WibuKey

The Key is in Your Hands!

WkNet

File based protection in the network

Version 5.20, Edition November 2006

Contents

1 WkNet: File Based Protection

With WkNet, the network encryption of software or data files requires Explicit Encryption. Automatic Encryption support for WkNet is planned for the future. WkNet API calls are executed asynchronously, which is not as simple as the synchronous structure of WkLAN. This chapter explains the principles of the WkNet protection.

The WkNet network protection does not depend on any special hardware or network protocol. It is therefore possible to use nearly any existing network, even heterogeneous networks. WkNet is based on a shared encrypted file. This file is called the **WkNet Server File**. It executes the limitation of the number of simultaneous network copies. For this purpose, each client program allocates a specific entry called a **slot**. This slot is freed when the client application is properly terminated. The maximum slot number in a single WkNet Server File is 250. The number of slots which may be allocated at same time is read by the WIBU-KEY Server from the connected server WIBU-BOXes and stored into the WkNet Server File. If a started application cannot find a free slot it must terminate itself with a corresponding error message. The slot administration is handled internally in the WIBU-KEY calling driver. Upon termination, the application releases its slot automatically. Should an application happen to crash, its slot remains occupied. Such a slot may be freed manually by the end user with the aid of the WIBU-KEY Server Monitor. The WIBU-KEY Server running on the network server computer updates the encrypted WkNet Server File regularly. This update of the server file is done in fixed intervals by the WIBU-BOX connected to the server. The interval may be defined by the software developer. It can extend from 10 seconds up to 1 hour, typically values are 1 up to 5 minutes. In general, the following rule is valid: The smaller the time interval, the higher the protection and the faster the encryption result but also the higher the network and complete system overhead. An end user can work with protected software copies at least twice as long as this time interval specifies, without the necessity to connect a WIBU-BOX to the WIBU-KEY Server.

Protected software started from a network client will check in the specified interval whether the updating of the server file has occurred correctly. Should this not be the case, the software assumes that the WIBU-BOX is not connected to the WIBU-KEY Server. In the following, further operations of the client application are not possible. The end user is usually given the possibility to save changed data.

1.1 WkNet Configuration

The implementation of WkNet includes the creating of the WkNet Server File which connects the clients and the server. Also the configuration of a WIBU-KEY Server, the configuration of the clients, and the protection of the desired software so that it is compatible with WkNet.

1.1.1 Creating a WkNet Configuration File

A WkNet Server File is created by the **WKCRYPT** program using the special **/NET** command line option. Further options define the maximal number of network users for a program and the buffer size for the network encryption.

After starting a WIBU-KEY Server which manages a WkNet Server File, this file is encrypted for a specific Firm Code, User Code, and Selection Code. When the file is created by WKCRYPT you can decide, if you want to encrypt the file immediately or if you want to create the file non-encrypted.

Advantages and disadvantages of a **pre-encrypted** WkNet Server File

⇧ It can be accessed by a WkNet client even when the WIBU-KEY Server isn't running. Then a proper error "network not ready" (99) appears.

⇧ It avoids the case that the created WkNet Server File may be used by any Firm Code, User Code, and Selection Code. The file is fixed to the codes of the pre-encryption.

⇩ The created WkNet Server File may be used only for a specific Firm Code, User Code, and Selection Code which avoids a flexible use of the file in dependence of the used code of a specific end user.

Advantages and disadvantages of a **non-encrypted** WkNet Server File

⇧ It can be used for any Firm Code, User Code, and Selection Code in dependence of a specific end user.

⇩ When the end user starts the WkNet client and reads the WkNet Server File before the WIBU-KEY Server has been updated the file, an error "bad net system" or "corrupt WkNet File" (101) appears. This may be confusing for the end user.

⇩ The set WkNet Server File parameters are not fixed to specific Firm Code, User Code, and Selection Codes. There is no control which codes use critical parameters like number of maximum slots, encryption buffer size.

The first encryption of a non-encrypted WkNet Server File is automatically done by the first server access to file. A WkNet client or the WIBU-KEY Server Monitor cannot access non-encrypted WkNet files.

The first 32 bytes of a WkNet Server File contain an extended file name. This text is terminated by the file end identifier 0x1A and appears, for example, when the file is printed with the system console command TYPE. An extended file name may contain up to 32 characters and may be specified with the /N option. As default the following text will be used:

```
WIBU-KEY-Std. WkNet Net File
```

At the end of the WKCRYPT specification line, the name of the WkNet Server File itself must be specified.

In summary, the WkNet file generation syntax is:

WKCRYPT /NET [*Option…*] *FileName*

The following options are available for **WKCRYPT /NET**:

Option	Brief Description
/1	sets a WkNet file version 1 which is supported by the old WKNET.EXE server process for DOS.
/2	sets a WkNet file version 2. Such a file supports the listing of the WIBU-BOXes which are connected at the server and supports an optional customer area in the WKNet Server File, set by the **/R** option. It is recommended to use a WkNet file version 2 unless the file must be managed by the old WKNET.EXE server process for DOS.
/B*Number*	specifies the buffer size of the encryption block in every slot of a WkNet Server File. The specified value may be between 0 and 255. The default length is 32 bytes. This block is used to encrypt data which is transferred from a WkNet client to the WIBU-KEY server. The length of the encryption block specifies the maximum length of a data sequence for encryption.
/F*FirmCode*	specifies the Firm Code for a pre-encrypted WkNet Server File.
/N:*Text*	allows the specification of a name which is set as extended WkNet Server File name.

/Q*Number* specifies the maximum physical number of users who may simultaneously use the software protected by a WkNet Server File.

When a WkNet file is protected by a WIBU-KEY server, this maximum specified number is reduced by the "logical" number, as defined by the Network Entry of the WIBU-BOX which calculates the User Quantity and controls the WkNet Server File.

The physical number defines the size of the WkNet Server File. Because larger files are read and verified more slowly, the value specified should not be larger than appropriate for the network protection. The value may be between 0 and 250. When this option is not set, the maximum physical number will be set to 32 as default.

/R*Number* reserves a customer specific area of *Number* bytes at the end of the WkNet Server File. This area may be used by software developers to store application specific data that is not changed by the WIBU-KEY Server. By default, no such area will be reserved.

/S*SelectionCode* specifies the Selection Code for a pre-encrypted WkNet Server File. If the Selection Code is within the range 0 up to 65,535, a WIBU-BOX version 1 may be used for the WkNet protection because encryption algorithm 1 is used. If the Selection Code is larger than 65,535, Selection Code 2 is always used.

If this option is not specified, Selection Code 1234 is used as default.

When several WkNet Server Files are created by a WKCRYPT command file, the Selection Code must be specified separately for each WkNet Server File creation line. Every file creation resets the Selection Code to the default value.

/U*UserCode* specifies the User Code for a pre-encrypted WkNet Server File.

/U[**M**]*UserCodeNumber*[*Delta*]
sets the (*Master*) User Code for the following encryption, programming, or network file creation specifications.

/U disables any set User Code. This option may be used to create a non-encryption of a WkNet Server File after creating a pre-

	encrypted WkNet Server File within one WKCRYPT command file.
/V	activates the extended output mode (*verbose*), which issues more information on the monitor during the execution of the process.
/?	prints a brief description of the program and all options which are permitted in combination with the **/NET** option.

Example: Creating a WkNet Server File

```
WKCRYPT /NET /N:"WkNet for DataBase" /Q60 /B128
        WKNET.DAT
```

generates a file called WKNET.DAT with the description "WkNet for DataBase". It sets the number of users to 60, and the buffer length to 128 bytes. This WkNet file is not encrypted.

```
WKCRYPT /NET /F10 /U13 /N:"WkNet for DataBase"
        /Q60 /B128 WKNET3.DAT
```

creates a pre-encrypted file WKNET3.DAT for Firm Code 10, User Code 13, the default Selection Code 1234 and encryption algorithm 1.

Note: On the end user's side, a shipped WkNet Server File should be saved for a simple reinstall after a crash of a server system. In rare cases, a crashed WIBU-KEY Server may damage the WkNet Server File encryption. A restarted server cannot decrypt this wrong encryption. As result a *WkNet Server File is corrupt* error message occurs. To solve such a conflict, the saved WkNet Server File can be used to overwrite the corrupt file.

1.1.2 Connecting the WIBU-KEY Server to a WkNet Server File

The connecting of a WIBU-KEY Server to a WkNet File may be specified independently of using a WIBU-KEY Server application but must be done at the computer where the server will be executed in future.

Because this setting is read from WIBUKEY.INI when the WIBU-KEY Server is started, this setting must be done *before* the server start. Any change in the WIBU-KEY Server settings requires the manual stop and restart of the server application.

The WIBU-KEY Setting is done in the *ServerWkNet* page of the WIBU-KEY Control Panel Applet. In the *WkNet Setting* section of this page the following parameters must be set for each WkNet Server File which is managed by the WIBU-KEY Server:

- The Firm-/User-Code column entry contains the Firm Code and User Code

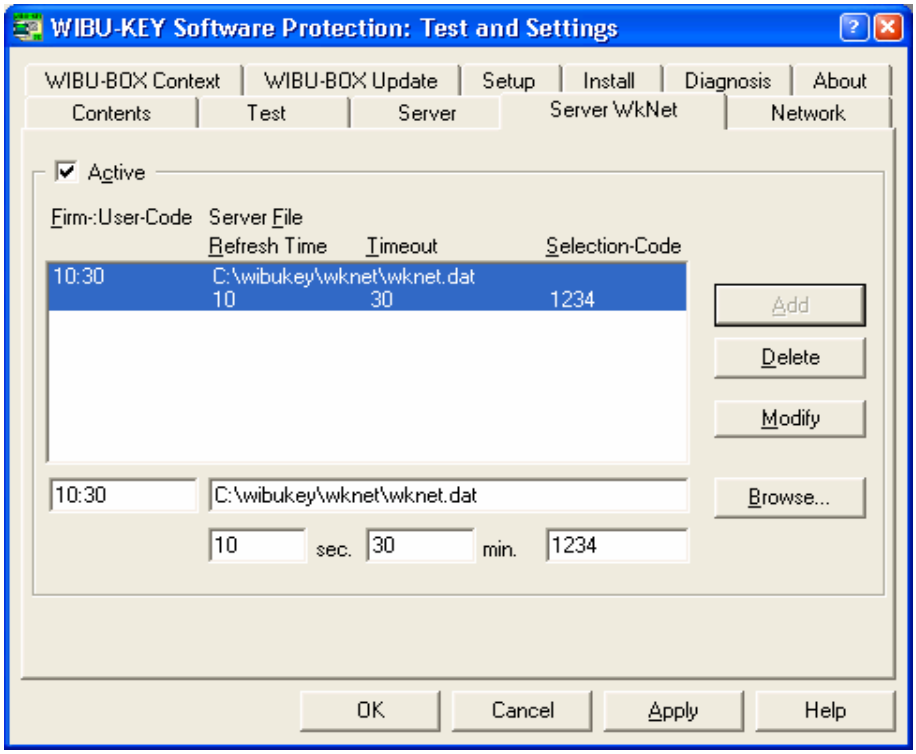

Fig.: WIBU-KEY Control Panel Applet: ServerWkNet page

  which must correspond with the WkNet Server File and the desired WIBU-BOX Network Entry.

- The Server File column entry specifies the name of the WkNet Server File which is to be controlled by the WIBU-KEY Server. This file doesn't need to be stored locally on the WkNet Server computer. It can be stored on any station within the network system. It is only necessary that the WkNet server as well as all workstations can perform read and write operations on this file. The Windows dialog for searching the WkNet Server File can be opened via the *Browse...* button.

- The Refresh Time column entry specifies the time in seconds for the update or refresh process of the contents of the WkNet Server File. These values may be in the range from 10 seconds up to 3600 seconds (1 hour). The default value is 10 seconds.

The *Refresh Time* value must be the same as the value which is set in the protected WkNet client application. Otherwise errors like *network not ready* will occur in indefinite repetition. If the value of the refresh time has been changed in the WkNet client application, it is necessary to set the server's refresh time to the same value.

- The Timeout column entry specifies the time in minutes after which the WIBU-KEY Server cancels clients that have not accessed the WkNet Server File in this time interval. As default no WkNet timeout is set. The permitted value range of Timeout depends on the selected Refresh Time value. The Timeout value has to be a minimum of three times and a maximum of 249 times the Refresh Time value:

Refresh Time (seconds)	min. Timeout (minutes)	max. Timeout (minutes)
10	1	41
20	1	83 (1 h 23 min)
30	2	124 (ca. 2 h)
60 (1 min.)	3	249 (ca. 4h)
300 (5 min.)	15	3735 (ca. 2 days, 14 hours
600 (10 min.)	30	7470 (ca. 5 days)
3600 (1 hour)	180	44820 (ca. 1 month)

- The Selection Code is another parameter that is needed for the encryption. This must be the same value which is specified in the protected WkNet client application. If the values differ, the WkNet Server File is encrypted incompatibly with the WkNet client and an error code "Network is damaged" is returned on the client side. If no Selection Code is specified, the default value is 1234.
- One WIBU-KEY Server can control several WkNet Server Files simultaneously. The values in the edit boxes are added to the list with the *Add* button to receive a new WkNet Server File in a new row in the WkNet list box. The entries can be removed with the *Delete* button. The *Modify* button changes the currently selected WkNet Server File row.

Note: The Novell Netware Server NLM reads its setting from the WIBUKEY.INI file at the server computer. Because a Novell Server does not support interactive tools like the WIBU-KEY Control Panel Applet, the setting must be done manually in the WIBUKEY.INI files. For this, each WkNet Server File has its own topic with the Firm Code and User Code. The variables of this topic have the same names as explained above and the same value settings.

1.1.3 WkNet Server Configuration

On the server side, the WkNet configuration is more complex than the WkLAN configuration because each WkNet Server File must be installed manually. The required WIBU-KEY drivers and the WIBU-KEY Control Panel Applet must be installed including the calling and kernel drivers.

Note: It is a good idea to test the WIBU-BOXes locally by the WIBU-KEY Control Panel Applet before you start the WIBU-KEY Server.

After that, the best suitable WIBU-KEY Server must be selected:

- WKSVW32.EXE for Windows 95/98/Me/NT/2000/XP. On Windows NT this server can be started as system service or as application. On Windows 95/98/Me/NT/2000/XP the driver is started as application. A service may be started without a logged-in user.
- WKSVW16.EXE for Windows for Workgroups 3.11. This is 16-bit variant of WKSVW32.EXE. It supports WkNet but the limited multitasking support of Windows 3.11 reduces the system throughput more than on Windows 95/98/Me or Windows NT.
- WKSVMAC for Mac OS 8/9. This is the variant of WKSVW32.EXE as Macintosh application.

- WKSVNLM for Novell Netware 3.11, 3.12, 4.0, 4.01 or 4.10.

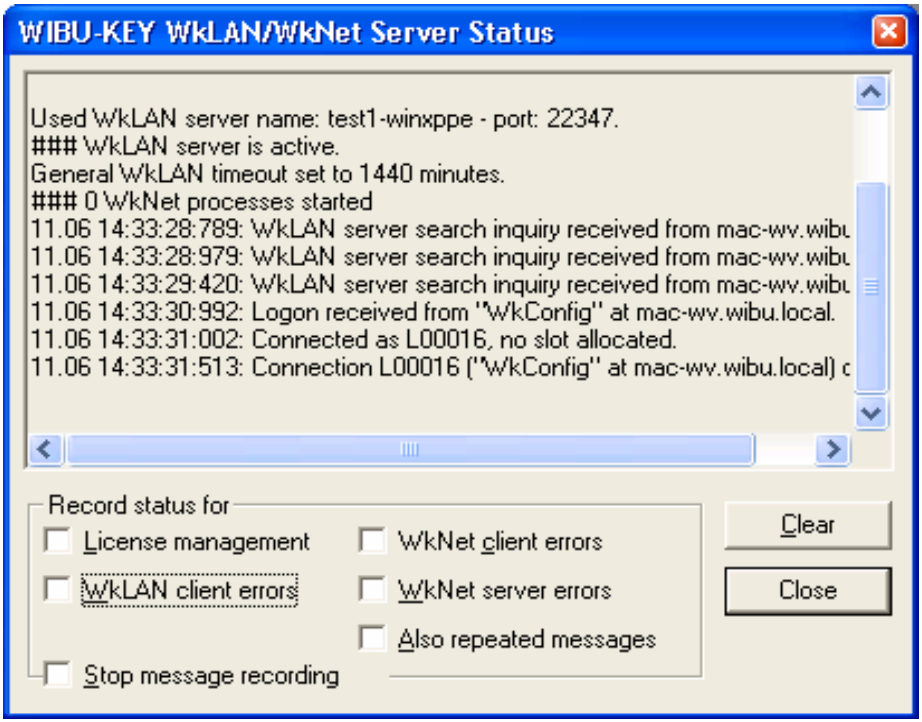

Fig.: The status text of the WIBU-KEY Server

For a proper test execution of WIBU-KEY Server, the following requirements must be done:

- A WIBU-BOX with one or more proper WIBU-KEY Network Entries must be connected.
- The corresponding settings for addressing the WkNet Server File.

The WIBU-KEY Server lists all WkNet Server Files with its controls at the end of initial text in the server status window.

1.1.4 WIBU-KEY Server for Novell Netware

The WKSVNW.NLM file must be copied into the SYSTEM directory of the server and can be started with following command line:

LOAD WKSVNW.NLM *WibuKeyIniFileSpec*

It uses the WIBUKEY.INI file which contains additional control information. This file is a readable text file and uses the syntax of Windows INI files. For the WkNet protection only sections with the title of the form

```
[WkNet FirmCode:UserCode]
```

are used in which FirmCode and UserCode specify valid Firm Codes and User Codes. To guarantee that WKSVNW.NLM finds the WkNet file it is needed that the complete path of this file is specified in the WIBUKEY.INI file. Novell doesn't use driver letters like C: but normally drive names like SYS: or DOS:. The WKSVNW.NLM uses the environment settings in the [DRIVER] section of the WIBUKEY.INI, compatible with the WKWIN.DLL.

```
[WkNet 10:13]
ServerFile=dos:\test\wknet.dat
RefreshTime=10
TimeOut=30
Selection=1234
```

This section contains entries with specified key words followed by an equation sign (=) and the contents. Comments start with a semicolon (;):

Entry Key Word	Meaning
`ServerFile`	contains the complete path of the WkNet Server File.
`RefreshTime`	specifies the time interval in seconds in which the WkNet Server File is updated. The default value is 10 seconds.
`TimeOut`	specifies the time-out value in minutes. It frees automatically an allocated slot in the **WKNET.DAT** file if the allocating process has not accessed its slot in the specified time. If this entry is not set, no TimeOut is set.
`Selection`	specifies the Selection Code which is used by the protected software. This value must be specified by the software developer. The default value is 1234.

1.1.5 WkNet Server Test

Before a WkNet protected program is tested, the proper updating of the WkNet Server File by the WIBU-KEY Server should be checked. For this, two WIBU-KEY tool programs are available:

- WkSvMon is a Win32 Graphic Windows Application

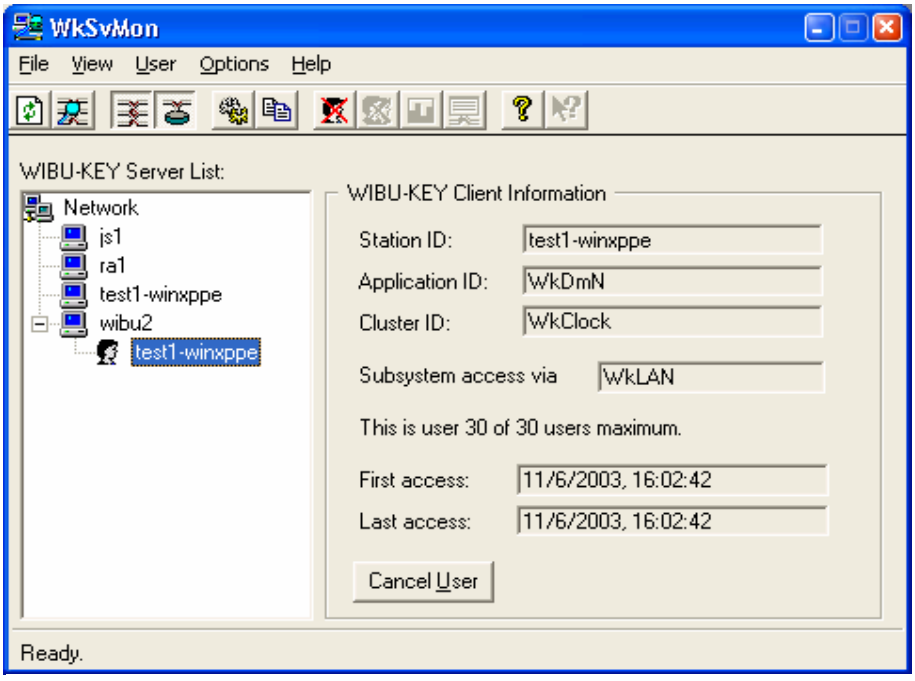

Fig.: WkSvMon listing the first and last access to the WkNet Server File

- WkNetLst is a DOS Application.

To test the updating of a WkNet Server File with WkSvMon, start this file and open the desired WkNet Server File or select a user. The update time should be increased to the set refresh time interval when the refresh button of WkSvMon is pressed. Compare the time of the first and the last access.

When no Win32 system is available, the simple WkNetLst DOS program can be used from the command line. To test the update interval of the WkNet Server File *wknet.dat* on system *server1*, the following line may be entered and output may appear:

Example: WKNETLST Call

```
D:>wknetlst \\server1\data\wibukey\wknet.dat
WKNETLST - WIBU-KEY WkNet Network File List/Cancel Tool.
Version 2.50 of 98-Jul-03 for DOS.
Copyright (C) 1991-98 by WIBU-SYSTEMS AG. All rights
reserved.

WkNet file: version 2.00, last update at 10.05.01 13:08:56
```

By calling WkNetLst repeated times, the listed *last update* should be increased in the specified refresh time interval.

1.1.6 Protecting Software for WkNet

Note: With WKCRYPT version 2.50, WkNet does *not* support the Automatic Encryption of executables. It is planned for future versions of WKCRYPT, the supported platforms will be 16 bit Windows and Win32.

The incorporation of the WkNet network protection into an application must be done explicitly. For the basics about Explicit Encryption and using the WIBU-KEY API from within the source code of your application see the Programmer's Guide.

WkNet protects software which runs over a long period. A user should be able to work with this software interactively. Programs executed within a brief period cannot be protected because the interrogation of the network protection system occurs in intervals and not instantly. This includes many non-interactive programs which are started from the command line.

With interactive programs, the verification of the updating of the WkNet Server File and the data encrypting via the WIBU-BOX at the Server occurs upon activating specific commands (for example from a menu) and/or in a time controlled fashion.

Note: The WkNet access of a protected client application permits the use of most parts of the WIBU-KEY API (see WIBU-KEY Programmer's Guide). The two most important restrictions are the limited length and the time delay with which the encryption occurs. Furthermore, the asynchronous calling of the API functions is required which increases the overhead and the complexity in using the WIBU-KEY API functions.

1.1.7 Connecting a Client to a WkNet Server File

Every client application which is protected by WkNet must access the WkNet Server File. There are some requirements to access such a file:

- There must be read and write access to the WkNet Server File. This access must be possible byte by byte at any file location, provided that these locations are not periodically blocked by other stations.
- It must be possible to occasionally lock individual areas of the server file against access from other stations. For this purpose, a byte-oriented record locking is necessary; a sector or file-oriented locking is not adequate.

If both requirements are valid any operating system may be used to access a WkNet file.

For Windows 3.11/95/98NT/2000, DOS and Apple Macintosh, ready to use drivers are available in dynamically or statically linkable driver libraries. The kernel drivers

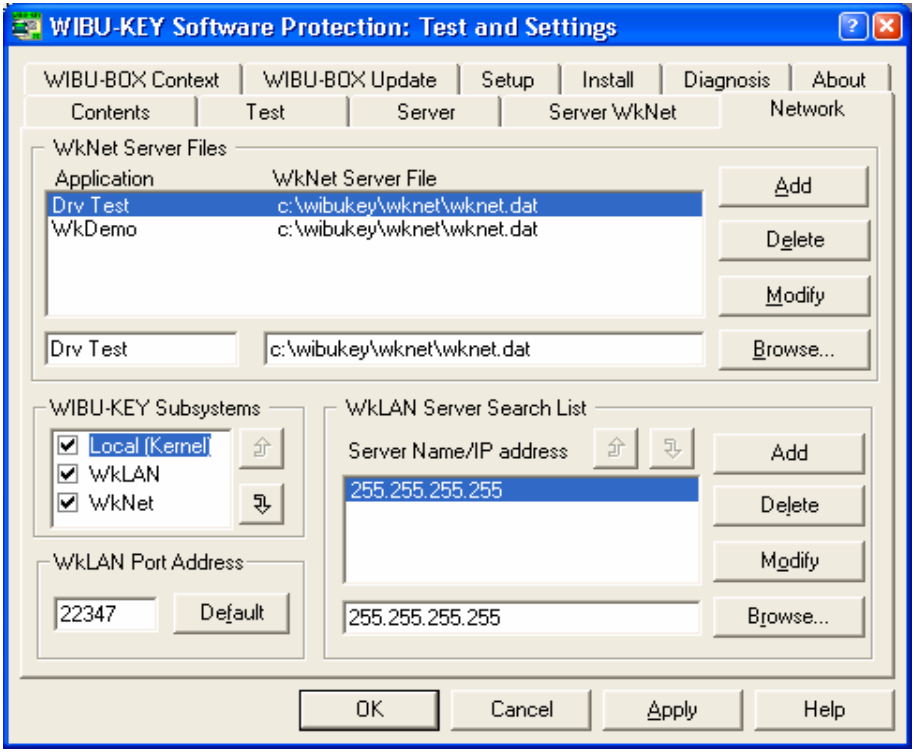

Fig.: Control Panel Applet: Network page

for 95/98/NT/2000 are not required to support WkNet on client side.

The path to the WkNet Server File may be specified at two different locations:

- The most flexible way is to store the path for a specific application in the WIBUKEY.INI (in the system registry in the future) and enter the name by the WIBU-KEY Control Panel Applet (see below). This storing is not supported by DOS, Apple Macintosh or the Source Driver for universal system adaptation.
- The protected client application itself stores the path to the WkNet Server File and transfers it by the WIBU-KEY API to the driver. This option is supported by all driver variants.

On Windows, the path to the WkNet Server File may be specified interactively by the *Network* page of the WIBU-KEY Control Panel Applet. For this, the *WkNet Server Files* area is used: In the left row, the name of the application is entered. In the right row, the full path name of the WkNet Server File is entered.

The values in the text input boxes can be transferred to the list box by the *Add* button. The *Delete* button removes entries. The *Modify* button is for the modification of an existing entry in the list. With the *Browse...* button it is possible to open a Windows dialog box to search for a WkNet Server File.

1.2 WkNet Client Test

After creating a suitable WkNet Server File, configuring and starting the WIBU-KEY Server for this WkNet Server File, and setting the WkNet access on the client side, you can test a WkNet protected file. For this use the WkDmN example WKDMN16.EXE for Windows 3.x and WKDMN32.EXE for Windows 95/98/Me/NT/2000. They are in the *Samples* directory of the WIBU-KEY CD-ROM. Before starting this example, following preparations should be done:

- Create a WkNet Server File which is controlled by a WIBU-KEY Server for Firm Code 10, User Code 13 and Selection Code 1234.
- Check the WkNet Server File update by the WIBU-KEY Server Monitor or WkNetLst application.
- Access the WkNet Server File on the client computer using the name WKDMN and the path of file.
- Be sure that no local WIBU-BOX is attached at the client computer to avoid that the WKDMN application utilizes the local WIBU-BOX.

Now you can start WkDmN16.EXE or WkDmN32.EXE.

Fig.: Notification of a WkNet Server access by WkDmN

When a WkNet Server is found by WkDmN, it displays following message box. After pressing *OK* you can *Add* or *Subtract* in the main windows of the dialog to add or subtract numbers.

During this application execution, the WkNet server access is permanently checked by sending data to the WIBU-BOX at the server. If you remove the WIBU-BOX at the server, the *WkDmN* application displays an error message box after at most 30 seconds stating that the protecting WIBU-BOX is not found.

1.3 Selecting a Local or Network Subsystem from a Client Application

It is recommended to implement the WIBU-KEY access to an application in such a manner that the application can be protected by a local WIBU-BOX, by WkLAN or by WkNet and that the end user of the protected application can choose the best protection method based on their individual needs.

WIBU-KEY has three different subsystems:

- *Local (Kernel)*, the access on local WIBU-BOXes via the kernel driver,
- *WkLAN*, the protocol based access to WIBU-BOXes at a WIBU-KEY Server.
- *WkNet*, the access to WIBU-BOXes at a WIBU-KEY Server via WkNet Server Files.

This list describes the default search order from first to last subsystem.

The end user may choose in the *Network* page of the WIBU-KEY Control Panel Applet:

- Which WIBU-KEY subsystems are accessed by applications on the specific client computer.
- The search order of the three subsystems.

Note: This setting does not influence the operation of the WIBU-KEY Server. The server application always addresses connected WIBU-BOXes at the local server computer, no matter if the local subsystem access is activated or not.

The setting is done in the *WIBU-KEY Subsystems* list box:
A check box before the option means that this subsystem is active for searching. The sequence for searching the subsystems can be changed with the *Up* and *Down* buttons. In the example above, the WkLAN and the local subsystem are selected. The WkNet subsystem is unselected.

2 Requirements of WkNet to the Network

WkNet sets a number of demands on the network in which the software to be protected is installed and should run. *WkNet* has been specifically devised to minimize these requirements and to ensure that the software, which is protected by *WkNet,* can run with practically any network without problems.

In general, a network consists of a number of stations and one or more servers. The server is distinguished by the fact that files, which have been stored on its disk, may be called up from all other stations (provided this is not inhibited by the administration for reasons of safeguarding or data protection). The terms *station* and *server* need not necessarily exclude each other. Indeed, network systems exist in which, contrary to other systems, stations may also be servers. Finally, systems also exist for which the term *server* does not explicitly exist, since the files of one station may not be directly accessible from another.

For purposes of simplicity, the following description of *WkNet* distinguishes between *station* and *server*. In order to employ *WkNet* in a network, the following prerequisites must apply:

- A file may be installed on the *server* for which read and write access is warranted for all stations which incorporate *WkNet* protected software
- On each station from which *WkNet* protected software may be initiated by the user, read and write access to a specific file from the server must be warranted.

For *WkNet*, it is irrelevant which operating system runs on the server or stations. Even the system hardware, which is used by the server or stations, is principally not of importance. However, one single system within the network must be capable of

access to the WIBU-BOX, which incorporates the network protection *WkNet*. This system must fulfill the following requirements:

A direct access to the WIBU-BOX must be possible. On the one hand, this demands hardware, which is compatible to IBM-PC hardware, and on the other hand requires an operating system, which does not prevent access to the hardware.

- The safeguarding program, which continually checks the WIBU-BOX hardware and transfers test results on to the protection file on the server, must be executable by the system. Such a system may be Windows 9x, Windows NT/2000, Netware 3.x, Netware 4.x, Netware 5.x, DOS or Mac OS. The required *WkNet* processes for this system are part of the WIBU-KEY Base-Kit. For Windows and Netware, the *WkNet* server is a true process.

All other system stations are not subject to any restrictions whatsoever; with the exemption that access to the file on the server must be possible. This access must comply with the following demands:

- Read and write access to the file on the server must be warranted. This access must be possible byte by byte at any file location, provided that these locations are not (periodically) blocked by other stations.
- It must be possible to occasionally block individual areas of this file on the server against access from other stations. For this purpose, a byte-oriented *record locking* is necessary; a sector or file-oriented *locking* is not adequate.

WkNet therefore only imposes minor demands on the characteristics of a station. It need neither be a PC nor must it possess a DOS operating system. Hence, with *WkNet*, one cannot only protect OS/2 or UNIX software on PCs but also, for example, UNIX software on any other hardware platforms, and also *Macintosh* programs. Due to the fact that the configuration of the safeguarding file on the server is constant, the access to this file can be transferred to various operating systems or processors with acceptable effort.

2.1 Requirements on the Software to be protected

Due to the fact that the interrogation of the network safeguarding system via *WkNet* only occurs intermittently and not instantly, one cannot protect applications that are executed within a brief period. This incorporates many non-interactive programs that are started from the command line. This confinement

does not involve the limitation of the copying within a network, but just the copy protection for the distribution of the program between numbers of networks.

WkNet can principally protect all programs in which one can interactively work for a longer period. The incorporation of the copy protection *WkNet* in an application must be conducted, exactly in which three main processes must be implemented:

- Following the initiation of the application, the checking of the server file, the search of a free slot and its exclusive occupation for the application.
- As the application runs, the periodic checking of the server file updating via its time entry. Following the interruption of this updating, the automatic storing or blocking of the application until the updating occurs in the correct fashion.
- Upon terminating an application, the release of the server file slot so that another can occupy it.

In addition, in order to increase the reliability of the copy protection, an application may encode or decode data via the WIBU-BOX with a time delay. For this purpose, the following operations may occur:

- Transferal of a short (up to maximal 256 bytes) but otherwise any data configuration in the user-specific slot with the specification of a firm, user and Selection Code for the encoding.
- Following the successful verification of the server file update, the reading out of the filed data structure that is then automatically encoded or decoded by the *WkNet* process via the connected WIBU-BOX.

Note that this encryption permits the same variation profusion as the base API. The only two restrictions are the limited length and time delay with which the encoding occurs.

With interactive programs, the verification of the updating of the server file and the optional encoding via the WIBU-BOX occurs upon activating specific commands (for example from a menu) and/or in a time controlled fashion.

2.2 Enhancement of the Security of WkNet

The *WkNet* server file can be read-in and also changed from any station. This is not only open to the protection of applications, but also to attempts to manipulate and avoid the *WkNet* protection. In this respect, there are two main server file targets:

- The cancellation of the protection for the illegal transferal of protected software from one network to another. One possibility is, for example, to simulate the updating of the server file, hence making the resident *WkNet* process and also the WIBU-BOX superfluous.
- The cancellation of the copy protection within a network (i.e. a disposal of the slot limitation). This may, for example, be accomplished via sabotaging programs that increase the number of free slot allocations, hence simulating free slots for an application.

An unprotected server file which, in addition is described in complete detail within the scope of this documentation, may be manipulated with little effort to cancel the *WkNet* copy protection. An additional coding of the server file at critical locations may prevent this. This coding rests on a WIBU BOX sequence and through an additional program-specific algorithm, specified by the software producer. With both methods, all data that is stored in the critical server file locations is coded by the application to be protected and the WkNet process before being written on the file. The data decoded upon being read out of the file by the application.

A further protection enhancement is the delayed coding and decoding of data via the WIBU-BOX directly. Since it is not possible to simulate the complexity of the WIBU-BOX with software, an exceptionally high degree of protection is warranted.

WkNet Configuration

WkNet must not be explicitly incorporated in just one application, but the whole system must be configured. This configuration is confined to the following *WkNet* sections:

- **Server file**. The server file is created by the program **WKCRYPT** (see WIBU-KEY User's Guide). For this purpose, the maximal number of possible slots and the maximal buffer length for the coding via the WIBU-BOX must be specified.
- **Server process**. The server process (**WKSVW32** for Win32 or **WKSVNW** for Novell Netware) accesses the server file using the configuration data specified in the WIBU-KEY Control Panel Applet (see WIBU-KEY User's Guide).
- **Hardware WIBU-BOX**. In the WIBU-BOX, two entries must be made with the producer-specific Firm Code: The User Code, which is used for the verification of the protection effectiveness, and a second User Code with a higher number than the other User Code, and through which the maximal number of simultaneously running copies within the network is defined. This is determined by the difference between the two User Codes.

The complete communication occurs over the server file within a specific interval that is defined by the software producer. This interval must be at least 10 seconds and can extend up to a number of hours. In general, the following holds: The smaller the time interval, the higher the protection (and also the network and system loading). In general, a value of between 30 seconds and 1 hour is normal. As a rule, one can work with protected software copies at least twice as long as this time interval specifies, without the necessity of the incorporation of a WIBU-BOX within the network.

The decryption of the data, encrypted by the WIBU-BOX and controlled via the specified Selection Code, is handled on the protected application side with a 32 byte sequence which is created by **WKCRYPT** and integrated into the application. If the Selection Code is 0, this sequence may be omitted, but as a result, the protection quality of *WkNet* is reduced.

The encryption process specified for *WkNet* must also be integrated into the application to be protected in the same manner. Moreover, a 32 byte sequence must be integrated into the application in order to return the encoding —which was conducted directly by the resident *WkNet* process via the WIBU-BOX-to its original state. This 32 byte sequence is either generated by WKCRYPT in the form of a table of constants for the incorporation in the source code, or directly encoded in the completed DOS EXE file with **WKCRYPT** and the option **/CX**. In both cases, the same Selection Code is used, which is also specified by the resident WkNet process.

The maximal number of running copies within a network is defined by two values:

- The number of slots in the server file which were defined upon creation with the program **WKCRYPT**. This value is a maximal value that cannot be exceeded. *WkNet* supports up to 255 slots and hence, up to 255 running copies of given software within the network.
- The difference between the basic User Code which is used for the software protection and a further User Code which is registered in the same WIBU-BOX, together with the same Firm Code. Should a number of such User Codes be present in the WIBU-BOX, the smallest value that happens to be larger than the base User Code is employed. If the difference is smaller than the number of slots specified by **WKCRYPT**, all subsequent slots are continually occupied and can therefore not be used for program access.

WKNET is configured by WKCRYPT to 10:13. Should the system incorporate a WIBU-BOX with the entries 10:11, 10:13, 10:20, 10:30 and 195:14, WKNET reduces the maximal number of users to 7, since the entry 10:20 will be drawn on for the difference calculation. The entry 10:11 is too small, 10:30 is too large. The entry 195:14 possesses a different Firm Code.

As a result of the definition of the maximal number of copies, which may run within a network via the entry in the WIBU-BOX, this value can be easily adapted and also changed at a later date for the customer via a corresponding re-programming of the WIBU-BOX.

Index

Although nobody's perfect, WIBU-SYSTEMS always endeavors to attain the highest possible standards. Should you nevertheless happen to discover any errors in this documentation, or indeed, in a program, then please contact us. This naturally holds for any suggestions for improvement and criticism.

Should you happen to feel that any one of our products falls short of your expectations, please invest a few seconds, and briefly define the problem in the following space:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

This may be mailed to us at the following address:

WIBU-SYSTEMS AG, Rueppurrer Strasse 52-54
76137 Karlsruhe, Germany

Call us or send an email:

Phone: **+49-721-93172-0**
Fax: **+49-721- 93172-22** Web Site:
Email: **support@wibu.com**
http://www.wibu.com