
Locating The Author
-------------------
```

If you like this program and would like to register, reward me for my
efforts, or for bug reports, program ideas, or questions, or just to tell
me about where you live.

I can be contacted through INTERNET EMAIL at:

                        dhomas@jupiter.sun.csd.unb.ca

Or by MAIL at:
                           scott dhomas trenn
                          Scotware\Software
                         797 Mitchell Street
                           Fredericton, NB
                           CANADA E3B 3S8

Registered users may send me $5.00 (CANADIAN) and a disk, for the latest
version of this software.

Be sure to include the version number you are currently using, and the
name of the program you want, so I do not send you a version you already
have or a program you do not want. The version number can be found in the

```
titlebar,  at the top of this documentation file, or by typing:

                            Version TimeKeep

at a shell prompt.  It looks similar to    94.Oct.15

When reporting bugs or errors generated when you run TimeKeep, please
include any error messages given by TimeKeep and also the version number
you are using.  This will aid me in tracking down the problem. Thank-you.

If you have an EMAIL address please include that with your letter.
```

1.20 Thank Yous

```
Thank Yous
----------

Thanks to the following people for beta-testing, suggestions, bug reports
and support:

- Greg Perry         - Ashgrove, Qld              AUSTRALIA
- Troy Morehouse     - Fredericton, New Brunswick CANADA
- Nickey MacDonald   - Fredericton, New Brunswick CANADA


                                                  - dhomas trenn
```

1.21 New in this version

```
New In This Version
-------------------

- Modified UnRegistered version to allow partial saving.  This should make
  it easier to decide if you want to get the full registered version.

- Added 'Cancel' TimeIn/TimeOut gadget.

- Background color is set to the BackFillColor tooltype setting, for
  (Un)Registered text.

- Invoice.rexx: Fixed missing quotes.

- InvoiceText.rexx: Fixed missing quotes.
```

1.22 Previous versions

```
New In Version 94.Oct.15
------------------------

- Added 'Client SoFar' to keep track of current amounts.
```

– Invoice.rexx: Updated to support 'Client SoFar'.

– Fixed bug in GetTime function when using RoundMethod = "<|>".

– Added InvoiceText.rexx for generating ASCII text Invoices. These text
  files can be imported into programs that do not support ARexx or
  provide other means of importing TimeKeep invoices.

New In Version 94.Jul.26

– Fixed Enforcer hits in the Add Client routine.

– TimeKeep now also creates the empty .time file for a client, if it does
  not already exist, when an expense is added.

– Invoice.rexx: Removed a line that did nothing.

– Invoice.rexx: Work section is no longer added if no time information is
  found.

– Added SoFar.rexx.

New In Version 94.Jun.16

– First Release