Chapter 1. Introduction

This chapter provides an introduction to TimeLOCK 3.0. The topics that will be covered in this chapter are:

What is TimeLOCK?

Preview Software's TimeLOCK™ features an elegant, easy-to-use solution for software publishers and resellers wishing to distribute their applications over the Internet. The TimeLOCK Client Builder converts any Windows 95 or NT application into a client that can communicate with a TimeLOCK-enabled server to process on-line purchases. In a matter of minutes, software publishers and resellers can prepare applications for electronic distribution. Once this process is completed, the application can be posted to a Web site for downloading. Users can freely download the TimeLOCKed application, but must purchase it (using an electronic ordering system) before it is permanently unlocked.

TimeLOCK is the only software utility of its kind that can be used at any level of the software distribution chain. With TimeLOCK, publishers, distributors and resellers can reach a wider audience of prospective customers while minimizing their costs of marketing and selling. Sales can be tracked through each merchant and profits allocated accordingly.

Preview's TimeLOCK offers a fully tested and Microsoft approved method for electronic software distribution (ESD) that includes a facility for secure electronic funds transfer. TimeLOCK uses RSA public key encryption to prevent unauthorized use or tampering with your application. It is no longer necessary to create demonstration versions. TimeLOCK provides a secure mechanism to allow customers to try your software for a limited number of times before purchasing. At the end of the trial period, users can purchase on-line and the application will be unlocked immediately. Users avoid the hassle of having to perform a new installation.

TimeLOCK can also be used for electronic distribution without a trial period. For applications that will be used only once by most customers, publishers will not want to offer free trials. In such cases, it is still useful to TimeLOCK the application, because end users can download the product *before* purchasing. If the download fails, no customer support is required—the customer can just download another copy.

No matter which means of distribution is chosen, or what purchase plan is selected, TimeLOCK provides a secure means of implementing and managing the purchase transaction. TimeLOCK gives you control over how your application is made available to the end user, and does not interfere with the normal functioning of your application. TimeLOCK does not take control over the application but concentrates on maintaining a secure distribution environment.

The TimeLOCK Client Builder is licensed for use with one software title for sale through a single merchant for a period of one year. (Please refer to your license agreement for details of the terms and conditions of your license.) On-line purchases require the TimeLOCK Transaction Server software, which is a separate product. Transactions can be processed through third party clearinghouses operating the TimeLOCK Transaction Server or by purchasing the server software from Preview Software.

Key Concepts

To use the TimeLOCK Client Builder, it is helpful to understand a few key concepts relating to electronic distribution channels and purchase methods.

Electronic Software Distribution Channels

As with physical distribution, there are several different channels for electronic software distribution (ESD). In some cases a publisher may want to avoid retail distribution costs and sell directly to end users. Often, however, publishers can reach a wider market by offering their products for sale through on-line resellers. In some cases, a publisher may want to use a distributor to manage large numbers of resellers. Most publishers will use a combination of these distribution channels.

TimeLOCK accommodates any model for performing electronic software purchases, ensuring the security of the transaction, and setting the conditions for trial use of the product. Merchant and distributor information can be entered by different parties at different points in time. Security mechanisms prevent unauthorized tampering with the terms and conditions set by the publisher for sale of its application.

The following terms are used to describe all parties involved in the development, distribution, and sale of software.

§ ***Publisher*** or ***Developer***: The person or business who builds the software application.

§ ***Distributor***: The distributor maintains relationships with the publisher and the merchants who have "electronic storefronts" or other means of selling software products. The distributor delivers software products to merchants.

§ ***Merchant***: The merchant maintains an "electronic storefront" or other means of selling software products.

§ ***Clearinghouse***: The clearinghouse manages all transaction processing functions (e.g., fraud screening, credit card processing, directing payment to appropriate banks.) At the time that the customer decides to make the purchase, a message is automatically sent to the clearinghouse, which processes the payment and unlocks the application for the customer. The clearinghouse then notifies the interested parties (*e.g.*, publisher and distributor) that the sale was made.

§ ***Channel Partner***: A vendor who markets, distributes, or supports a product developed by a software manufacturer. A channel partner is usually one of the following: distributor, merchant, or clearinghouse.

There are two types of Electronic Software Distribution:

§ **Direct Distribution**: In this case, the publisher assumes the roles of the publisher, distributor, and merchant. For example, publishers offering their applications for sale from their own Web site are engaged in direct distribution.

§ **Multi-Tier Distribution**: In this case, the publisher uses one or more channel partners to distribute or sell to end-users. For example, when a publisher offers its product for sale through an on-line reseller, the on-line reseller may act as merchant or distributor.

Note: In this manual, we will refer to the most general case of multi-tier distribution where the publisher, distributor, merchant, and clearinghouse are distinct organizations. Should your situation be different (*e.g.*, if you are a merchant who deals directly with the publisher to handle all distribution of the product) please make the necessary substitutions when reading through the examples.

Purchase Methods

Try Before You Buy: You can turn your application into a fully functional trial application by integrating your application with TimeLOCK. A customer may use your TimeLOCKed application, free of charge, for a period of time that you specify. If the customer does not purchase the application before the trial period expires, they will continue to see a marketing message with an option to purchase, but will not be able to run the application. If the customer chooses to purchase the application, they will receive a unique key to deactivate TimeLOCK and "unlock" the software.

One of the benefits of using TimeLOCK to distribute your application electronically on a Try Before You Buy basis is that the presentation of the Client (or End-User) Interface is managed by TimeLOCK. The Client Interface supports options for purchasing the software electronically and presenting trial information. As a developer, you need not worry about creating dialogs, presenting usage meters or displaying any trial-related information to the end user when the application is run during the trial period.

Purchase Only:. TimeLOCK can easily be used to enable applications to be purchased over the Internet or by phone, fax, or e-mail. As with Try Before You Buy, TimeLOCK manages the Client Interface for you. You need not worry about creating dialogs or writing code for establishing transactions with a clearinghouse. TimeLOCK takes care of all of these functions and makes sure that the purchase is made in a secure environment.

Example of an Application using TimeLOCK

To illustrate how TimeLOCK may be implemented, we will use the following example scenario throughout this document:

§ **Publisher:** ***Dynasoft*** is the publisher of a product called ***Blackjack Pro***.

§ **Distributor:** Dynasoft has two distributors for Blackjack Pro, one of whom is *Electronic Software Distribution* or ***ESDist***.

§ **Merchant:** ESDist signs up several merchants, one of whom is ***GameWare.com***.

§ **Clearinghouse:** GameWare.com contracts with a clearinghouse called ***InterTrans*** to make purchases.

We will be considering the case where Blackjack Pro is available for trials of 30 days or 100 uses. Customers may download trial copies from GameWare.com's Web Page. They may purchase the software at any time, via a secure Internet transaction, voice/fax, e-mail, or mail.

Figure 1-1illustrates the relationship between the organizations involved in developing and marketing Blackjack Pro.

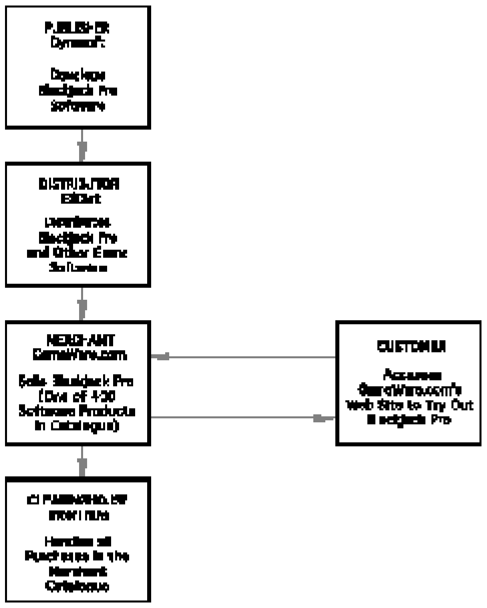

PUBLISHER
Dynasoft
Develops Blackjack Pro Software
DISTRIBUTOR
ESDist
Distributes Blackjack Pro and Other Game Software
MERCHANT
GameWare.com
Sells Blackjack Pro (One of 400 Software Products in Catalogue)
CUSTOMER
Accesses GameWare.com's Web Site to Try Out Blackjack Pro
CLEARINGHOUSE
InterTrans
Handles all Purchases in the Merchant Catalogue

Figure 1-1 Blackjack Pro Electronic Distribution Model

How Does TimeLOCK Work?

Once you have developed a Windows 95 or Windows NT application, TimeLOCK is used to prepare the application for secure electronic distribution and electronic commerce. This is done using one of two TimeLOCK integration methods:

§ **Instant Integration** is accomplished by entering information into a series of dialog boxes in the TimeLOCK Client Builder. The Client Builder then modifies the program .executable (*.exe) file. No source code changes are necessary.

§ **Programmed Integration** requires modifying the application's source code using the TimeLOCK API. The TimeLOCK Client Builder prepares skeleton source code and other files to simplify the process.

Both TimeLOCK integration methods are described in detail in *Getting Started* to help you decide which approach is appropriate for your application and software distribution scheme. The Client Builder, described below, is used to select the TimeLOCK integration method, desired security mechanisms, any trial parameters, and payment method.

Once the publisher and all channel partners have used the Client Builder to integrate TimeLOCK functionality, the software is ready for distribution. TimeLOCK automatically:

§ Enables on-line purchase of the software

§ Protects the application from tampering and unauthorized access

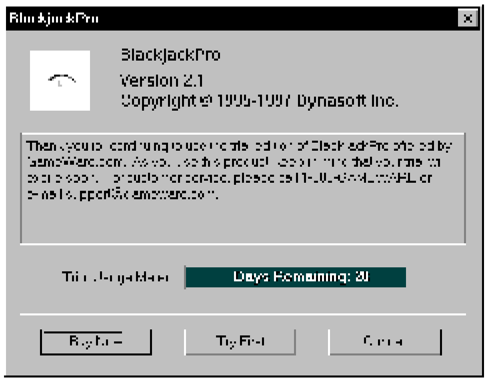
§ Manages any trial period

§ Keeps statistics on product usage

With Instant Integration, information given in the TimeLOCK dialog screens (discussed in the section on the Client Builder) will be used to generate the Client Interface seen by the end user. Developers using Programmed Integration may use appropriate TimeLOCK API calls to create their own, custom Client Interface.

In Figure 1-2, we show an example of an application that has been prepared for ESD using the Instant Integration method. Note that TimeLOCK automatically generated the screen and its contents. The publisher did nothing more than specify product information and trial parameters when running the Client Builder. TimeLOCK's Client Interface is discussed in more detail on the following pages and in the chapter entitled *Client Interface*.

Note: We refer to applications that have been integrated with TimeLOCK as *TimeLOCKed* or *wrapped* applications.



BlackjackPro
Version 2.1
Copyright © 1995-1997 Dynasoft Inc.

[illegible]

Figure 1-2 Example of Blackjack Pro Distributed for Trial Use

Blackjack Pro appears with a welcome screen displaying information on the customer's trial. The customer may elect to Purchase the application at this time, or simply start using the application on a trial basis.

Should the customer decide to purchase the application, TimeLOCK will take them through a series of dialog necessary to purchase the software. The TimeLOCK Transaction Server unlocks the application once the purchase has been completed.

If, for some reason, the customer is unable to make an Internet connection, the application can be purchased by phone, fax, or e-mail. For these transactions, it will be necessary for the customer to manually enter an unlocking code. This is described in detail in the *Client Interface* chapter.

The *Getting Started* chapter will further explain the use and role of TimeLOCK integration and the Client Builder. *The Client Builder* chapter walks the reader through each screen of the Client Builder using an example application called Blackjack Pro.

The Client Builder

The TimeLOCK Client Builder uses a Wizard to guide you through the steps necessary to integrate Try Before You Buy (TBYB) and electronic commerce capabilities into your application. There are five basic steps in the integration process:

1 **Start**: Specify information on you, the Client Builder user, and the distribution model.
2 **Publisher**: Specify publisher information including trial and purchase options.
3 **Distributor**: Specify distributor information.
4 **Merchant**: Specify merchant information including purchase and support information.
5 **Integration**: Specify information for the chosen TimeLOCK integration method.

Each of these steps corresponds to a tab in the Client Builder. A brief overview of the purpose of each step is provided below. The information that you enter will depend upon how you intend to distribute your product. Detailed instructions are provided in the *Getting Started* and the *Client Builder* chapter. *Getting Started* will give you information on all steps involved in the wrapping process. The *Client Builder* chapter will take you through each of the screens in the Client Builder.

Prerequisites for Client Builder
Client Builder Steps

Prerequisites for Client Builder

In order to use the Client Builder, you will need to obtain the following files from Preview Software or your clearinghouse. Please see the section on *Security, Key Files and Encryption* in this chapter for more information on the purpose of these files.

§ **Product Key File (*.prd)**: The Product File identifies the application. You must have the product file and associated password to TimeLOCK the application.

§ **Privilege Key File (*.prv)**: The Privilege Key File establishes access privileges for use of the Client Builder. You must have a privilege file and know its password in order to use the Client Builder. The Client Builder uses information in this file to determine what type of information you are authorized to change (for example, whether you are a publisher, distributor, or merchant). **Note**: publishers have access privileges to modify all information in the Client Builder.

§ **Clearinghouse Key File (*.clr)**: If you are using a clearinghouse to process Internet purchases and generate unlocking keys, you will be asked for this file. The clearinghouse file (*.clr) specifies the clearinghouse that end users will connect to when making purchases over the Internet.

The Client Builder will create up to four files that are used in creating the TimeLOCKed application:

§ **License Information (*.lif) File**: is a file used to store the options selected in the Client Builder. The License Information file is modified each time the Client Builder is run. This file should have the same name as the Product Key File. You will name this file and save it in your TimeLOCK directory. See the section on *Security* for additional information on the License Information file.

§ **BOB (*.bob) File**: BOB stands for Bag Of Bits. In a multi-tier distribution environment, your application executable and related files are assembled into a BOB that can be passed securely from publisher to distributor to merchant. The BOB file is not produced if you supply all merchant and distributor information when you first use the Client Builder. See the section on *Security* for additional information on the BOB file.

§ **License (*.lic) File**: contains all license information on the TimeLOCKed application. This file will be distributed to the customer along with the TimeLOCKed application software.

§ **Setup (*.exe) File**: is a self-extracting executable used to create the installation file for the TimeLOCKed application that can be distributed to customers. In most cases, it will be necessary for you to use your installer software to create a new setup.exe file that incorporates the TimeLOCKed executable file in place of your application's original executable file.

Client Builder Steps

1. **Start**

§ Specify Privilege file (*.prv) to determine which access rights you will have in use of the Client Builder.
§ Specify License Information file (*.lif) being created or modified.
§ Publisher may define distribution method (Purchase Only or Try Before You Buy). The distributor and merchant may see what Distribution method is being used.
§ Choose an Integration method (Instant or Programmed).

2. **Publisher**

§ Provide information about the publisher.
§ Define trial parameters for Try Before You Buy distribution.
§ Select security options.
§ Provide Trial User and End User License Agreements.

3. **Distributor**

§ Provide information on the distributor (if any).
§ Specify the Product File for this application. This file is obtained from Preview Software or your clearinghouse.
§ Select purchase methods (Internet or Phone/Fax/E-mail) available to customers.
§ Identify clearinghouses authorized to process transactions. Clearinghouse information is supplied through .clr files.

4. **Merchant**

§ Provide information on the merchant and customer service contacts.
§ Set product price and expiration date.
§ Provide text for messages to end-users.

5. **Integration**

§ Integrate TimeLOCK functionality into the application using Instant or Programmed Integration.   The screens and required information will vary depending on what type of distribution model and integration method is being used.
§ Generate new application executable. If your are using Instant Integration, the new application executable will be generated for you by the Client Builder.   A backup of the original executable will be made for you. If you are using Programmed Integration, the Client Builder will generate source code and you will need to recompile your application.
§ In some cases, you will use the Client Builder to create a setup file for customers.

The TimeLOCK Client End-User Interface

The TimeLOCK Client Interface is the set of screens and dialog boxes seen by a customer when purchasing or running a TimeLOCKed application.   The appearance of the Client Interface depends on which TimeLOCK integration method is chosen:

§ Instant Integration uses information specified in the Client Builder to create the Client Interface. If Instant Integration is used, the Client Interface may not be customized (outside of bitmaps and user messages) and will appear almost exactly as in Figure 1-2 in How Does TimeLOCK Work?

§ Programmed Integration allows the publisher to use the default TimeLOCK Client Interface as in Figure 1-2 in How Does TimeLOCK Work?, or to create a custom version using the TimeLOCK API.

Default Purchase Dialogs

Default Purchase Dialogs

The default Client Interface generated by TimeLOCK consists of a Main Dialog that gives Product and Company information and a series of Purchase Dialogs known as the Purchase Wizard. These dialog boxes gather information from the customer to complete the purchase transaction.

§ **Main Dialog**: The Main Dialog generated by TimeLOCK is the first screen displayed to the customer when starting the TimeLOCKed application. This screen gives product details, trial information (if applicable), and an option to purchase the software. Figure 1-2 shows an example. If the TimeLOCKed software has been purchased, the Main Dialog will appear without an option to purchase.

§ **Purchase Wizard**: If the user decides to purchase the application, the Purchase Wizard will provide appropriate instructions. These instructions are defined in the merchant section of the Client Builder. Figure 1-3 shows the first screen of the Purchase Wizard generated for our example. The *Client Interface* chapter will take you through all the screens of the Purchase Wizard.

Figure 1-3 Purchase Wizard for Blackjack Pro Example

The customer may purchase a TimeLOCKed application using the Internet or phone, fax, or e-mail. The clearinghouse, using the TimeLOCK Transaction Server or the TimeLOCK Key Module (see related documentation for information on these tools), generates a unique unlocking key for the customer. If a purchase is made over the Internet, the unlocking key is sent to the customer electronically. If a purchase is made using the phone, fax, or e-mail, the customer is given an unlocking key over the phone. When either the electronic or phone unlocking key is applied, TimeLOCK performs the necessary environment update so that the next time the application is run, no trial information will appear.

Note: A phone unlocking key is not as secure as an electronic unlocking key. Preview Software suggests that a phone unlocking key be used temporarily until an electronic key can be provided.

The *Client Interface* chapter walks you through each screen of the Client Interface generated by Instant Integration so that you may see precisely how it works.

Security

An important concern when developing an application for electronic distribution and electronic commerce is security. You are providing a fully functional version of your product, and want to be confident that the locking mechanisms are adequate to safeguard your software from unauthorized use. In addition, in creating the TimeLOCKed software, you must ensure that the application is not modified inappropriately by an unauthorized user of the Client Builder. TimeLOCK has a number of different security options to protect your software.

TimeLOCK uses a number of hidden tags and markers to protect the TimeLOCKed application from unauthorized use. Along with the unlocking keys described earlier, these provide a high degree of security for your application. However, TimeLOCK uses *RSA public key encryption* to ensure that each step of the TimeLOCK process—from development to implementation—is protected from outside tampering. TimeLOCK handles all encryption keys and developers do not need to understand how these security mechanisms are implemented. This section provides a brief overview of TimeLOCK security. For a complete description of these security mechanisms, see the white paper, *TimeLOCK Security of Electronic Software Distribution*.

Encryption and Key Files

Key Distribution

Encryption and Key Files

There are two areas where encryption is used to protect the TimeLOCKed application from unauthorized use or modification:

§ Preventing an unauthorized user from using a TimeLOCKed application

§ Ensuring that only a valid publisher, distributor, or merchant defines the parameters for TimeLOCKing a particular application

The above is done by encrypting License (*.lic) and License Information (*.lif) files and by generating Key Files using encryption technology.

Note: Key Files are generated using RSA Data Security technology. Only Preview Software and its clearinghouse partners are authorized to generate these key files. The following files are provided by Preview Software. Please see the *Key Distribution* section in this chapter for more information.

Preventing an unauthorized user from using a TimeLOCKed application:

There are two files that use encryption to protect an application from unauthorized use.

§ **License File (*.lic)**: TimeLOCK stores information on the TimeLOCKed application, including any trial parameters and security mechanisms, in a License file (*.lic) on the customer's computer. The License (*.lic) file is encrypted using RSA Data Security encryption technology to protect it from tampering. Thus, for example, a user can not directly edit the License file to change it from a trial to a purchased copy.

§ **Product Key File (*.prd):** Encryption is used to generate and protect the unlocking keys. This prevents unauthorized users from using unlocking keys with software that has not been purchased. Thus, when a trial has expired or someone is trying to use the product illegally, the product ceases working and they are "locked out." Only when the customer has purchased the software or obtained an extension from the merchant, are they given an unlocking key to use the software and update the License File.

Ensuring that only a valid publisher, distributor, or merchant defines the parameters for TimeLOCKing a particular application:

The Client Builder is also protected from unauthorized access. This is important to ensure that only an authorized publisher, distributor, or merchant changes information on the TimeLOCKed application. Encryption is used to generate key files that verify the user of the Client Builder. These files are used to determine what level of privileges a user has to modify the License Information (*.lif) for the TimeLOCKed application:

§ **Privilege File (*.prv):** The publisher and its channel partners must obtain a privilege file from Preview Software in order to use the Client Builder. This file, *.prv, will protect anyone from altering the License Information and License files who does not have proper authorization. This file also verifies what level of privileges the user has. For example, it will allow a user with distributor and merchant privileges to change distributor and merchant information. That user may not change publisher information.

§ **License Information File (*.lif)**: The Client Builder stores information in an encrypted License Information file (*.lif). This file is encrypted to protect it from tampering. In addition, the Client Builder prompts the user for the password to the License Information file (*.lif) to verify that the user may modify this information.

§ **Bag Of Bits or BOB File (*.bob):** The Bag Of Bits (BOB) file is an encrypted binary file that is used only in multi-tier distribution. Its purpose is to ensure the integrity of the License Information (*.lif) file and TimeLOCKed executable (*.exe) file as they are given to channel partners. That is, in multi-tiered distribution, the publisher must provide a BOB file to protect the key, license, and application files from tampering as they are given to the distributor or merchant. If the BOB file does not exist, or has been tampered with, the Client Builder will give the user an appropriate message. The user must then request the BOB file from the publisher.

§ **Product Key File (*.prd):** A publisher or channel partner may use TimeLOCK with more than one application. Each application is assigned a unique product code to prevent unauthorized modification to that application. The product file must be provided in the Client Builder to prevent an unauthorized publisher, distributor, or merchant from TimeLOCKing an application for which they have not been granted access rights.

The following summarizes where keys are generated and accessed in the distribution process.

§ **Product Key (*.prd)**: A private key used to sign products stored in the *.lif and *.lic files. There is one key per product. This key is generated by Preview Software.

§ **Publisher Key (*.prv)**: Enables the publisher to use the Client Builder to edit license information stored in the *.lif file. It is stored in the *publisher*.prv file and accessed when the Client Builder is run. This key must be obtained from Preview Software.

§ **Distributor Key (*.prv)**: Enables the distributor to use the Client Builder to edit license information stored in the *.lif file. It is stored in the *distributor*.prv file and accessed when the Client Builder is run. This key must be obtained from Preview Software.

§ **Merchant Key (*.prv)**: Enables the merchant to use the Client Builder to edit license information stored in the *.lif file. It is stored in the *merchant*.prv file and accessed when the Client Builder is run. This key must be obtained from Preview Software.

§ **Clearinghouse Key (*.clr)**: Used to secure communications between the TimeLOCKed application and the clearinghouse. This ensures that purchases may be made securely. This key is stored in the *clearinghouse*.clr file. This key must be obtained from Preview Software.

§ **Electronic Unlocking Key**: A public key used for verification that enables the user to use the product after purchase. This key is typically generated by the clearinghouse, or whomever is assigned the task of distributing electronic unlocking keys, and stored in a file on the customer's computer after purchase. This key is generated using TimeLOCK's Transaction Server product. See related documentation or contact Preview Software for more information on this product.

§ **Phone Unlocking Key**: An unlocking key generated and distributed by the clearinghouse over the phone. It will be stored in the customer's License (*.lic) file once entered. This key is generated using TimeLOCK's Key Module as discussed in the *Key Module* chapter.

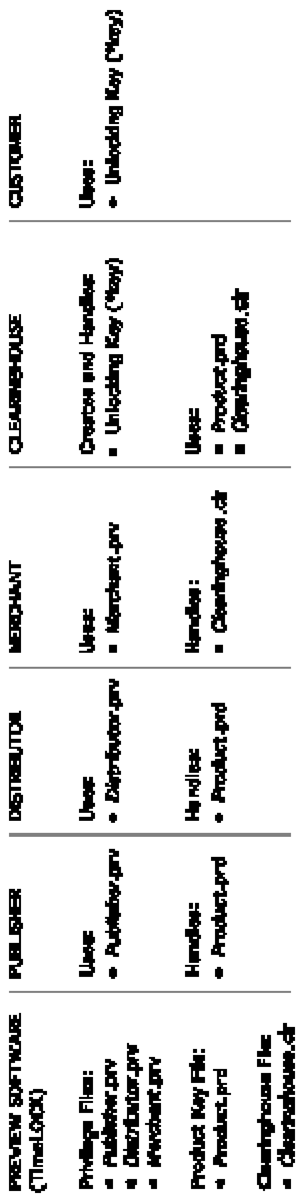
Key Distribution

Key Distribution

Figure 1-4 shows which organization in the Electronic Distribution Model, generates and uses each of the above keys.

Private keys are obtained from Preview Software. Preview Software will verify the identity and access privileges of the person making the request. The keys are then transmitted via secure channels. All keys are password protected and the owner is required to change the password generated by Preview Software before using the key files. Private keys and passwords must be carefully maintained and secured by their owners.

Note: For security reasons, keys have a lifetime of one year. It is the responsibility of the owner to request new keys from Preview Software.



PREVIEW SOFTWARE (TimeLOCK)	PUBLISHER	DISTRIBUTOR	MERCHANT	CLEARINGHOUSE	CUSTOMER
Privilege Files: ▪ Publisher.prv ▪ Distributor.prv ▪ Merchant.prv	Uses: • Publisher.prv	Uses: • Distributor.prv	Uses: ▪ Merchant.prv	Creates and Handles: ▪ Unlocking Key (*key)	Uses: • Unlocking Key (*key)
Product Key File: ▪ Product.prd	Handles: • Product.prd	Handles: • Product.prd	Handles: ▪ Clearinghouse.clr	Uses: ▪ Product.prd ▪ Clearinghouse.clr	
Clearinghouse File: ▪ Clearinghouse.clr					

Figure 1-4 Key Distribution

Security Mechanisms with TimeLOCK integration

There are a variety of optional mechanisms that may be used to secure an application distributed electronically.   These options are used for:

§ Copy protection
§ Generating unlocking keys
§ Securing the executable from tampering

These mechanisms are selected when running the Client Builder and are described in detail in the *Client Builder* chapter.

What is New in TimeLOCK 3.0

TimeLOCK 3.0 represents a major upgrade to the TimeLOCK product suite. Developers can take advantage of a number of new techniques to ensure that their software applications are delivered in a secure yet flexible environment.

Electronic Commerce

§ Instant integration allows you to integrate secure on-line ordering capability into applications without modifying source code.
§ The Purchase Wizard allows connections through the Internet and by phone, fax, or e-mail. This allows customers to purchase software at any time, quickly, easily, and securely.
§ Direct and Multi-tiered distribution models are supported. This means purchase transactions can be processed through most clearinghouses or using the TimeLOCK 3.0 Transaction Server on your own web site.

Wrapper Technologies

§ Purchase Only Wrapper (Digital Delivery Envelope). Users download software before purchasing, but are required to purchase before they may use the application.
§ Try Before You Buy (TBYB) Wrapper. Users may use an application on a secure trial basis. This is more effective than creating a demo to sell your product.
§ Trials may be restricted using five different measurements: number of executions, number of days, expiration on specific date, cumulative usage time, or unlimited trial.
§ Flexible Wrapper tool: A single tool, the Client Builder may be used for either Purchase Only or Try Before You Buy distribution.
§ Instant Integration does not require any modification of source code. The Client Builder takes care of the TimeLOCK integration from start to finish. This enables all members of the Electronic Software Distribution model to specify distribution parameters.
§ The developer has complete control over the end-user interface with optional API calls.
§ Programmed Integration may be used to automatically generate fully commented source code in C/C++, Visual Basic, or Delphi.

Security

A number of new security mechanisms have been implemented to prevent both unauthorized creation of a TimeLOCKed application and unlocked use of an application:
§ Industry-standard RSA encryption algorithms employed.
§ TimeLOCK wrapper is tamper resistant.
§ All trial status information is encrypted and tamperproof.
§ Compression of executable file retards reverse engineering efforts.
§ Executable is secured against tampering; hackers can't jump past the locking code.
§ Unlocking keys are unique and can't be forged.
§ Communication between client and server is secure against eavesdropping and man-in-the-middle attacks.

What is in This Manual

The following chapters will describe in detail the TimeLOCK tools used by Developers to create TimeLOCKed applications. Specifically:

- Chapter 2, **Getting Started**, describes how to install TimeLOCK and get started creating TimeLOCKed applications. This chapter will also describe the two methods of integration: Instant and Programmed.
- Chapter 3. The Client Builder, will walk you through each screen and option of the Client Builder. Examples are given to describe the procedures for using TimeLOCK under different electronic distribution scenarios.
- Chapter 4. Reset Wizard, shows how you may reset the TimeLOCKed application for use in testing the application before release.
- Chapter 5. Troubleshooting, lists the steps to go through when problems occur running the TimeLOCKed application.
- Chapter 6. Client Interface, will walk you through each screen and option of the Client Interface generated by Instant Integration. This will help you to test your application's end-user interface.
- Chapter 7. The Key Module , explains how the Key Module is used to generate Phone Unlocking Keys for a TimeLOCKed application.
- Chapter 8. Using the TimeLOCK API , explains and gives examples of how developers may use the TimeLOCK API for Programmed Integration.
- Chapter 9.Function Reference , defines all the functions in the TimeLOCK API and gives examples.
- The Glossary includes definitions for all TimeLOCK terms.
- **Index**

Chapter 3. The Client Builder

In this chapter we will describe the Client Builder and how it is used to integrate TimeLOCK into your application. The following sections cover all aspects of the Client Builder:

The “How To” section will walk you through every screen that may be encountered when using the Client Builder. Keep in mind that only a small subset of these screens will apply to your use of the tool.

What is the Client Builder?

The publisher, distributor, and merchant use the Client Builder to integrate an application with TimeLOCK functionality. The Client Builder will walk you through the steps involved in integrating an application with TimeLOCK to meet the needs of your product release, evaluation program, and its associated marketing plan. The publisher, merchant, and distributor need only answer a few simple dialogs in order to complete the integration. Please refer to the *Getting Started* chapter to see what information will be required before proceeding with this section.

Information entered in the Client Builder is used by TimeLOCK to create the following:

§ **License and License Information Files**: The License Information (*.lif) and License (*.lic) files are used to ensure that a customer uses a TimeLOCKed application only under a valid trial or purchase agreement. The Client Builder will create or modify license information used to verify purchase or trial information based on product and company information, security options, and any applicable trial parameters. Please see the *Introduction* and *Getting Started* chapters for more information on this file.

§ **TimeLOCK Integration**: The Client Builder will create the TimeLOCKed executable for Instant Integration or generate the source code for Programmed Integration. You will need to supply the location for some of your application files. Please see the *Getting Started* chapter for details on what types of files you will be prompted for.

§ **Client (End-User) Interface**: The merchant must define user messages that will appear in the Client, or End-User, Interface. These messages are used to display trial information or to purchase the TimeLOCKed application.

In this chapter we will take you through each screen of the Client Builder to show you exactly how it is used to create a TimeLOCKed application. Not all of the screens will appear when you use the Client Builder. Please see the *Getting Started* chapter for what type of information will be required.

Security of the Client Builder

In order to use the Client Builder, you must be authorized to change information in the License (*.lic) and License Information (*.lif) files. Each organization involved in electronic distribution (publisher, distributor, and merchant) is given a privilege file (*.prv). This file contains information on which parameters you may modify. You must provide the location of this file and an access password in order to proceed. Options that you are not authorized to change will be grayed out.

Since the Client Builder modifies License Information, you must be authorized to modify the License Information (*.lif) file. The Client Builder will ask you to provide the password for the License Information (*.lif) file for your application. Options that appear grayed over and filled in cannot be changed at your privilege level. Should you need to change these parameters, contact the distributor or the product publisher for support. Please see the *Introduction* and *Getting Started* chapters for more information on security mechanisms and prerequisite files.

When To Use Client Builder

The Client Builder is used at different points in the integration process. Depending on who is performing the roles of publisher, distributor, and merchant the Client Builder may be used more than once. Of course, you may use the Client Builder at any time to simply verify the contents of an existing License Information file. In this case, please use the **Close** button to exit the Client Builder.

How to Use the Client Builder

The Client Builder is easy to use and can be completed very quickly. Following this section we will provide a complete description of each screen of the Client Builder. However, the information required to complete a session will vary according to your distribution model. If you are a publisher who markets and sells products directly, you will need to use all of the screens. If not, for example you are a distributor, you may need to use only a few screens. For this reason, we suggest you read the *Example Scenarios* section in the *Getting Started* chapter for examples of how the Client Builder is applied under a number of different distribution scenarios.

Each tab in the Client Builder corresponds to an organization involved in Electronic Software Distribution: publisher, distributor, and merchant. You will perform those tasks appropriate to your organization's role. Once you have completed the screens under your level (e.g., Dynasoft completes all screens associated with publisher) select the **Finish** button to complete the Client Builder.

Once the Client Builder has finished, one or more of the following files are created:

§ **License Information File (*.lif)**: is distributed with the TimeLOCKed application to all channel partners. This file will be used to create the customer's License (*.lic) file.

§ **License File (*.lic)**: is distributed with the TimeLOCKed application to all customers and contains information on the file and its purchase or trial status.

§ **Bag Of Bits File (*.bob)**: is used under certain distribution models to ensure the integrity of the License and License Information files. A BOB file is not necessary in direct, Try Before You Buy distribution environments. This file is used temporarily to perform the integration and will not be used or distributed to customers.

§ **Setup File (*.exe)**: will be used by the customer to install the TimeLOCKed application.

In the next several pages, we will guide you through all of the screens used in the Client Builder. Note that some of the screens displayed will depend on what options you choose for integration. For example, if you choose not to create a trial version of your application, none of the trial options will be enabled.

Common Options

There are a number of common buttons or fields that appear on most of the screens in the Client Builder. These options are described as follows:

§ **Back**: You may return to the previous screen by selecting the **Back** button. This is useful if you need to view or modify information that has been entered in previous screens.

§ **Next**: You may proceed to the next screen by selecting the **Next** button.

§ **Close**: You may exit the Client Builder at any time by selecting the **Close** button. You will be asked if you want to save changes to the License Information (*.lif) file before exiting.

§ **Help**: You may access the online help instructions, by selecting the **Help** button, to guide you through and help explain each of the steps. Some of the steps may not apply to your organization. You may skip these steps or leave certain options blank.

§ **Filenames**: Filename fields offer pull-down menus so that you may simply select from a list of previously used files rather than type in the complete pathname. The last used file is displayed in the filename field by default. You may also use the Browse box to select the filename.

§ **Show Me**: Whenever messages are entered in the Client Builder, you may select the **Show Me** button to see how these messages will appear in the Client Interface.

§ **Finish**: The last step of the Client Builder will have a **Finish**, rather than **Next**, option. Clicking on **Finish** will save all changes to the License Information (*.lif) file and end your session with the Client Builder.

§ **About**: Some screens have an **About** button that, when selected, will display information on the Client Builder.

Start the Client Builder

There are two ways to start the TimeLOCK Client Builder:

§ Double-click on the file `ClientBuilder.exe` in the TimeLOCK folder.
§ Choose **Run** from the **Start** menu and select `ClientBuilder.exe` from the TimeLOCK folder.

Client Builder Login

Because every user of the Client Builder has certain access privileges, identification is necessary before the Client Builder can proceed. The privilege level is stored in a security file with the prv extension (e.g., Dynasoft.prv). Options that the user has no authority to change will appear grayed over in the Client Builder. Options that you do not want to fill in may be left blank. Note that the Privilege (*.prv) file and access password are provided by Preview Software.

In this step, you are prompted for the name of the privilege file. Once you click **Next** you will be asked for the file Password.

Input Fields	Max Length	Description	Example
File name	255	*.prv file containing your access privileges for the Client Builder.	DynaSoft.prv
Access Password	5-12 characters	Used to determine privileges for editing the license file.	123abc8swe

User Information

Once you have entered a valid password, the User Information screen will appear with detailed information on your privilege level.

License Information File

The Client Builder will create a License Information (*.lif) file to store the results of the Client Builder. This file will be used by the Client Builder to create the customer License file (*.lic). In this screen, you are asked to name a new LIF file or load an existing one. You are also asked for an access password to verify the set of options you have authority to change. When you select **Next** you will be prompted to enter either:

§ A new password when creating a new LIF file.

§ The current password when loading an existing LIF file.

Input Fields	Max Length	Description	Example
File name	255 characters	License file that will be edited by the Client Builder.	Blackjack Pro.LIF
Access Password	5-12 characters	Used to determine privileges for editing the license file.	123abc8swe

Questions for Creating a New File

If you are creating a new License Information (*.lif) file, the following screen will appear asking for information on the distribution model and Integration method you wish to use:

(1) Do you want to allow users to try your application before purchasing? Select:

§ **Yes** if you wish to distribute your application on a Try Before You Buy Basis. You will be prompted for trial parameters in later screens.

§ **No** if you wish to distribute your application for Purchase only.

(2) How will you integrate TimeLOCK functionality into your application? Select:

§ **Instant Integration:** if you wish to integrate TimeLOCK functionality directly into the executable. Your application will use the End-User dialogs described in the *Client Interface* chapter.

§ **Programmed Integration:** if you wish to generate source code for the integration. This is desirable when you wish to embed your application or otherwise customize the interface shown in the *Client Interface* chapter.

Questions for Creating New File

(Instant Integration only)

If you are using Try Before You Buy distribution and Instant Integration, you will be asked if you can modify your setup files. This is important to create the installation file for the TimeLOCKed application. If you select:

§ **Yes**: When you are finished with the Client Builder, you must modify your setup script to include the License (*.lic) file and TimeLOCK DLL (tl30inj.dll).

§ **No**: When you are finished with the Client Builder, you must complete a few additional steps as described below.

If You Cannot Modify your Setup Script

If you can not modify your setup script, follow the steps below to complete the TimeLOCK integration.

1. You should have a setup script that includes your application files.

2. Run the Client Builder, continuing from this step. When you are finished with the Client Builder, you will get a message telling you to create an installation file using the new TimeLOCKed executable. Do not include the License (*.lic) file in this installation.

3. Run the Client Builder a second time and choose **Next** in the **Questions for Opening Existing File** screen. You will be taken to the **Installation Directory** screen.

4. Create a new setup executable file. This will incorporate the License (*.lic) file, TimeLOCK DLL (tl30inj.dll), with your own setup file.

Questions for Opening Existing File

Under some distribution models, the Client Builder must be run more than once to complete the integration. If you are loading an existing License Information (*.lif) file, you will be asked if you want to make any modifications to the integration. This screen will also appear when the distributor and merchant use the Client Builder in a multi-tier distribution environment.

§ **If you do not wish to modify information**, simply press **Next** to proceed with the next step of the Client Builder as determined by information in the License Information File.

If you wish to modify information, select:

§ **Edit & Modify license information**: to alter information previously entered in the Client Builder.

§ **Instantly Wrap the application EXE**: this option is made available when an executable has not been previously wrapped with Instant Integration.

§ **Create TimeLOCK self-extracting setup.EXE**: this option is made available when the distribution scheme requires creation of a new self-extracting executable.

Application Information

User: Publisher

Throughout the TimeLOCKed application, the TimeLOCK Client Interface will display information on the product and publisher. This information is supplied by the publisher in the Client Builder and will ultimately be displayed in the TimeLOCK Purchase Wizard or within standard message boxes in response to some user action.

Input Fields	**Max Length**	**Description**	**Example**
Publisher Name	70 characters	Publisher company name	Dynasoft Publishing
Product Name	N/A	Name of the product that has been integrated with TimeLOCK	Blackjack Pro
Version Number	N/A	Release number of the product	3.0
Copyright Year(s)	N/A	The Copyright Year is the year(s) in which the product was copyrighted	1995-1997
SKU	N/A	Used by publisher to uniquely identify the product	Blackj123

Trial Information

(Try Before You Buy only)

User: Publisher

If you selected Try Before You Buy as a distribution method, you will be asked to select trial parameters.

You must first define the distribution parameters for the TimeLOCKed product:

§ **Limited Trial**: The application will be TimeLOCKed for trial use, subject to the parameters set in the Trial Type panel.

§ **Unlimited Trial:** The application will be TimeLOCKed for trial use, but the trial will never expire. TimeLOCK will simply return Trial User State information perpetually until the application is purchased.

If you select Limited Trial, you will need to specify how long the trial will last. TimeLOCK supports four ways of measuring the trial:

§ **Number of Executions**: The number of times you wish the end user to have access to your application.

§ **Number of Days**: The maximum number of days the software is available for access after initial use.

§ **Exact Usage Time**: The exact INTEGER number of hours that your application should be available for CONSECUTIVE use. In other words, the actual usage time is recorded and checked each time the application runs. This is the most secure and accurate estimation of the time a user has spent with your application

§ **Expiration Date**: The date of expiration by month (1-12, where January = 1), the Day of the Month (1-31), and the year (entire number, i.e., 1996 *not* 96 or '96). You may use the arrows to set these values.

For optimum security and customer service, Preview Software recommends that a combination of trial measures be used.

Copy Protection Security Options

User: Publisher

The options highlighted in this screen depend on whether the product is distributed for Purchase Only or Try Before You Buy and whether you are using Instant or Programmed Integration.

Restore Original Application? This option appears when the application is packaged for Try Before You Buy distribution and Instant Integration is used. When an application is unlocked for permanent use, you have the option of restoring the original application as it was created before integration with TimeLOCK, or supporting a number of TimeLOCK security options. In this section you are asked which option you prefer:

§ **Yes**: When an Unlocking Key is supplied for the TimeLOCKed application, the original application will be restored on the customer's computer. All TimeLOCK functionality will be removed. No security options will be available and you may proceed to the next screen.

§ **No**: When an Unlocking Key is supplied for the TimeLOCKed application, the application will have any number of security mechanisms to prevent unauthorized use. You may now select the Security Options you want implemented in the unlocked application.

There are a number of mechanisms the publisher may use to protect the TimeLOCKed application from unauthorized use once it has been purchased. Note that each of these mechanisms involves a trade-off between increased security and customer support. These tradeoffs are described in this section.

Note: When this screen appears for the first time, none of the Security Options will be filled in. You must explicitly select **Yes** or **No** in order proceed to the next screen.

Copy Protection Schemes: This set of options appears when either the user has selected to *not* restore the original application (in above section) or Programmed Integration is used. The following schemes ensure that purchased applications cannot be copied to other machines.

§ **Software Binding**: An executable file copied from another machine will not run, because markers in the Windows Registry and hidden files will not be copied over.

§ **Hardware Binding**: An executable file copied from another machine will not run, because the computer's features are used to identify the purchased copy.

§ **Bind to Application**: An executable file copied from another machine will not run, because each executable has a unique marker, or brand, that will be changed when copied. The License file will check the brand of the executable and compare it with that in the *.lic file. Since the two are not equivalent, the executable will not run.

Note: Since binding occurs after the TimeLOCKed application has been installed, some virus checkers will signal an alarm to the user. In this case you must use Programmed Integration which allows you to use the TimeLOCK API to create a warning to customers to ignore this message.

If you elect not to use any of the above three options, be aware that a purchased application can be copied to another machine simply by copying the License (*.lic) file and executable. Note that if you support Try Before You Buy, a user could easily obtain a trial copy to use with the License (*.lic) file.

However, using copy protection schemes may create problems for customers who work in mobile environments. Specifically, copy protection prevents customers from moving the TimeLOCKed application from one machine to another without getting a new unlocking key. This can be a problem in today's portable world where customers may want to use a copy of the application on a laptop, home computer, or different location. In addition, if Hardware Binding is implemented, the executable will cease working whenever the user upgrades or modifies his hardware.

Securing the Executable from Tampering

It is almost impossible for even a sophisticated user to bypass TimeLOCK's authentication checks. However, if you wish to reduce the chance of this occurring even further, security codes may be used to prevent an unauthorized user from editing the executable as described below.

§ **Make application tamper proof**: When you select this option, the TimeLOCKed application's executable is given a security code that is also included in the License file. Before running an application, TimeLOCK will verify that the security codes match. If someone tampers with the executable, the security code for the executable changes and TimeLOCK will prevent the TimeLOCKed application from running.

Trial User License Agreement

User: Publisher

The software license agreement for a trial version will be stored in the License file associated with the TimeLOCKed application. You may enter the text of the agreement in this screen, or simply import the ASCII file containing the text of the agreement by clicking on the **Import** button.   A trial user will not be able to use the TimeLOCKed application without accepting the terms of the Trial User License Agreement.

Input Field	Max Length	Description	Example
Trial user license agreement	N/A	Text of license agreement for trial users	This is a trial copy of Blackjack Pro subject to all the rules and regulations of …

End User License Agreement

User: Publisher

The software license agreement that is distributed with every copy of your application will be stored in the License file associated with the TimeLOCKed application.   You may enter the text of the agreement in this screen, or simply import the ASCII file containing the text of the agreement by clicking on the **Import** button. A customer will not be able to purchase the TimeLOCKed application without accepting the terms of the End User License Agreement.

Input Field	Max Length	Description	Example
End user license agreement	N/A	Text of the end-user software license agreement	This copy of Blackjack Pro is subject to all the rules and …

Select Product Key File

User: Distributor

TimeLOCK may be used to integrate more than one product with TimeLOCK. In order to distinguish those products supported by the distributor, the Client Builder asks for the file that contains distributor and Product Identification information. The key and related information is stored in a file with a prd extension. This file is distributed by Preview Software. For more information on this file see the *Introduction* and *Getting Started* chapters. Once the file is loaded, the Distributor ID and Distributor SKU fields are filled in.

Input Fields	**Format**	**Description**	**Example**
File name	255 characters	Product file (*.prd) that contains identifying product information.	Blackjackpro.prd

Product Key Information

Product and distributor information will appear in this screen based on information in the Product Key (*.prd) file.

Transaction Method

User: Distributor

TimeLOCK allows customers to purchase TimeLOCKed applications by Internet and, in the case of Try Before You Buy, by phone, fax, or e-mail. You may enable your application to support one or more of these methods in the Client Interface. If you select:

§ **Internet transaction**, you will be asked for additional information for making purchases over the Internet in the next steps of the distributor section.

§ **Phone**, you will be asked for additional information in the next steps of the merchant section. The Phone option is only available when a product is packaged for Try Before You Buy distribution.

Clearinghouse Information

(Internet Transaction only)

User: Distributor

If the distributor selected Internet Transaction in the Transaction Method screen, the Clearinghouse Information screen will be displayed. The clearinghouse provides the information used in this screen. Preview Software creates a file (*.clr) that contains server information. If this field appears blank, then contact the distributor and request that they contact the clearinghouse for this information. If Internet Transactions are not supported at this time, then de-select the Internet option in the previous screen.

To add server information select**, Add**. You will be asked to enter the name of the file containing server information for the clearinghouse. This file should have a clr extension.

Input Field	**Max Length**	**Description**	**Example**
Clearinghouse Key File	255 characters	File containing information on clearinghouse server.	InterTrans.clr

Merchant Information

User: Merchant

Information on the merchant will be used in the Client Interface to purchase the product electronically.

§ Merchant name
§ Product SKU
§ Merchant account number: used to identify the merchant to clearinghouse.
§ E-mail address for customer service
§ Phone number for customer service

Input Field	**Max Length**	**Description**	**Example**
Merchant Name	125 characters	Merchant name	GameWare.com
SKU Number	125 characters	Product SKU used by merchant	gw_dynbj30
Account Number	125 characters	Merchant account number	gw1234
E-mail Address	125 characters	E-mail address for customer service	support@gameware.com
Phone Number	125 characters	Phone number for customer service	1-800-GAMEWARE

Product Price Information

User: Merchant

Price information will appear in the Purchase Wizard of the Client Interface so that the customer will know the current software price.

§ Price
§ Expiration Date is the last day for which the price is valid.

Input Fields	**Format**	**Description**	**Example**
Price	xxx.yy	Current price of the application	79.95
Expiration Date	Xx/yy/zzzz	Last day on which merchant's price is valid.	12-31-1997

Phone Transaction Information

(Phone Transaction only)

User: Merchant

Might be nice to clarify that this screen is for phone purchases only.

If a user chooses to purchase your TimeLOCKed application by phone, the Purchase Wizard of the Client Interface will display three separate messages. Each message may be a simple instruction or just general information. We recommend that these messages give the user a good idea of how to purchase your product.   Note that TimeLOCK generates default messages based on information entered in previous screens.

Input Fields	**Max Length**	**Example**
Instruction 1	150 Characters	Please telephone 1-800-GAMEWARE or e-mail purchase@Gameware.com to purchase this product
Instruction 2	150 Characters	Please provide your registration number and have your VISA MC or AMEX credit card ready.
Instruction 3	150 Characters	A permanent unlocking code will be provided for you to enter. Once the code is entered, press OK.

Message to New User

User: Merchant

This screen asks you to enter the message that will be displayed to a customer the first time they run the application. You may want to provide an enthusiastic message welcoming the user and thanking them for trying your application. This information will appear the first time the application is run, and never again. A default message is generated based on information entered in previous screens.

Input Fields	**Max Length**	**Example**
New User Message	255 characters	Welcome to the 1997 edition of Blackjack Pro for Windows 95! You have been granted a free trial to use a fully-functional edition of Blackjack Pro for 30 days …

Message to Trial User

User: Merchant

This screen asks you to enter the message that will be displayed to a user each time a trial version is run, provided the user is still in the trial period. You may want to provide a standard marketing message welcoming the user back and reminding them that they can purchase the application at any time by following the instructions in the Purchase Wizard of the Client Interface. A default message is generated based on information entered in previous screens.

Input Fields	**Max Length**	**Example**
Trial User Message	255 characters	Thank you for continuing to use this trial version of Blackjack Pro for Windows 95. As you use this product, keep in mind that your trial will expire soon. To purchase, simply click the Purchase button…

Expired Trial Message

User: Merchant

This asks you to enter the message that will be presented each time the user attempts to run the application following the expiration of the trial period. Typically, one might want to provide a marketing message stating the trial period has expired and reminding the user that they can purchase the application at any time by following the instructions in the purchase dialog. This information will appear each time the application is run after the trial period has expired. A default message is generated based on information entered in previous screens.

Input Fields	Max Length	Example
Expired User Message	255 characters	This trial edition of Blackjack Pro, offered by GameWare.com, is no longer accessible. To purchase, simply click the purchase button below and follow the instructions…

Customize Bitmaps

User: Merchant

TimeLOCK allows you to customize the Client Interface by adding your own bitmaps to the Main Dialog and Purchase Wizard screens. You can use this feature to incorporate the product or company logo. Note that the recommended size of the bitmaps is 60w x 60h pixels for the Main Dialog and 128w x 325h for the Purchase Wizard.

If your bitmap is smaller than the recommended size, it will be centered in the bitmap area. If no bitmap is specified for the Main Dialog it will appear blank. If no bitmap is specified for the Purchase Wizard, a default bitmap will be used. You can use the **Show Me** button to see exactly how the interface will look.

Input Field	Max Length	Description	Example
Bitmap for Main Dialog	255 characters	File containing bitmap for Main Dialog	Bitmapfile.bmp
Bitmap for Purchase Wizard	255 characters	File containing bitmap for Purchase Wizard	Bitmapfile.bmp

Instant Integration

(Instant Integration only)

If you are using Instant Integration to TimeLOCK your application, you will now be asked for the location of the application's executable file (*.exe). A status bar will appear showing you the progress of the integration.

Note: The executable file will be overwritten with the new executable. A backup copy will be made for you with the .bak extension. For example, blackjackpro.bak will be equivalent to the original version of Blackjack Pro before TimeLOCKing.

Input Field	Max Length	Description	Example
Application executable	255 characters	Executable for the application to be TimeLOCKed	Appliation.exe

What To Do Next

This screen will display specific information on what has been created in the top window. The bottom window will tell you precisely what steps are required to complete the integration. Below is a description of what is displayed in this screen under certain scenarios.

1. Finish for Direct, Instant Integration only

If you are using Instant Integration as in the previous screen, the executable for the TimeLOCKed application and the associated License (*.lic) file will have now been created.
In this example, blackjackpro.exe and blackjackpro.lic were created and are ready to be used with the TimeLOCK DLL (tl30inj.dll) to create the installation files. The TimeLOCK portion of the application is complete.

2. Multi-Tier Distribution, Try Before You Buy only

The Client Builder has created the BOB file to be used to secure distribution of your application to distributors and merchants. The BOB file should be included with the License Information (*.lif) and application executable files when passing the TimeLOCKed software to channel partners. See the *Example Scenarios* section of the *Getting Started* chapter for more information.

3. Multi-Tier Finish or Purchase Only Distribution: Finish for Self Extracting File option

This screen will appear when you have completed all the steps involved in TimeLOCKing your application and when you have created a self-extracting executable using the Client Builder. You will be told the name of the file created for you and that you may distribute it to customers.

4. Programmed Integration Part 1

If you are using Programmed Integration, you will need to run the Client Builder two times. This screen appears at the end of the first pass through the Client Builder.

You are now finished using the Client Builder and are ready to begin modifying the source code to suit your distribution needs. When you are done, you will need to run the Client Builder a second time to create the application License (*.lic) and setup (*.exe) files.

5. Programmed Integration Part 2

The executable for the TimeLOCKed application, using Programmed Integration, and the associated License (*.lic) file have now been created. The top window displays specific information on the files that have been created for your application. The bottom window tells you what steps are required to prepare the files for distribution.

In this case, you must prepare installation files using the wrapped executable file, license (*.lic) file, and TimeLOCK runtime DLL for the API (tl30api.dll). The TimeLOCK portion of your integration is complete.

6. Other Finish for Multi-tier Distribution

In a multi-tier distribution environment, other screens may appear to the publisher, distributor, or merchant when completing the Client Builder. These screens typically provide a friendly message letting the user know that the Client Builder was completed successfully and that the License Information (*.lif) file has been modified. In some cases, the screen will let the user know what files must be passed to the appropriate channel partner to complete the TimeLOCK integration.

Installation Directory

(Multi-Tier or Direct, Purchase Only Distribution)

In a multi-tier distribution environment or when direct, Purchase Only distribution is used, the Client Builder will create a new self-extracting executable for you.   This executable is used in a

§ Multi-Tier Model: to secure the transfer or files between the publisher and channel partners in a multi-tier distribution model.

§ Direct, Purchase Only Model: to create the self-extracting executable that is given to customers to install the TimeLOCKed application.

In order to create this file, TimeLOCK must incorporate your current installation files.   For this reason you must provide the installation directory for your application.

Input Field	**Max Length**	**Description**	**Example**
Installation Directory	255 characters	Directory that contains the installation files for your application.	C:\blackjackpro

Installation File(s)

(Multi-Tier or Purchase Only Distribution only)

In order to create the self-extracting executable described in the previous screen you must specify the current setup executable for your application.   You can select the setup file from the list of files in your installation directory displayed in the bottom window.

Installation Directory for Creating BOB File

(Multi-Tier Distribution, Try Before You Buy only)

In a multi-tier, Try Before You Buy distribution environment, you need to ensure that the files are secure from unauthorized use as they are passed from publisher to distributor to merchant. To do this, TimeLOCK uses an encrypted file, the BOB file (*.bob) to preserve the integrity of the TimeLOCKed executable and associated files. For more information on the BOB file see the *Security of the Client Builder* section in this chapter or the *Introduction*.

In this screen you are asked to provide the installation directory for your application. The files in this directory will be used to create the BOB file.

Create BOB File

(Multi-Tier Distribution, Try Before You Buy only)

In a multi-tier, Try Before You Buy distribution environment, you will now be asked to name the BOB file discussed on the previous page.

Input Field	Max Length	Description	Example
BOB file	255 characters	Name of the BOB file to be created	blackjackpro.bob

Create Self-Extracting Setup Executable

(Multi-Tier Finish or Purchase Only Distribution)

In the last stage of Multi-Tier distribution, or in Purchase Only distribution, you will be asked to name the setup file for your TimeLOCKed application. This file is a self-extracting executable containing all your product files including the TimeLOCKED application's executable and License file. This file will be given to customers to install the TimeLOCKed software.

Input Field	Max Length	Description	Example
Self-Extracting Executable	255 characters	Name of the self-extracting executable file that will be used by customers to install the TimeLOCKed application.	setup.exe

Specify License Filename

(Purchase Only Distribution)

If you are distributing your application on a Purchase Only basis, you will be asked to name the License (*.lic) file for your application. The License file contains information on the state of the user. When the customer purchases the application, the license file will note the purchase and be saved to the *Previewsoft* directory.

TimeLOCK Keys

(Programmed Integration only)

If you are using Programmed Integration, you will be given information on the TimeLOCK keys to use when modifying your source code. The:

§ **TimeLOCK ID Key** will be used to verify the application.
§ **License Key** will be used to access the License file (*.lic) for the application.

Clicking **Reset** will generate a new key. For further details on the purpose and use of these functions, see the chapter on using the TimeLOCK API.

TimeLOCK Return Values

(Programmed Integration only)

When a wrapped application is run, TimeLOCK returns one of the randomly assigned values to the application based on the state of the user. These values are used to determine whether the application may be run or whether the customer must first purchase the software. You will need to make note of these return values so they can be integrated into your source code. Alternatively, you may use the code generated by the Client Builder. If you wish to select your own values for return codes, you may do so at this time. These values all must be integer values greater than zero.

Clicking **Reset** will generate a new set of return values. For further details on the purpose and use of these return values, see the function *getUserState* in the chapter on using the TimeLOCK API.

Input Fields	**Max Length**	**Description**	**Example**
New User Return Code NEW	Long Integer Value (1 to 65535)	Denotes NEW user	345
Trial User Return Code TRIAL	Long Integer Value (1 to 65535)	Denotes TRIAL user	65500
Purchased User Return Code PURCHASED	Long Integer Value (1 to 65535)	Denotes user who has PURCHASED the TimeLOCKed application	7000
Expired User Return Code EXPIRED	Long Integer Value (1 to 65535)	Denotes the trial period has EXPIRED	9

Create Source Code

(Programmed Integration only)

To generate the source code for the application:

§ Select the **Source Code Language** from the pull-down menu. You may select from C/C++, Visual Basic, or Delphi

§ Click on **Create 32-bit target** to begin generating the code. You will be asked where to save the code. It will be saved in a text (*.txt) file by default. . The Progress bar will be displayed as the code is being generated.

§ Click on **View Source** to use Notepad to view the source code created for your application.

Analyze Application and Create License File

(Programmed Integration only)

As mentioned in the previous page, once you have integrated the TimeLOCK source into your application source code you will need to run the Client Builder a second time. After logging in and supplying the application Privilege (*.prv), Product (*.prd), and License Information (*.lif) files, the **Question for Opening Existing File** screen will appear. If you select Next on this screen (optionally you may add or modify license information first), the screen below will be displayed. This screen uses information in the TimeLOCKed application's executable to create the License (*.lic) file.

You must provide the:

§ Location of the TimeLOCKed application executable

§ Name for the License file

Once this is done, you may click on **Next** and the License (*.lic) file will be created. You should now have all the files needed to distribute the application to customers as described in the next screen.

File Progress

This screen displays progress on creating or generating a file.   For example, it will show the creation of the TimeLOCKed executable.

About Button

The About button displays basic product and version information for any TimeLOCK component.

Change Password

This screen prompts you for the new password. Enter the password in both input boxes to confirm the new string. If the new and confirmation passwords do not match, you will be asked to re-enter the password.

Chapter 4. Reset Wizard

This chapter describes the Reset Wizard and how it used to test TimeLOCKed applications before release.   The following sections will explain how the Reset Wizard works and when to use it with your application:

What is the Reset Wizard

The Reset Wizard is used by a publisher to test Phone Unlocking for Instant and Programmed Integration. During testing, the state of a TimeLOCKed application may change a number of times. Typically, the trial information will change or the state of the copy may switch from “NEW” or “TRIAL” to “PURCHASED”. It is useful to be able to reset the software to “NEW” so that testing may be done continuously. The Reset Wizard resets the TimeLOCK environment by reversing changes made to the application. This is done by removing tags and markers and modifying the License (*.lic) file. The result is that the copy of the TimeLOCKed application reverts to a “New User” state. All trial and purchase information has been removed.

How to Use the Reset Wizard

You must have publisher level privileges to use the Reset Wizard. The Reset Wizard verifies your privileges by prompting you for your Privilege (*.prv) file. You are then prompted for the name of the License Information (*.lif) file and its associated password. This information is used to reset the License file (*.lic) and remove the hidden tags and markers. We will now take you through each screen in the Reset Wizard.

Starting the Reset Wizard

There are two ways to start the Reset Wizard:

§ Double-click on the file `ResetWizard.exe` in the TimeLOCK folder.
§ Choose **Run** from the **Start** menu and select `ResetWizard.exe` from the TimeLOCK folder.

This screen appears upon starting the Reset Wizard.

Login to the Reset Wizard

- You must supply the pathname for your Privilege file (*.prv) and its access Password.
- The **Privilege file** field will display the path for the last such file used.   You may also use the **Browse** box to select an alternate file.   Once you enter this information click on the **Login** button.

Note: You must have publisher level privileges in order to use the Reset Wizard.

License Information File

You must now supply the License Information (*.lif) file for the TimeLOCKed application you wish to test.   To do this:

- Enter the pathname for the LIF file in the **License Information File** field.   You may use the **Browse** box to locate the proper directory.
- In the **Password** field enter the Access Password for the LIF file followed by the **Return** key.

Application License File

The Reset Wizard now looks for the License file (*.lic) in the directory containing the License Information file (*.lif).   If it is not found the Application License File screen will appear.   It will prompt you for the name and location of the License file (*.lic).   You may use the Browse box to locate the file.

If the file has been deleted you must check the box labeled "Check here if the file has been deleted."   In this case, the Reset Wizard will remove all hidden tags and markers that may be left over on your machine. Deleting the file from disk is not enough to reset the environment.   You must run the Reset Wizard.

Reset Confirmation

Before TimeLOCK proceeds to reset the environment you are asked to confirm that you wish this to be done. The RESET screen will appear. Press:

§ **Yes**: to reset the environment
§ **No**: to close the Reset Wizard without making any changes

Information Screen

Once TimeLOCK has finished resetting the environment, a message will appear letting you know the reset was successful.   This messsage appears in the Information screen.

Chapter 5. Troubleshooting

In this chapter we will discuss some of the steps that can help you isolate and solve a problem before you call technical support. The following sections will be covered:

Check the TimeLOCK Product Documentation

This is one of the most productive ways to find answers to questions. You can consult several types of documentation:

§ **On-line Help**: On-line Help includes procedural and reference information on common features, functions, and error messages. You can access Help through the Help menu.

§ **Read Me file**: This file contains late-breaking information about configuration issues, new features, and known bugs. You can open README.WRI in an editor or in Notepad.

§ **Samples**: The TimeLOCK Developer version includes sample programs that illustrate TimeLOCK programming tasks.

§ **TimeLOCK's Web site**: TimeLOCK's own developer Web site is dedicated to bringing up-to-date development and product information on TimeLOCK and derivative products to new and current TimeLOCK customers.   Look for the link to the Developers' FAQs.

Reproduce the Problem

Reproducing the problem is the first step in solving it. If you can reproduce the problem, it is easier to start finding solutions. The following questions may give you more insight on the problem:

§ **Does the problem occur with just this one program?** You may want to try one of the samples to see if you can reproduce the problem with it. If you cannot reproduce the problem with other programs, think about what is different about the program.

§ **Does the problem occur on just your machine?** If so, the problem may be related to your system configuration. Try using an entirely different product ID and product name or modify your system configuration to see if the problem still occurs. It is a good idea to try to make your machine as much like the average machine as possible.

§ **What version of TimeLOCK are you using?** Knowing the version of the Client Builder, Key Module, Reset Wizard and the TimeLOCK Runtime DLL makes it easier to reproduce (or avoid) the problem in the future.

§ **Under what circumstances does the problem occur?** Does the amount of available memory affect the problem? How about other programs that are running in the system? Which component seems to be causing the problem?

Isolate the Problem

After determining what circumstances cause the problem, you may be able to isolate it. Once a problem is isolated, it is much easier and quicker to fix or work around it.

If you are using the TimeLOCK API, try isolating the section of code or component that is causing the problem. You can use the information about what conditions it reproduces to help isolate the component.

Chapter 7.   The Key Module

In this chapter we will describe the TimeLOCK Key Module and how it is used to generate Phone Unlocking Keys for TimeLOCKed applications.   This information will be presented in two sections:

What is the Key Module?

The Key Module is used to generate Phone Unlocking Keys for TimeLOCKed applications.   The Phone Unlocking Key is a 16-digit number that is used to change the status of a TimeLOCKed application to either lengthen a trial period or record a purchase. Typically, a distributor uses a clearinghouse to process these orders.   In other cases, the publisher or distributor may use its own customer support staff to handle phone, fax, or e-mail requests.   In this chapter, we will refer to the organization that processes requests for Phone Unlocking Keys as the clearinghouse.

Note: Phone Unlocking Keys are 16-digit numbers generated using a security scheme developed by Preview Software.   They are not as secure as Electronic Unlocking Keys that are generated using an RSA scheme. Electronic Keys are handled by a separate software package called the TimeLOCK Transaction Server.   Please contact Preview Software for information on this product.   For more information on security mechanisms available in TimeLOCK, see the *Introduction*.

When running the Purchase Wizard of a TimeLOCKed application, the customer may have the option to purchase the product by telephone.   In this case, the Phone Purchase screen of the Purchase Wizard (see the chapter*, Client Interface*) will prompt him for an Unlock Key.   At this point the customer contacts the clearinghouse, or appropriate organization, noting the name and registration number given in the Wizard.   The clearinghouse then runs the Key Module to generate the Phone Unlock Key.

How to Use the Key Module

The publisher may need to help the clearinghouse, or other organization, set up the Key Module to support the publisher's TimeLOCKed applications. We will describe how the Key Module works in the following sections:

Starting the Key Module
Key Module Main Dialog

Note: You must have publisher or distributor level privileges in order to use the Key Module.

Starting the Key Module

There are two ways to start the Key Module:

§ Double-click on the file `KeyModule.exe` in the TimeLOCK folder.

§ Choose **Run** from the **Start** menu and select `KeyModule.exe` from the TimeLOCK folder.

The Login screen appears. You must supply the pathname for your Privilege (*.prv) file and its access password. The **Privilege file** field will display the path for the last such file used. You may also use the **Browse** box to select an alternate file.

Key Module Main Dialog

Key Module Main Dialog

Once you have logged in, the Key Module Main Dialog will appear.

The Key Module scans the current working directory for all Product Key (*.prd) and associated Key (*.key) files and uses information in those files to fill in the **Product** and **Distributor ID** fields. The Product Key files are generated by Preview Software.   You may use the **Change Directory** button to change the working directory to that which contains the Product Key files for your applications.

Use the pull-down menu in the Product field to select the TimeLOCKed application of interest.   There is one Product Key file for each TimeLOCKed product.   This file contains the Distributor SKU used to identify the Product in the Key Module **Product** field and the distributor in the **Distributor ID** field.

Unlocking Products by Phone
Restoring Trials

Unlocking Products by Phone

When a customer requests a Phone Unlock Key for a specific application, they must supply the product name and Registration Number as they appear in the Phone Purchase screen of the Purchase Wizard. The Registration Number is a 16-digit number generated by the TimeLOCK Purchase Wizard. Once this information is supplied, the Key Module user performs the following actions to generate the Unlock Key:

§ Select the product name in the **Product** field using the pull-down menu. The **Distributor ID** will be filled in automatically based on the contents of the Product Key (*.prd) file.

§ Enter the customer registration number in the **Registration #** field.

§ Click on **Unlock Code** to generate the Unlock Code specific to the registration number and application title. The **Unlock Code** field will be filled in.

§ Give the Unlock Code to the customer. The customer must enter the Unlock Code in the Purchase Wizard to convert the TimeLOCKed application into a fully-functional unlocked version of the product.

Note: The customer should create a back-up copy of his License (*.lic) file before entering the Unlock Code.

Restoring Trials

When a customer wants to restore a trial period (e.g., if they want just a few more days to review a TimeLOCKed application, or if, for some reason, the TimeLOCKed application has been corrupted), they must supply the product name and Registration Number as they appear in the Client Interface. The Registration Number is a 16-digit number generated by the TimeLOCK Purchase Wizard. Once this information is supplied, the Key Module user performs the following actions to generate the Restore Key:

§ Select the product name in the **Product** field using the pull-down menu. The **Distributor ID** will be filled in automatically based on the contents of the Product Key (*.prd) file.

§ Enter the customer registration number in the **Registration #** field.

§ Click on **Restore Code** to generate the restore code specific to the registration number and application title. The **Restore Code** field will be filled in.

§ Give the Restore Code to the customer. The customer then enters the Restore Code in the Client Interface to convert the TimeLOCKed application into a new trial version of the product.

Note: The customer should create a back-up copy of his License (*.lic) file before entering the Restoration Code.

About the Key Module

This screen gives current information on TimeLOCK’s Key Module facility.

Chapter 6.   Client Interface

The Client Interface, also known as the End-User Interface, is the set of screens and dialogs that appear to a customer when using a TimeLOCKed application.   In this chapter we will describe how the Client Interface is created from information entered in the Client Builder.   We will also walk you through all screens that may be generated by TimeLOCK.   This is done in two sections:

What is the Client Interface

The Client Interface is the set of screens and dialogs that appear to an end-user when running a TimeLOCKed application that has not been purchased.   Once the application has been purchased, the TimeLOCK screens are no longer necessary and the publisher's application interface is all that the user will see.

The information that appears in the Client Interface is based on data provided by the publisher, distributor, and merchant in the Client Builder.   E.g., User messages, bitmaps, product name, version, and price all appear in screens in the interface.   The screens that are used to purchase the application are also determined by the purchase transactions selected by the distributor.

Instant Integration will use the TimeLOCK Client Interface by default. With Programmed Integration, a developer must use the TimeLOCK API function *ShowMainDialog* to use this interface.   See the chapter *Using the TimeLOCK API* for details on how to customize the Client Interface when using Programmed Integration.

How to Use the Client Interface

The Client Interface consists of a set of dialogs and wizards that typically display trial information and give the user the option to purchase the application. If the user elects to purchase the application, the Purchase Wizard will appear. The user either selects options or enters user information to complete each screen of the Purchase Wizard.

Some of the buttons common to several screens are:

§ **Back**: You may return to the previous screen by selecting the **Back** button. This is useful if you need to view or modify information that has been entered in previous screens.
§ **Next**: You may proceed to the next screen, once the current screen is complete, by selecting the **Next** screen.
§ **Cancel**: You may exit the Client Interface without saving any changes by selecting the **Cancel** button.

Main Dialog
Trial User License Agreement
Purchase Wizard

Main Dialog

The Main Dialog contains information on the product and associated trial (if applicable). The message displayed will vary depending on the state of the user. The text of the message is specified in the merchant section of the Client Builder.

§ **New User**: The New User message appears as it was entered in the Client Builder.
§ **Trial User**: The Trial User message appears as it was entered in the Client Builder.
§ **Expired Trial**: If this is a trial copy and the trial has expired, the user will get a message giving them an option to purchase. The user will not be able to run the application without an unlocking key. The unlocking key will be generated upon purchase of the product.

Note: If a bitmap was specified in the merchant section of the Client Builder, it will appear in the upper left corner of the Main Dialog. If no bitmap was specified, the entire box will have a gray background.

If the user selects:

§ **Buy Now**: the Purchase Wizard begins.
§ **Try First**: the application starts up. A trial license agreement may appear under conditions described in the next section. This only applies to applications that have been TimeLOCKed for Try before you buy distribution.
§ **Cancel**: the Main Dialog closes, the application does not start, and trial information is not changed.

Trial User License Agreement

If a Trial User License Agreement was defined in the Client Builder and the user has never accepted the Trial User License Agreement, then the User License Agreement screen will appear. Once the user selects:

§ **I Accept**: the License Agreement will not appear again during the trial period.
§ **I Don't Accept**: the user will not be able to run the trial application.

Purchase Wizard

The Purchase Wizard gives the user the friendly greeting that was entered in the merchant section of the Client Builder along with the following product information:

§ Product Name
§ Product Version
§ Purchase Price

The user may purchase the product by selecting **Next**.

Purchase Method
Customer Information for Internet Transactions
Payment Information for Internet Transactions
End-User License Agreement
Order Confirmation for Internet Transactions
Internet Status
Phone Purchase
Thank You

Purchase Method

This screen prompts the user to choose how they would like to purchase the software. The options displayed depend on what was chosen by the distributor in the Transaction Method screen of the Client Builder.

§ **Internet**: to purchase over the Internet
§ **Phone**: to purchase by telephone, fax, or e-mail

Customer Information for Internet Transactions

If a user selects **Internet Transaction** for his purchase method, the next display asks the user for information about who is purchasing the product. This information will be sent to the clearinghouse. The clearinghouse will forward the information to the publisher, merchant, or distributor as appropriate for the chosen distribution model.

Of course this information is useful for sending product material to the customer. But it also is a valuable tool in preventing fraud. The user will not be able to continue without supplying certain information as shown in the following table.

Input Field	**Description**	**Example**
First name	First name and middle initial	Jesse C.
Last name	Last name	Josephson
Company (optional)	Company name	Ruby Enterprises
Address (line 1)	First line of mailing address	1 Main Street
Address (line 2) (optional)	Second line of mailing address	Suite 1000
City	City for mailing address	Las Vegas
State	State for mailing address	NV
Zip	Postal code for mailing address	91111
Country	Country for mailing address (May use pull-down menu)	United States of America
E-mail (optional)	E-mail address	Jcj@ruby.com
Phone Number (optional)	Your telephone number	1-800-RUBYENT

Payment Information for Internet Transactions

The user is now prompted for particular payment information. This information will be sent through a secure communications channel. The RSA icon that appears in the lower part of the screen denotes that the information being entered will be sent using the RSA encryption standard.

Input Field	Description	Example
Credit Card Number	Credit card account number. TimeLOCK will apply a LUHN check to the number to verify that it has the correct format.	4111111111111111
Credit Card Type	Select from the list of Credit Card Types	Visa
Card Holder Name	Account name exactly as it appears on the card	Jesse C. Josephson
Card Expiration Date (Month/Year)	Use the pull-down menus to select the month and year	12 2000

End-User License Agreement

In order for a purchase to be completed, the user must accept the terms of the product license agreement. The end-user license agreement entered in the Client Builder is presented to the user in this screen. Once the user selects:

§ **I Accept**: the purchase transaction will proceed

§ **I Don't Accept**: the purchase transaction is cancelled

Order Confirmation for Internet Transactions

If the user is using the Internet to make a purchase, this screen displays a summary of the information entered by the customer. The customer can do one of three things:

- Modify the Information by selecting **Back** and re-entering invalid information
- Make the Purchase by selecting **Next**
- Cancel the order and quit the Purchase Wizard by selecting **Cancel**

Internet Status

A screen with information on the status of the order will be displayed once the order has been submitted. At this point, the clearinghouse is processing the order. Any information on the success or failure of the order will be sent to the customer. If the order is completed successfully, the clearinghouse will update the customer's License (*.lic) file to modify his status from trial to purchase.

Phone Purchase

When a customer selects **Phone** as their purchase method, they get a screen that tells them how to contact the merchant to purchase the software by phone, fax, or e-mail.   When the purchase has been completed successfully, the customer will be given a Phone Unlocking Key, a 16-digit number.   They must enter the key in the **Unlock Code** field to “unlock” the TimeLOCKed application.

Thank You

Once a purchase has been completed successfully, a message appears letting the customer know that the application has been unlocked. The user can now select **Done** and start using the software. Note that the Thank You screen appears no matter which purchase method was used.

Chapter 8.   Using the TimeLOCK API

(Programmed Integration only)

In this chapter, we will describe how to use the TimeLOCK API to customize the Client Interface using Programmed Integration with your application source code.   There are five major components to Programmed Integration that will be discussed in the next sections:

This chapter is designed to familiarize you with the methodology for integrating the TimeLOCK API set into your application. With effective integration, you can be assured of accurate trial information and well-managed security mechanisms.

The TimeLOCK Application Programming Interface   (API) simplifies the development of wrapped applications by hiding the complexities and tedium of low-level system identification, security and time-maintenance programming.   TimeLOCK accomplishes this by establishing a unique trial envelope around the application that maintains information specific to the application and its current user.

Connections to this mechanism and functionality may be accomplished in a number of ways, but always must be preceded by a Verification/Authentication call to the *verifytimeLOCK32* function which provides coincident handshaking between TimeLOCKed application and the TimeLOCK runtime DLL.

Examples are given for C, C++, and VisualBasic

Using The TimeLOCK API

There are several features of the TimeLOCK API you should take note of before beginning to modify your source code:

Static Libraries

You can link in the TimeLOCK runtime libraries statically or dynamically. The advantage to the static approach is that if a user wants to tamper with the application they must physically edit the source code. In contrast, dynamic libraries may be replaced completely.

Client Builder

Before and after coding, you must run the Client Builder. The Client Builder is used to generate the unique hash code that protects your executable from tampering.   Please see the *Getting Started* and *Client Builder* chapters for details on how this is done.

Before You Begin

One of the most important things to remember is that TimeLOCK does not take control over the TimeLOCKed application. It merely maintains usage and system-level information to accurately monitor the use of the application. It is the TimeLOCKed application that will provide or inhibit functionality based upon the information returned from the TimeLOCK API.

There are only three tasks you must complete to maintain the TimeLOCKed environment:

§ Get the state of the user
§ Open the environment
§ Close the environment

These tasks may be accomplished by integrating a few function calls (usually upon program startup) as described in the following sections. Regardless of the nature of the application being wrapped, TimeLOCK is used in a single, consistent manner. This allows developers to apply the same programming techniques to a broad range of Windows applications.

Authentication

Before you gain access to the information in the LicenseFile (*.LIC), you must verify or *authenticate* your application by calling the *verifyTimeLock32* function and providing the license file name, your password, and an empty buffer as parameters. If the password is found to be authentic to the associated TimeLOCKed application, TimeLOCK will fill in the buffer parameter with a 24 digit string that was provided in the Client Builder step called TimeLOCK Key.   Your application uses this code to verify and authenticate the TimeLOCK Runtime DLL.

Example (32-bit Visual Basic Application):
Example (32-bit C application)
Example (32-bit MFC C++ application)

Example (32-bit Visual Basic Application)

```
Global Module
' TSF File Read Errors
Global Const TLOCK_ERROR_WRONGVERSION = -5
Global Const TLOCK_ERROR_NOACCESSYET   = -4
Global Const TLOCK_ERROR_FILEREAD        = -3
Global Const TLOCK_ERROR_BADPASSWORD   = -2
Global Const TLOCK_ERROR_FILENOTFOUND = -1
Global Const TLOCK_ERROR_SUCCESS          = 1
'assuming the tl30api dll is in the windows or local directory...

Declare Function verifytimeLOCK32 Lib _
"tl30api.dll" _
(ByVal lpszFileName As String, ByVal strPassword As String, _
ByVal strResult As String) As Long

Declare Function getUserState Lib _
"tl30api.dll" _
(ByVal strPassword As String) As Long

Declare Function getUserName32 Lib _
"tl30api.dll" _
(ByVal strResult As String) As Integer

Declare Function getRegNum32 Lib _
"tl30api.dll" _
(ByVal strResult As String) As Integer

Declare Function getUserNumExecutions Lib _
"tl30api.dll" _
(ByVal strPassword As String) As Integer

Declare Function showMainDialog Lib _
"tl30api.dll" _
(ByVal strPassword As String) As Long

Declare Function trialEnvironmentOpen Lib _
"tl30api.dll" _
() As Integer

Declare Function trialEnvironmentClose Lib _
"tl30api.dll" _
() As Integer

End Global Module
```

```
Private Sub Form_Load()
   Dim x As Long
   Dim userType As Long
   Dim buffer As String * 24
   Dim buff As String
   Dim szFileName As String
   Dim szPassword As String
   szFileName = "wizfile.lic"
   szPassword = "123456"
   'From the Client Builder - TimeLOCK ID Key is
   ‘    025709477523044373699139
   'New User return code                 = 75489
   'Trial User return code              = 67657
   'Purchased User return code       = 107685
   'Expired User return code          = 63843

   x = verifytimeLOCK32(szFileName, szPassword, buffer)
   Select Case x
      Case TLOCK_ERROR_FILEREAD
         MsgBox "TLOCK_ERROR_FILEREAD"
      Case TLOCK_ERROR_BADPASSWORD
         MsgBox "TLOCK_ERROR_BADPASSWORD"
      Case TLOCK_ERROR_FILENOTFOUND
         MsgBox "TLOCK_ERROR_FILENOTFOUND"
      Case TLOCK_ERROR_SUCCESS
         MsgBox "TLOCK_ERROR_SUCCESS"
         buff = Left(buffer, 25)
         ' perform string comparison to ensure you have the
         ' correct string returned from timeLOCK.. (Strcmp
         ' returns 0 if both strings are equal...
         If StrComp(buff, "025709477523044373699139", 1) = 0 _
         Then
            userType = showMainDialog(szPassword)
            Select Case userType
               Case TLOCK_ERROR_WRONGVERSION:
                  MsgBox "TLOCK_ERROR_WRONGVERSION"
               Case TLOCK_ERROR_NOACCESSYET:
                  MsgBox "TLOCK_ERROR_NOACCESSYET"
               Case TLOCK_ERROR_BADPASSWORD:
                  MsgBox "TLOCK_ERROR_BADPASSWORD"
```

```
          Case userType > 0
            ' value must have been greater than 0, so check
            ' it against your TrialWizard assigned long
            ' integer return values for user type
            If userType = 75489 Then
              ' new user
              trialEnvironmentOpen '
              MsgBox "new user"
            ElseIf userType = 67657 Then
              ' trial user
              trialEnvironmentOpen
              MsgBox "trial user"
            ElseIf userType = 107685 Then
              ' Purchased user
              MsgBox "Purchased user"
            ElseIf userType = 63843 Then
              ' expired   user
              ' shut the app down or limit functionality
              ' or whatever you want to do when a user
              ' has fully utilized the trial period
              MsgBox "expired   user (shutting down)"
              End:
              ' in this case we return 0 from the Winmain
              ' function before we enter the message loop
              ' or show the main window...
            End If
        End Select
      Else
        MsgBox "TimeLOCK returned an unequal string:" & _
              buff
        End:
      End If
    Case Else
      MsgBox "(shutting down)", "Unknown Error", _
             MB_TASKMODAL
      End:
  End Select
End Sub

Private Sub Form_Unload(Cancel As Integer)
  Dim userType As Long
  Dim szPassword As String
  szPassword = "123456"
  UserType = getUserState(szPassword)
  If userstate = 75489 or userstate = 67657 Then
    trialEnvironmentClose
End Sub
```

Example (32-bit C application)

```
int x;
char buffer[] = "0000000000000000000000000";
switch(verifytimeLOCK32((LPCTSTR)m_szTDKFileName,
       (LPCTSTR)m_szPassword, (LPTSTR) buffer))
{
  case TLOCK_ERROR_FILEREAD:
    strcpy(buffer,"TLOCK_ERROR_FILEREAD");
    break;
  case TLOCK_ERROR_BADPASSWORD:
    strcpy(buffer,"TLOCK_ERROR_BADPASSWORD");
    break;
  case TLOCK_ERROR_FILENOTFOUND:
    strcpy(buffer,"TLOCK_ERROR_FILENOTFOUND");
    break;
  case TLOCK_ERROR_SUCCESS:
    //compare with TrialWizard assigned TimeLOCK ID Key
    if (!strcmp(buffer, "123456781234567812345678")
    {
      // RETURN CODE MATCHES THE PRE_ASSIGNED ID!
      // now check the state of the user (i.e. NEW
      // TRIAL EXPIRED PURCHASED)
    }
    else
    {
      // THE RETURN CODE DOES NOT MATCH THE
      PRE_ASSIGNED ID!
      // should probably shut down - no validation
    }
  default:
    //do not proceed
    break;
}
```

Example (32-bit MFC C++ application)

```
#include "stdafx.h"
#include "mfctrial.h"

#include "MainFrm.h"
#include "mfctrialDoc.h"
#include "mfctrialView.h"
#include "tl30api.h"


/////////////////////////////////////////////////////////////////////////////
// The one and only CMfctrialApp object

CMfctrialApp theApp;

/////////////////////////////////////////////////////////////////////////////
// CMfctrialApp initialization

BOOL CMfctrialApp::InitInstance()
{
  // Standard initialization
  long x;
  long userType;
  char buffer[] = "000000000000000000000000000";
  char szFileName[] = "wizfile.lic";
  char szPassword[] = "123456";

  // From the Client Builder - TimeLOCK ID Key is
  //    025709477523044373699139
  // New User return code          = 75489
  // Trial User return code        = 67657
  // Purchased User return code    = 107685
  // Expired User return code      = 63843

  x = verifytimeLOCK32((LPCTSTR)szFileName,
                       (LPCTSTR)szPassword,
                       (LPTSTR) buffer);
  switch (x)
  {
    case TLOCK_ERROR_WRONGVERSION:
      MessageBox (NULL, "TLOCK_ERROR_WRONGVERSION",
                  "verifytimeLOCK32",MB_TASKMODAL);
      return FALSE;
    case TLOCK_ERROR_NOACCESSYET:
      MessageBox (NULL, "TLOCK_ERROR_NOACCESSYET",
                  "verifytimeLOCK32",MB_TASKMODAL);
      return FALSE;
    case TLOCK_ERROR_FILEREAD:
      MessageBox (NULL,"TLOCK_ERROR_FILEREAD",
             "verifytimeLOCK32",MB_TASKMODAL);
      return FALSE;
    case TLOCK_ERROR_BADPASSWORD:
      MessageBox (NULL, "TLOCK_ERROR_BADPASSWORD",
                  "verifytimeLOCK32",MB_TASKMODAL);
      return FALSE;
    case TLOCK_ERROR_FILENOTFOUND:
      MessageBox (NULL, "TLOCK_ERROR_FILENOTFOUND",
                  "verifytimeLOCK32",MB_TASKMODAL);
      return FALSE;
    case TLOCK_ERROR_SUCCESS:
    {
      // Compare with Client Builder assigned
      // TimeLOCK ID Key
      if (!strcmp(buffer, "025709477523044373699139" ))
      {
        // THE RETURN CODE MATCHES THE PRE_ASSIGNED ID!
        // Your conditional execution is determined here
        // based uponthe various values only your
        // application knows.   For instance, an IF THEN
        // scenario or continued SWITCH scenario will work
        // fine - but you do need to know the values that
        // were provided by you in the Client Builder.

        userType = showMainDialog((LPCTSTR)szPassword);
        switch (userType)
        {
          case TLOCK_ERROR_WRONGVERSION:
             MessageBox (NULL,"TLOCK_ERROR_WRONGVERSION",
                           "showMainDialog",MB_TASKMODAL);
             return FALSE;
          case TLOCK_ERROR_NOACCESSYET:
             MessageBox (NULL, "TLOCK_ERROR_NOACCESSYET",
                           "showMainDialog",MB_TASKMODAL);
             return FALSE;
          case TLOCK_ERROR_BADPASSWORD:
             MessageBox (NULL, "TLOCK_ERROR_BADPASSWORD",
                           "showMainDialog",MB_TASKMODAL);
             return FALSE;
          default:
          {
             // value must be > 0, so check it against your
             // Client Builder assigned return values for
             // user type
```

```
            if (userType == 75489)
            {
               // new user
               trialEnvironmentOpen();
            }
            if (userType == 67657)
            {
               // trial user
               trialEnvironmentOpen();
            }
            if (userType == 107685)
            {
               // Purchased user
            }
            if (userType == 63843)
            {
               // expired   user
               // shut the app down or limit functionality
               // or whatever you want to do when a user
               // has fully utilized the trial period
               MessageBox (NULL,
                           "expired   user (shutting down)",
                           "showMainDialog",MB_TASKMODAL);
               return FALSE;
               // in this case return 0 from Winmain
               // before we enter the message loop or
               // show the main window...
            }
            break;
         }   // end default case of switch (usertype)
      }   // end switch userType
   }   //   end if strcmp
   else
   {
      // the returned 24 character timeLOCKID DID NOT
      // match the string you assigned in the Client
      // Builder you should probably shut down...
      MessageBox (NULL,
         "the returned timeLOCKID DID NOT match",
         "verifytimeLOCK32", MB_TASKMODAL);
      return FALSE;
      // in this case we return 0 from the Winmain
      // function before we enter the message loop or
      // show the main window...
   }
   break;
}   //   end case TLOCK_ERROR_SUCCESS
default:
     // verifytimeLOCK32 returned a value other than those
   // predefined.   You should probably shut down...
   MessageBox (NULL, "Shutting Down",
                     "verifytimeLOCK32",MB_TASKMODAL);
   return FALSE;
   // in this case we return 0 from the Winmain function
   // before we enter the message loop or show the main
   // window...
   } // end switch (x)
   // If you are not using these features and wish to reduce
   // the size of your final executable, you should remove
   // from the following the specific initialization
   // routines you do not need.

#ifdef _AFXDLL
   Enable3dControls();   // Call this when using MFC in a
                              // shared DLL
#else
   Enable3dControlsStatic();   // Call this when linking to
                                     // MFC statically
#endif

   LoadStdProfileSettings();   // Load standard INI file
                                     // options (including MRU)

   // Register the application's document templates.
   // Document templates serve as the connection between
   // documents, frame windows and views.

   CSingleDocTemplate* pDocTemplate;
   pDocTemplate = new CSingleDocTemplate(
         IDR_MAINFRAME,
         RUNTIME_CLASS(CMfctrialDoc),
         RUNTIME_CLASS(CMainFrame),     // main SDI frame window
         RUNTIME_CLASS(CMfctrialView));
   AddDocTemplate(pDocTemplate);

   // Parse command line for standard shell commands, DDE,
   // file open
   CCommandLineInfo cmdInfo;
   ParseCommandLine(cmdInfo);

   // Dispatch commands specified on the command line
   if (!ProcessShellCommand(cmdInfo))
     return FALSE;

   return TRUE;
}
```

Getting User Information

Once the authentication is successfully completed, the next step for the trial application is to determine the *State* of the user. You need to identify the user as being in one of the following four states:

§ **NEW**: first time user
§ **TRIAL**: continued trial use
§ **PURCHASED**: purchased and registered the application by unlocking
§ **EXPIRED**: trial period fully used with zero time or sessions remaining

If the user is determined to be `NEW`, TimeLOCK will automatically generate a statistically unique 16-digit numeric string that will be assigned to this user. This is the Registration Number.

Determining the state of the end user can be accomplished in one of two ways. If the display of the TimeLOCK Main Dialog is desired, the function *showMainDialog* must be used. This function will not only return accurate user identification information, but will also automatically display the Main Dialog interface. If, however, you wish to simply determine the state of the user and *not* display the Main Dialog interface, call the *getUserState* function.

The *getUserState* function is useful for trial implementations where you do not want to display the TimeLOCK interface. Sub-system processes, operating system services or expiring beta releases are all good examples for this transparent. It is important to remember that a user cannot perform the steps necessary to purchase your application if they cannot access the TimeLOCK Purchase Dialog (shown when the user clicks the Purchase button in the TimeLOCK Main Dialog). One alternative would be to implement the *getUserState* function until the return value indicates `EXPIRED` and, at that point, call *showMainDialog*.

Both the *showMainDialog* and *getUserState* functions will return either a negative value representing one of the error return codes or one of the values that were assigned by you in the Client Builder step called Trial Return Codes.

Once the trial application has received the state of the user, application functionality is then either provided or inhibited based on this information.

Example (32-bit C application):

Example (32-bit C application):

```
long userType;
char buffer[] = "000000000000000000000000";
switch (verifytimeLOCK32((LPCTSTR)m_szTDKFileName,
                         (LPCTSTR)m_szPassword, (LPTSTR)
                         buffer)
{
  case TLOCK_ERROR_FILEREAD:
    strcpy(buffer,"TLOCK_ERROR_FILEREAD");
    break;
  case TLOCK_ERROR_BADPASSWORD:
    strcpy(buffer,"TLOCK_ERROR_BADPASSWORD");
    break;
  case TLOCK_ERROR_FILENOTFOUND:
    strcpy(buffer,"TLOCK_ERROR_FILENOTFOUND");
    break;
  case TLOCK_ERROR_SUCCESS:
    //compare with Client Builder assigned TimeLOCK ID Key
    if (!strcmp(buffer, "123456781234567812345678"))
    {
      // THE RETURN CODE MATCHES THE PRE_ASSIGNED ID!
      // your conditional execution is determined here
      // based upon the various values only your
      // application knows, for instance, an IF THEN
      // scenario or continued SWITCH scenario will work
      // fine - but you do need to know the values that
      // were provided by you in the Client Builder.
      userType = showMainDialog((LPCTSTR)m_szPassword);
      switch (userType)
      {
        case TLOCK_ERROR_WRONGVERSION:
          break;
        case TLOCK_ERROR_NOACCESSYET:
          break;
        case TLOCK_ERROR_BADPASSWORD:
          break;
           default:
          // value must have been greater than 0, so check
          // it against your Client Builder assigned long
          // integer return values for user type
```

```
                if (userType = 12345)
                {
                   // new user
                   trialEnvironmentOpen();
                }
                if (userType = 23222)
                {
                   // trial user
                   trialEnvironmentOpen();
                }
                if (userType = 4566)
                {
                   // expired   user
                   // shut the app down or limit functionality
                   // or whatever you want to do when a user
                   // has fully utilized the trial period…
                }
                if (userType = 222)
                {
                   // Purchase user
                }
                break;   // from default case
          }   // end switch(userType)
       }   // end if strcmp()
   break;   //   end case TLOCK_ERROR_SUCCESS
default:
   //do not proceed
    break;
}
```

Opening the Trial Environment

Once the authentication and the subsequent determination of the state of the user is complete, the trial application should then make a call to the *trialEnvironmentOpen* function *only* if the user state is either Trial User or New User.   If the user state is anything else, there is no need to start up system timers and increment usage counters, which are the functions performed by *trialEnvironmentOpen*.

Example (32-bit C application):

Example (32-bit C application):

```
if (trialEnvironmentOpen())
        ;//("Successfully updated environment parameters.");
    else
        ;//("Unsuccessful in opening trial environment.");
```

Closing the Trial Environment

Whenever the trial application is exited, the application must close the trial environment to record information about usage during this trial session. This is done quite simply with a call to *trialEnvironmentClose*.

This call also resets the an application-specific global ‘access’ setting to false, so that all subsequent calls to the TimeLOCK API will result in a `TLOCK_ERROR_NOACCESSYET` or `TLOCK_ERROR_BADPASSWORD` return value until the trial application re-authenticates with the verification functions described in the Authentication section above.

Example (32-bit C application):

Example (32-bit C application):

```
if (trialEnvironmentClose())
        ;//("Successfully closed trial environment.");
    else
        ;//("Unsuccessful in closing trial environment.");
```

Customizing the Client Interface

The *showMainDialog* function is used to display the first dialog to the customer.   This function is typically used to display product and trial information along with an option to purchase. The merchant section of the Client Builder is used to specify the message presented to the user and the bitmap displayed.   If no message or bitmap is specified in the Client Builder, the screen will appear with a plain gray background.

The *showMainDialog* function determines the state of the TimeLOCKed application and displays the appropriate message. E.g., The message displayed will be equal to the New, Trial, or Expired User Messages, depending on whether the state of the user is NEW, TRIAL, or EXPIRED.   If the state of the user is PURCHASED, this screen will not appear at all.

Chapter 9. Function Reference

This chapter gives detailed instructions for using each of the functions in the TimeLOCK API. The available functions, in alphabetical order, are as follows:

getRegNum32

BOOL getRegNum32(LPTSTR strResult)

Parameter	**Description**
strResult	Address of buffer for user registration number

Comments

The *getRegNum32* function is called by the TimeLOCKed application to retrieve the user's registration number. This function will either return TRUE or FALSE, depending on the outcome of the *verifyTimeLOCK32* function. This authentication step is necessary to gain access to the information contained within the License Information (*.lif) file and the trial environment itself.

The size, in characters, of the buffer specified by the *strResult* parameter should be set to at least 17 to allow sufficient room in the buffer for the 16-character registration number.

Returns

TRUE on success
FALSE on failure

getUserName32

BOOL getUserName32(LPTSTR strResult)

Parameter	**Description**
strResult	Address of buffer for user name.

Comments

The *getUserName32* function is called by the TimeLOCKed application to retrieve the user's name. If the user has not purchased, the returned string will depend on the trial state: "New User", "Trial User", "Expired User".

This function will either return TRUE or FALSE, depending on the outcome of the *verifyTimeLOCK32* function. This authentication step is necessary to gain access to the information contained within the License Information (*.lif) file and the trial environment itself.

The size, in characters, of the buffer specified by the *strResult* parameter should be set to at least 50 to allow sufficient room in the buffer for the user's name.

Returns

TRUE on success
FALSE on failure

getUserNumDaysLeft

int getUserNumDaysLeft(LPCTSTR lpszPassword)

Parameter	**Description**
lpszPassword	Null-terminated string representing the password assigned by you to the License Information (*.lif) file in the Client Builder.

Comments

The *getUserNumDaysLeft* function is called by the TimeLOCKed application to return the number of days left in a trial.   This is applicable only if the trial session type has been set to expire in a certain number of days or to expire on a specific date. Otherwise, the return information from this function will be meaningless to the TimeLOCKed application.

Returns

If the function is successful, the return value is an integer value (0 or greater) representing the number of days left in the trial session.   If the return value is less than zero, it will be equal to one of the following 4 error constants.

TLOCK_ERROR_NOACCESSYET	-4
TLOCK_ERROR_FILEREAD	-3
TLOCK_ERROR_BADPASSWORD	-2
TLOCK_ERROR_FILENOTFOUND	-1

TimeLOCK API Error Constants

getUserNumExecutions

int getUserNumExecutions(LPCTSTR lpszPassword)

Parameter	**Description**
lpszPassword	Null-terminated string representing the password assigned by you to the License Information (*.lif) file in the Client Builder.

Comments

The *getUserNumExecutions* function returns the number of times the TimeLOCKed application has been executed (launched). Regardless of the type of trial selected (*i.e.,* Number of Days, Specific Date), this function will always return accurate information regarding the number of times this TimeLOCKed application has been run (as long as the TimeLOCKed application calls the *trialEnvironmentClose* function upon application shutdown).

Returns

If the function is successful, the return value is an integer value (0 or greater) representing the number of application uses in the trial session. If the return value is less than zero, it will be equal to one of the following four error constants.

TLOCK_ERROR_NOACCESSYET	-4
TLOCK_ERROR_FILEREAD	-3
TLOCK_ERROR_BADPASSWORD	-2
TLOCK_ERROR_FILENOTFOUND	-1

TimeLOCK API Error Constants

getUserNumMinutesUsed

long getUserNumMinutesUsed (LPCTSTR lpszPassword)

Parameter	**Description**
lpszPassword	Null-terminated string representing the password assigned by you to the License Information (*.lif) file in the Client Builder.

Comments

This function may be used to monitor usage time as the primary means of allowing or disallowing further use. Alternatively, it may be a second safety net designed to ultimately catch the situations where the end user continually backdates his system date for a TimeLOCKed application that is set to expire in a certain number of days.

This function accurately returns the exact number of minutes the TimeLOCKed application has been used cumulatively. In other words, the total usage time since the application was first loaded onto the end-user system is returned to the calling application.

Returns

If the function is successful, the return value is a long integer value (0 or greater) representing the number of application uses left in the trial session. If the return value is less than zero, it will be equal to one of the following five error constants.

TLOCK_ERROR_WRONGVERSION	-5
TLOCK_ERROR_NOACCESSYET	-4
TLOCK_ERROR_FILEREAD	-3
TLOCK_ERROR_BADPASSWORD	-2
TLOCK_ERROR_FILENOTFOUND	-1

TimeLOCK API Error Constants

getUserState

long getUserState(LPCTSTR lpszPassword)

Parameter	**Description**
lpszPassword	Null-terminated string representing the password assigned by you to the License Information (*.lif) file in the Client Builder.

Comments

Called by the TimeLOCKed application, this function performs user identification, returning the value representing the state of the user. For this function to succeed a prior call to the *verifyTimeLock32* function must have been successfully made. In other words, the authentication step is necessary to gain access to the information contained within the License Information (*.lif) file and the trial environment itself.

Typically, an application would provide or inhibit functionality based upon the state of the user or the error value itself. To do this, the original long integer values originally assigned to represent each of the four states of an end user (`NEW`, `TRIAL`, `EXPIRED` and `PURCHASED`) would need to be known. These values were assigned in the TimeLOCK Return Values step of the Client Builder.

The *getUserState* function is useful for trial implementations where the developer does not desire the display the TimeLOCK interface. Sub-system processes, operating system services or even expiring beta releases are all good uses for this *transparent implementation*. It is important to remember that a user cannot perform the steps necessary to purchase your application if they cannot access the TimeLOCK Purchase Wizard (shown when the user clicks the **Buy Now** button in the TimeLOCK Main Dialog). You could implement the *getUserState* function until the return value indicates Expired, and at that point, call *showMainDialog*.

Returns

If the function is successful, the return value will be positive (representing one of the four types of users), negative (representing one of the ERROR constants below) or equal to 0 (user pressed the Cancel button in the TimeLOCK Main Dialog).

TLOCK_ERROR_WRONGVERSION	-5
TLOCK_ERROR_NOACCESSYET	-4
TLOCK_ERROR_FILEREAD	-3
TLOCK_ERROR_BADPASSWORD	-2
TLOCK_ERROR_FILENOTFOUND	-1

TimeLOCK API Error Constants

showMainDialog

long showMainDialog(LPCTSTR lpszPassword)

Parameter	**Description**
lpszPassword	Null-terminated string representing the password assigned by you to the License Information (*.lif) file in the Client Builder.

Comments

Called by the TimeLOCKed application, this function determines the state of the TimeLOCKed application user, as well as display of the TimeLOCK Main Dialog. The Main Dialog will not be displayed in the event that the user has been identified as a PURCHASED user. Unless the end user has pressed the Cancel button in the Main Dialog, this function will always return either an ERROR constant or the state of the user.

Before the TimeLOCKed application can successfully make this function call, a previous call to *verifyTimeLOCK32* must have been made to authenticate both the TimeLOCK Runtime DLL and the TimeLOCKed application itself.

Typically an application would respond to this return value by providing or inhibiting functionality based upon the state of the user or the error value itself. To do this, the original long integer values that were assigned to represent each of the four states of an end user (NEW, TRIAL, EXPIRED and PURCHASED) would need to be known. These values were assigned in Client Builder.

Returns

If the function is successful, the return value will be positive (representing one of the four types of users), negative (representing one of the ERROR constants below) or equal to 0 (user pressed the Cancel button in the TimeLOCK Main Dialog).

TLOCK_ERROR_NOACCESSYET	-4
TLOCK_ERROR_FILEREAD	-3
TLOCK_ERROR_BADPASSWORD	-2
TLOCK_ERROR_FILENOTFOUND	-1

TimeLOCK API Error Constants

showMainDialogEx

long showMainDialogEx(LPCTSTR lpszPassword, HINSTANCE hInst)

Parameter	**Description**
lpszPassword	Null-terminated string representing the password assigned by you to the License Information (*.lif) file in the Client Builder.
HInst	Handle to the instance of the main application.

Comments

This function sets the instance handle for the TimeLOCK main dialog to hInst.
For more information, see “showMainDialog”.

Returns

If the function is successful, the return value will be positive (representing one of the four types of users), negative (representing one of the ERROR constants below) or equal to 0 (user pressed the Cancel button in the TimeLOCK Main Dialog).

TLOCK_ERROR_NOACCESSYET	-4
TLOCK_ERROR_FILEREAD	-3
TLOCK_ERROR_BADPASSWORD	-2
TLOCK_ERROR_FILENOTFOUND	-1

TimeLOCK API Error Constants

trialEnvironmentClose

BOOL trialEnvironmentClose()

Parameter	**Description**
none	N/A

Comments

Once the authentication is complete, the user is identified as NEW or TRIAL and the *trialEnvironmentOpen* function has been called, the TimeLOCKed application must close the trial environment when the application is exited or whatever specific functionality that was TimeLOCKed is exited.

This call also resets the an application-specific global access setting to FALSE, so all subsequent calls to the TimeLOCK API will result in a TLOCK_ERROR_NOACCESSYET or TLOCK_ERROR_BADPASSWORD return value until the TimeLOCKed application re-authenticates with the verify functions.

Returns

The function will return a Boolean value representing success or failure.   TRUE (nonzero value) indicates success while FALSE (zero) indicates failure.

trialEnvironmentOpen

BOOL trialEnvironmentOpen()

Parameter	**Description**
none	N/A

Comments

Once the authentication step is successful and the subsequent determination of the state of the user is complete, the TimeLOCKed application should then make a call to the *trialEnvironmentOpen* function only if the application is still in a TRIAL mode. If this is not a trial user, there is no need to start up system timers and increment usage counters (which this function performs).

Returns

The function will return a Boolean value representing success or failure. TRUE (nonzero value) indicates success while FALSE (zero) indicates failure.

verifyTimeLock32

int verifyTimeLock32(LPCTSTR lpszFileName, LPCTSTR lpszPassword, LPTSTR strResult)

Parameter	Description
lpszFileName	Null-terminated string representing the file name of the License Information (*.lif) file created by the Client Builder.
lpszPassword	Null-terminated string representing the password assigned by you to the License Information (*.lif) file in the Client Builder.
strResult	Address of buffer for TimeLOCK ID Key that TimeLOCK will return to your application for authentication of the TimeLOCK Runtime DLL. This value or string should be exactly the same as the string you provided in the Client Builder.

Comments

Before the TimeLOCKed application can gain access to the information in the License Information (*.lif) file, it must verify or authenticate itself by calling this function and providing the original password as a parameter.   If the password is found to be authentic to the associated trialware, TimeLOCK will fill in the *strResult* buffer parameter with a 16-25 digit string that was entered by the developer in the Client Builder TimeLOCK Keys step. The TimeLOCKed application uses this return code to verify and authenticate the TimeLOCK Runtime DLL.

Returns

The function will return an integer value representing success or failure.   An integer value of 1 indicates success while an integer value less than 0 would indicate one of the following values.

TLOCK_ERROR _WRONGVERSION	-5
TLOCK_ERROR_NOACCESSYET	-4
TLOCK_ERROR_FILEREAD	-3
TLOCK_ERROR_BADPASSWORD	-2
TLOCK_ERROR_FILENOTFOUND	-1
TLOCK_ERROR_SUCCESS	1

TimeLOCK API Error Constants

TimeLOCK API Error Constants

Below is a table of the six error values that may be returned to the TimeLOCKed application upon completion of one of the TimeLOCK API functions. It is recommended that all TimeLOCK function calls made by the TimeLOCKed application result check these return values and take appropriate action in response.

Named Constant	Value	Description
TLOCK_ERROR_WRONGVERSION	-5	The License Information (*.lif) file that you are trying to access was created with a version of the Client Builder that does not allow access to the TimeLOCK API function call you are using.
TLOCK_ERROR_NOACCESSYET	-4	A TimeLOCK function call is being made that requires prior access verification. This can be accomplished by performing the Authentication. Once *trialEnvironmentClose* is called, the TimeLOCKed application will always need to re-authenticate before gaining access to the API and the information about the current user.
TLOCK_ERROR_ FILEREAD	-3	The License Information (*.lif) file is either corrupt or cannot be opened for read/write access.
TLOCK_ERROR_BADPASSWORD	-2	The password that has been sent to the function is not the same as the current password in the License Information (*.lif) file.
TLOCK_ERROR_ FILENOTFOUND	-1	The License Information (*.lif) file cannot be found in either the local directory or the Windows directory.
TLOCK_ERROR_ SUCCESS	1	Successful result.

Glossary

In this section we define the key concepts related to using TimeLOCK for Electronic Software Distribtion (ESD).

API

Applications Programming Interface.

Authentication

The process of verifying that your application is communicating with a valid version of the TimeLOCK runtime DLL. Also known as verification.

Authorization

The ability of a user to access a feature of a software application.

Bag Of Bits (BOB) File (*.bob)

The Bag of Bits (BOB) file is an encrypted file used in a multi-tier distribution model. Its purpose it to ensure the integrity of the License Information File and TimeLOCKed executable as they are passed to channel partners.

Channel Partner

A vendor who markets, distributes, or supports a product developed by a software publisher. Channel partners are usually one of the following: distributor, merchant, or clearinghouse.

Clearinghouse

An organization that processes credit card purchases for software that is distributed electronically. In some distribution schemes, the clearinghouse will also supply keys to unlock TimeLOCKed software.

Client Builder

A TimeLOCK tool used to integrate applications, select trial parameters, and purchase and security options. The Client Builder is used by the publisher and its channel partners.

Client Interface

The screens and dialogs that appear to a customer when using a TimeLOCKed application.   The Client Interface typically displays trial and purchasing information.   Also known as End-User Interface.   See also Main Dialog and Purchase Wizard.

Copy Protection

An option offered in the Client Builder that lets you limit the number of times your TimeLOCKed software can be installed.

Customer

An individual or organization who uses a software application on a trial or purchased basis.

Digital Signature

An electronic file which identifies the owner and/or author and/or contents of an electronic document.   Also known as a Signature.

Direct Distribution

The case where a publisher sells their products directly to end-users.

Distributor

An organization that maintains a relationship with a software publisher to sell the publisher's products electronically either directly or through a third-party merchant.

Dynamic Link Library DLL

Dynamic Link Library supported by the Windows environment. See your Microsoft Windows documentation for details.

Electronic Software Distribution

A method for distributing software applications electronically. (e.g., using the Internet, CD-ROM, fax, e-mail, modem.)

Electronic Unlocking Key

A unique code provided by the clearinghouse to unlock a TimeLOCKed application. The key is written to the License (*.lic) file to indicate that the application has been purchased. The electronic unlocking key is protected from counterfeiters.

Encryption

A Public/Private key mechanism to provide security.   An RSA scheme is used to encrypt TimeLOCK files to protect applications being distributed electronically.

End User

The user of a software application

End User Environment

The system used by your customers when running your application.   See also Client Interface or End-User Interface.

End-User Interface

The screens and dialogs that appear to a customer when using a TimeLOCKed application. The End-User Interface typically displays trial and purchasing information. Also known as the Client Interface. See also Main Dialog and Purchase Wizard.

End-User License Agreement (EULA)

A license agreement supplied by a software publisher that states the terms and conditions under which an end-user may use a purchased product.

Exact Usage Time

One of the five types of trials offered in the TimeLOCK Client Builder. It controls the trial period by keeping track of exact usage time for the TimeLOCKed software.

Expire on Specific Date

One of the five types of trials offered in the TimeLOCK Client Builder.   It controls the trial period by keeping track of the date on which the trial period should expire.

Expired Trial Message

A message displayed to a customer who tries to run a TimeLOCKed application after the trial period has expired.

Extended API

A group of special functions used for Programmed Integration to customize the Client Interface.

Hash Function or Hash Code

A mechanism which is used to protect the executable (*.exe) file of the TimeLOCKed application from tampering. Using hash codes ensures that even a sophisticated user cannot edit the executable to bypass authorization checks.

Instant Integration

One of the two methods for creating TimeLOCKed applications (the other is Programmed Integration). Instant Integration inserts TimeLOCK protection mechanisms into the application by using DLLs.

Integrity

The ability of a software application to prevent its unauthorized use.   TimeLOCK uses encryption schemes to protect the integrity of your TimeLOCKed application.

Key

A string of digits used to encrypt or data.   TimeLOCK uses keys to protect components of your TimeLOCKed application from unauthorized use.

Key Generation

A mechanism for generating the keys used by TimeLOCK.

Key Module

The part of TimeLOCK used by customer service staff to “unlock” TimeLOCKed software and convert it to purchased software.

Key Pair

The RSA Data Security scheme used by TimeLOCK requires both a public key and a private key be used to encrypt and decrypt protected information. The public and private keys are referred to as a key pair.

License File (*.lic)

This file contains a customer’s license information such as trial usage and purchase status.   This file is created from information in the License Information File (*.lif) used by the TimeLOCK Client Builder.

License Information File(*.lif)

This file is created and modified when the publisher, distributor, or merchant uses the TimeLOCK Client Builder.

Main Dialog

The dialog box that appears each time a user executes a TimeLOCKed software application.   Once the user purchases the application, this dialog no longer appears.

Main Dialog Message

The message displayed to the user in the Main Dialog of the TimeLOCK Client Interface.   This message is generated from information entered in the TimeLOCK Client Builder.

Manufacturer

The organization that creates, develops, and maintains the original application.   Also known as the publisher.

Markers

Internal codes set by TimeLOCK to manage the TimeLOCKed software environment.

Merchant

The organization responsible for selling software applications electronically. In a Direct Distribution scheme, the publisher also acts as a merchant.

Multi-tier Distribution

The case where a publisher sets up channel partners for distributing and/or selling software.

New User Message

A customized message that appears the first time a user runs the TimeLOCKed software application.

No Limits

One of the five types of trials offered in the TimeLOCK Client Builder.   It provides an unlimited trial period.

Number of Days

One of the five types of trials offered in the TimeLOCK Client Builder.   It controls the trial period by keeping track of the number of days on which the TimeLOCKed software application has been run.

Number of Executions

One of five TimeLOCK trial types.   It controls the trial period by keeping track of the number of times the TimeLOCKed software application has been run.

Password

A secure set of characters that is used to access the Client Builder for use by the publisher or channel partners.

Phone Unlock Dialog

A special dialog shown to customer service staff when providing an unlocking key to a customer who has purchased the software by telephone, fax, or mail.

Phone Unlocking Key

A key that may be distributed by telephone after a customer has purchased the TimeLOCKed application. This key does not provide the same level of protection as an Electronic Unlocking Key.

Private Key

The electronic file generated by the RSA Data Security scheme known only by one individual user. It is used to encrypt and decrypt information to be sent via secure channels.

Privilege File (*.prv)

This file contains information that specifies what type of information you are authorized to create or modify when using the TimeLOCK Client Builder. (E.g., publisher, distributor, or merchant.) This is created and distributed by Preview Software.

Product

A software application and associated materials that are created and maintained by an organization.

Product ID

A unique, 12-character string that uniquely identifies your product.

Product Key File (*.prd)

The public key associated with the product.   This file is created and distributed by Preview Software.

Programmed Integration

One of the two methods for creating TimeLOCKed applications (the other is Instant Inegration).   Programmed Integration requires that a Developer use the TimeLOCK API to integrate an application for Electronic Software Distribution.

Public Key

The electronic file generated by the RSA scheme that is owned by an individual user but is known by everyone.   It is used to encrypt and decrypt information to be sent via secure channels.

Publisher

The organization that creates, develops, and maintains the original application.   Sometimes known as the Manufacturer.

Purchase Wizard

The set of dialog boxes that appear when a customer selects the Purchase option from the Client Interface. The information in the Purchase Wizard is generated from information entered in the Client Builder. See also Client Interface.

Purchase Instructions

Instructions in the Client Interface which are customized by you using the Client Builder.

Purchase Only

A method for distributing software electronically. In this case, an application must be purchased before being used by a customer.

Purchased User

A user who has purchased your application.

Registration Number

A unique number generated by TimeLOCK to identify each user.

Reset Trial

An option offered in the Reset Wizard, which lets you clear all markers from your system so you can re-test the Phone Unlock mechanism.

Reset Wizard

A TimeLOCK tool that lets you test the phone unlock mechanism.

Security Keys

Special keys maintained by TimeLOCK to prevent unauthorized access to your application. These keys are distributed by Preview Software.

Server

A central computer used by an organization to store core information.   It may be accessed by numerous client interfaces.

Sign

The act of encrypting electronic information with a Private Key.

Signature

An electronic file which identifies the owner and/or author and/or contents of an electronic document.   Also known as a Digital Signature.

TimeLOCK API

The Applications Programming Interface (API) that provides access to all the TimeLOCKed software functionality in the TimeLOCK DLL.

TimeLOCK DLL

The Dynamic Link Library (DLL) that provides all of the TimeLOCKed software functions used by TimeLOCK.

TimeLOCK ID Key

A sequence of 24 numeric characters that TimeLOCK will return to your application to verify the identity of the TimeLOCK runtime DLL. This prevents someone from substituting an unauthorized DLL.

TimeLOCKed application

A software application to which TimeLOCK's Instant or Programmed Integration has been applied.

TimeLOCKing

The process of integrating a software application with TimeLOCK functionality using either Instant or Programmed Integration. Also known as “wrapping.”

tl30api.dll

The TimeLOCK 3.0 DLL that is distributed with an application if Programmed Integration is used.

tlinj30.dll

The TimeLOCK 3.0 DLL that is distributed with an application if Instant Integation is used.

Transaction

The act of passing information from one entity to another.   (E.g., when a customer purchases software from a merchant, a transaction occurs when the customer sends his credit card number to the merchant.)

Transparent Implementation

Using the TimeLOCK API to provide TimeLOCK functionality without displaying the Client Interface.

Trial User License Agreement (TULA)

A license agreement supplied by a software publisher that states the terms and conditions under which an end-user may use a product on a trial or evaluation basis.

Trial Restore Code

A special code provided to a customer who wishes to extend their trial period.

Trial Return Codes

Special codes returned to your TimeLOCKed software application to indicate the current state of the user (i.e., NEW, TRIAL, PURCHASED, EXPIRED).

Trial Type

A mechanism for controlling the trial period. TimeLOCK 3.0 supports five different trial types: Number of Executions, Number of Days, Expire on Specific Date, No Limits, and Exact Usage Time.

Trial User Message

A customized message that appears in the Main Dialog for a user whose state is TRIAL.

Try Before You Buy

A method for distributing software where an application may be used on a trial basis without cost. This allows prospective customers to evaluate the software before purchasing.

Unlock Code or Unlock Key

A unique code provided by your customer service staff to a user who has purchased your software. The Unlock code updates the License (*.lic) file to show that the TimeLOCKed software was purchased. See also Phone Unlocking Key and Electronic Unlocking Key.

User

The individual who runs a software application.

User State

One of four states used to determine the type of user (NEW, TRIAL, PURCHASED, EXPIRED) when running a TimeLOCKed application.

Validation

The act of determining whether a user has access to a particular software component.

Verification

The process of verifying or authenticating the TimeLOCK DLL being used by the TimeLOCKed software application.

Wrapping

The process of integrating a software application with TimeLOCK functionality using either Instant or Programmed Integration. Also known as “TimeLOCKing.”

TimeLOCK Integration Methods

Before describing how to begin TimeLOCKing an application, it is important to understand the two TimeLOCK Integration methods.

Instant Integration allows TimeLOCK's capabilities to be added to the product by the publisher, the distributor, and the merchant. Instant Integration is the simplest way to prepare TimeLOCKed applications.

Instant Integration inserts TimeLOCK protection mechanisms into the application by using DLLs. TimeLOCK's Client Interface displays information about the TimeLOCKed application based on information entered in the Client Builder utility.

Programmed Integration allows you to customize the Client Interface to suit your own needs by adding TimeLOCK's capabilities to your application's source code. This approach uses TimeLOCK's API functions. This method must be used if the software product checks to see if the source code was modified. A product of this type is a virus checker. For these products, Instant Integration may not be used.

At runtime, the calling information can obtain the state of the trial environment through calls to the TimeLOCK API. The developer, at design time, will produce code that conditionally executes based upon the returned information from TimeLOCK about the trial environment.

You should understand how your application is to be marketed and sold before settling on an integration method. The type of end-user will also help to determine the integration strategy. Once you know this, you are ready to determine the TimeLOCK integration method. Below we compare the two methods and lists some of the tradeoffs involved with each.

Programmed Integration
(requires access to the software product's source code)

§ Developer can design the look and feel of the purchase dialog boxes seen by the customer.
§ Developer may design a more restricted trial version. (E.g., software's print function is disabled until the product is paid for.)
§ Programmed Integration can be used with software with built-in virus checking and virus check software products, such as McAfee.

Instant Integration
(requires access to the software product's executable file)
§ Instant Integration is performed automatically by the Client Builder; no coding is necessary.
§ A developer is not needed to perform Instant Integration. This enables the application to be easily customized for different marketing and distribution strategies.

Notes:
§ The Purchase Wizard may not be customized outside of user messages.
§ Integration must be done on the application executable. It cannot be done on the setup.exe file.

Steps to Integrating TimeLOCK with Your Application

The steps to integrating TimeLOCK with an application will vary depending on the distribution model being used and the way the product is distributed. Under a direct distribution model, the publisher will perform all of the functions for TimeLOCKing software. Multi-tier distribution will require that the publisher, distributor, and merchant all perform certain functions for TimeLOCKing. Try Before You Buy typically requires more steps be performed than Purchase Only distribution. In the *Example Scenarios* section in this chapter, you will see what steps may be required for your distribution model.

Before You Begin

Before a user begins to use TimeLOCK to integrate an application they must assemble a number of pieces of information. Each organization is responsible for a particular set of information. Following is a list of tasks that must be completed before beginning to use the TimeLOCK software. These are listed according to which organization is responsible for the task.

Publisher Functions

1. **Assemble application files (*.exe)**: The publisher must prepare the application files including the application executable. Under certain scenarios, the publisher will also need to prepare the installation files for the application using a third party tool such as InstallShield or Wise.

2. **Obtain and distribute Privilege (*.prv) and Product Key (*.prd) Files**: The publisher must obtain the necessary Privilege (*.prv) and Product Key (*.prd) files necessary from Preview Software. These files are used by TimeLOCK to determine what TimeLOCKing utilities a user may implement in a particular application. The Product and Privilege files are discussed in detail in the *Introduction* chapter.
   - § Direct Distribution: The publisher needs Privilege (*.prv) and Product Key (*.prd) files. These must be obtained from Preview Software.
   - § Multi-Tier Distribution: The publisher must obtain Privilege (*.prv) files from Preview software for any merchants and distributors as well as the publisher. The publisher must also see that every channel partner is given a copy of the Product Key (*.prd) file.

3. **Obtain and distribute the Clearinghouse Key File (*.clr):** If the product is to be purchased electronically over the Internet, the publisher must obtain a clearinghouse Key File (*.clr) from Preview Software. This file is described in detail in the *Introduction* chapter. The publisher must give a copy of this file to appropriate distributors.

4. **Determine the integration method**: TimeLOCK supports two different integration methods: Instant Integration and Programmed Integration. Instant Integration incorporates TimeLOCK functionality directly into your application. Programmed Integration generates source code that a developer may use to incorporate into the application source code. The type of integration method you choose will depend on what kind of application is being sold and what type of end-user will be using the application. In the *TimeLOCK Integration methods* section, we will describe each method in detail to help you determine which is appropriate for your software.

5. **Determine how the setup scripts will be modified**: If you are going to distribute your software on a Try Before You Buy basis using Instant Integration, you will need to determine whether you can modify the setup scripts used in installation. This information will be used when running the Client Builder.

6. **Create any necessary Trial and End-User License Agreements**: The publisher may define the Trial and End-User License Agreements that must be accepted by a user before running a trial or purchasing the application.

7. **Assemble publisher information:** The publisher must supply information on itself and the product to the TimeLOCK Client Builder. This information will be presented to the end-user in the Client Interface.

8. **Trial Parameters:** If the product will be used on a Try Before You Buy basis, the publisher must specify under what conditions a trial will expire. The following parameters are available:
   - § Unlimited Trial
   - § Number of executions.
   - § Number of days.
   - § Exact usage time.
   - § Expiration on a specific date.

   These options are defined in the *Glossary* and in the *Client Builder* chapter.

9**. Copy protection and security options**: The publisher may choose to implement some security mechanisms to prevent unauthorized use of the application. One or more of the following may be used:
   - § Software Binding
   - § Hardware Binding
   - § Embedded Serial number
   - § Tamper Proof Application

   Note: If none of these options are selected, the original application executable will be restored when the application is unlocked. Details on these security mechanisms are given in the *Client Builder* chapter.

Distributor Functions

1. **Assemble distributor information**: The distributor must provide information on the distributor organization.

2. **Transaction Method:** Select the transaction method that customers can use to purchase the application. Two methods are supported: Purchase over the Internet and by phone, fax, or e-mail. The distributor may choose one or both of these methods.

Merchant Functions

1. **Assemble merchant information**: The merchant must provide information on the merchant organization including support contacts. This information will be displayed to the customer when running the application or making a purchase.

2. **Determine pricing**: The merchant must provide the price and its expiration date. This information will be displayed to the customer when using the Purchase Wizard.

Integration Functions

Once you have answered the questions relevant to your organization, you are ready to integrate TimeLOCK with your application. Integration consists of a few simple steps:

1. **Use the Client Builder** to enter the information particular to your organization. (E.g., publisher, distributor). The Client Builder is discussed in the next section. A complete description is given in the *Client Builder* chapter.

2. **Distribute files to channel partners (Multi-tiered Distribution only).** Give the new License Information (*.lif) file and application executable (*.exe), to the appropriate channel partner. In some cases a self-extracting executable (typically called setup.exe) will also be necessary.

3. **Modify source code (Programmed Integration only).** If Programmed Integration is used, you must edit your source code to customize the TimeLOCKed application. This includes defining the Client Interface.

4. **Test the TimeLOCKed application**. The Reset Wizard may be used to help in testing. See the *Reset Wizard* chapter.

5. **Prepare TimeLOCKed application files for customers**: Depending on the distribution model and integration method, you may need to modify installation files or prepare a new self-extracting executable. See the *Example Scenarios* section for details. You will need to include the following files: the application executable (*.exe), the License (*.lic) file, and the appropriate TimeLOCK DLL (tl30inj.dll or tl30api.dll).

6. **Distribute application files to customers**.

Using the Client Builder

The Client Builder is used to provide company and product information, customize the parameters of the trial, choose security options to prevent purchased products from unauthorized use, and select the TimeLOCK integration method. Full instructions on using the Client Builder are given in the *Client Builder* chapter.

When:

§ Instant Integration is selected, the Client Builder creates a new application executable containing TimeLOCK functionality and a License (*.lic) file to be used by the TimeLOCK Runtime DLL to create and maintain the TimeLOCK application.

§ Programmed Integration is selected, the Client Builder will generate source code in either C/C++, Delphi, or VisualBasic for use with your application.

If Instant Integration is used, you are ready to test the application once the Client Builder is completed. Refer to the chapter on the *Reset Wizard* for further information. Programmed Integration requires additional steps as discussed below.

Additional Steps for Programmed Integration

Now that you have used the Client Builder to generate appropriate source code, you are ready to begin integrating some function calls from the TimeLOCK API. The TimeLOCK functions should be accessed at the beginning of your application or process to open the TimeLOCKed environment. Then, when the user quits the application, the TimeLOCKed environment is simply shut down. The chapter titled *Using the API* gives full information on using the API set, and a complete function reference is included in the chapter titled *Function Reference.*

Note: You must use the Client Builder for Programmed Integration to generate the hash code for the TimeLOCKed application's executable (*.exe) file. This hash code ensures that the executable is safe from tampering.

The Client Interface

The Client Builder will generate the source code for the default end-user interface. If the application you want to TimeLOCK is a sub-system process, background application or service, it may not be appropriate to use the TimeLOCK Client Interface's Main Dialog each time at start up or even for purchase. You may prefer to display a custom dialog (from within your application itself) or, if the product is a beta release, you may simply want to let the trial period expire without ever displaying a dialog. TimeLOCK provides the flexibility to support TimeLOCK functions without requiring the display of the standard Client Interface. This is referred to as a *transparent implementation*.

To bypass the display of the Main Dialog, your application can call one of the User Information functions in the TimeLOCK API to find the number of days, hours or executions left in the trial session. You can also determine the state of the user (NEW, TRIAL, PURCHASED, or EXPIRED) by calls to API functions. This information can then be used to prepare messages for the end user at runtime, or simply to prevent execution of the application if the trial period has expired.

In designing your custom interface, remember that your user will benefit from knowing how long the trial period is, and how much longer he has to work with the application before the trial expires. For more information see the section called *Customizing the Main Dialog* in the chapter titled *Using the TimeLOCK API.*

Note: Once you are through developing the code for Programmed Integration of your application, you will need to run the Client Builder again to generate the secure hash code for your executable.

Once the TimeLOCK API is integrated, you are ready to test the application.

Test the Integration

To test the application, you should first create an installation routine and install the software on a few machines internally. It is a good idea to ask personnel unfamiliar with TimeLOCK to test your TimeLOCKed application. Listen to what they say and track down any errors. If you need to make changes in the interface, you can use the Client Builder to generate a new License (*.lic) file for your product.

TimeLOCK also includes a Reset Wizard that will allow you to reset trial information for your TimeLOCKed application. This is useful when doing a lot of testing that causes trials to expire or changes the user status to PURCHASED or EXPIRED. See the *Reset Wizard* chapter for more information.

Prepare Customer Installation Files

In some situations, the Client Builder may be used to prepare the self-extracting executable that may be distributed directly to customers. In these cases, a customer simply runs this executable to install the TimeLOCKed application and appropriate license file.

In other cases, you may need to use third-party software to create an installation file. In these cases, you will need to include the following in your installation files:

§ TimeLOCKed executable (*.exe)

§ License file (*.lic)

§ TimeLOCK DLL: tl30inj.dll for Instant Integration; tl30api.dll for Programmed Integration

Example Scenarios

Direct, Purchase Only Distribution, Instant Integration
Direct, Try Before You Buy Distribution, Instant Integration
Multi-Tier, Purchase Only Distribution, Instant Integration
Multi-Tier, Try Before You Buy Distribution, Instant Integration
Direct, Try Before You Buy Disrtibution, Programmed Integration

Direct, Purchase Only Distribution, Instant Integration

The publisher assumes all the roles of publisher, distributor, and merchant to distribute an application on a Purchase Only basis. The following steps should be completed in the order given:

1. The publisher prepares the application executable and obtains Privilege (*.prv) and Product Key (*. Prd) files from Preview.

2. The publisher uses the Client Builder to apply Instant Integration. The Client Builder will create a new TimeLOCKed executable (*.exe), License Information (*.lif) and License (*.lic) files.

3. The publisher creates the installation files using the new executable (*.exe) , License (*.lic) file, and TimeLOCK DLL (tl30inj.dll) with a third-party tool such as InstallShield or Wise.

Direct, Try Before You Buy Distribution, Instant Integration

The publisher assumes all the roles of publisher, distributor, and merchant to distribute an application on a Try Before You Buy basis. The following steps should be completed in the order given:

1. The publisher prepares the application executable and obtains Privilege (*.prv) and Product Key (*. Prd) files from Preview.

2. The publisher uses the Client Builder to apply Instant Integration. The Client Builder will create a new TimeLOCKed executable (*.exe), License Information (*.lif) and License (*.lic) files. It also creates the self-extracting setup executable.

3. The publisher distributes the setup file, typically called setup.exe, to customers.

Multi-Tier, Purchase Only Distribution, Instant Integration

The publisher, distributor, and merchant are distinct organizations who support a product distributed on a Purchase Only basis. The following steps should be completed in the order given:

1. The publisher creates the application's installation file using a third-party product such as InstallShield or Wise.

2. The publisher uses the Client Builder to configure the publisher section of the License Information (*.lif) file. He also creates a BOB file and self-extracting executable.

3. The publisher passes the self-extracting executable (*.exe), BOB and LIF files to the distributor.

4. The distributor uses the Client Builder to configure the distributor section of the LIF file.

5. The distributor passes the self-extracting executable (setup.exe), BOB and LIF files to the merchant.

6. The merchant uses the Client Builder to

§ configure the merchant portion of the LIF file.
§ specify the BOB file.
§ create a LIC file and integrate it with the application executable and TimeLOCK DLL (tl30inj.dll).

This produces a self-extracting executable (*.exe) file ready to download from the merchant's Web site.

Multi-Tier, Try Before You Buy Distribution, Instant Integration

The publisher, distributor, and merchant are distinct organizations who support a product distributed on a Try Before You Buy basis. The following steps should be completed in the order given:

1. The publisher uses the Client Builder to configure the publisher section of the License Information (*.lif) file and create a new executable using Instant Integration.

2. The publisher creates the application’s installation file using a third-party product such as InstallShield or Wise.

3. The publisher uses the Client Builder to create the BOB file from the installation file.

4. The publisher passes the self-extracting executable (*.exe), BOB, and LIF files to the distributor.

5. The distributor uses the Client Builder to configure the distributor section of the LIF file.

6. The distributor passes the self-extracting executable (*.exe), BOB, and LIF files to the merchant.

7. The merchant uses the Client Builder to

§ configure the merchant portion of the LIF file.

§ specify the BOB file.

This produces a self-extracting executable (*.exe) file ready to download from the merchant’s Web site.

Direct, Try Before You Buy Distribution, Programmed Integration

The publisher uses Programmed Integration to distribute an application on a Try Before You Buy basis. The following steps should be completed in the order given:

1. The publisher uses the Client Builder to create an LIF file and generate the source code for the TimeLOCKed application.

2. The publisher integrates the source code generated by the Client Builder into the application.

3. The publisher uses the Client Builder to create the LIC file.

4. The publisher creates the installation files using the new executable (*.exe), License (*.lic) file, and TimeLOCK DLL with a third-party tool such as InstallShield or Wise.