Terazona

Version 1.4.1

Installation and Configuration Guide

3350 Scott Blvd.,
Santa Clara, CA 95054
http://www.zona.net/

Contents

Introduction

Welcome to the Terazona Installation and Configuration Guide. This document demonstrates how to deploy Terazona in a variety of deployment configurations.

Audience

This guide is intended for people who will install Terazona.

Document Conventions

This guide uses a variety of formats to identify different types of information.

Convention	Function
`courier`	Identifies syntax statements, on-screen computer text, and path, file, drive, directory, database, and table names.
`<courier>`	Identifies variable names.
`bold courier`	Identifies text you must type.
italics	Identifies document and chapter titles, special words or phrases used for the first time, and words of emphasis.
underline	Identifies URLs, domain names, and email addresses.
Initial Caps	Identifies Window, menu, command, button, option, tab, keyboard, and product-specific names.
ALL CAPS	Identifies acronyms and abbreviations.
[ ]	Identifies an optional item in syntax statements.
{ }	Identifies an optional item that can be repeated as necessary within a syntax statement.
>	Identifies a separation between a menu and an option.
|	Identifies a separation between items in a list of unique keywords when you may only specify one keyword.

Special Message Conventions

Identifies information that will help prevent system failure or loss of data.

Identifies information of importance or special interest, including Notes and Tips.

Menu Conventions

This guide uses the **Menu > Option** convention. For example, "**Click Format > Style**" is a shorthand instruction for "Click the Format menu, then select the Style option."

Mouse Conventions

To select something, place the on-screen pointer or cursor on the item and click the left mouse button.

To view an **Options** menu, place the on-screen pointer or cursor on an item and click the right mouse button (or left mouse button if using a left-handed mouse). If a menu is available, it will open. (Clicking the right mouse or left mouse button is referred to in this guide as *option-click*.)

When the term *click the mouse on...*is used without qualification, it means to place the on-screen pointer or cursor on an item and click the left mouse button.

To drag something, click the mouse on it and drag the pointer to a different location before releasing the mouse button.

When selecting items from a list using the mouse, you can sometimes select more than one item by holding down the **Shift** or **Control** key while clicking the mouse.

To select a contiguous block of items, click on one item, hold the **Shift** key down, and click on a second item. All items between the two will be selected when multiple selection is enabled.

To select items from different locations when multiple selection is enabled, hold the **Control** key down. Each selected item will remain selected until you complete the action or click the mouse without holding the **Control** key down.

Additional Help

For additional information or advice, contact:

Contact Information United States

Phone	+ (408) 844 9646
Facsimile	+ (408) 844 9647
Internet	http://www.zona.net/
Email	Strategic Partners: strategicpartners@zona.net Developers: gamedevelopers@zona.net Recruitment: jobs@zona.net Information: contact@zona.net
Postal	Shanda Zona, LLC., 3350 Scott Blvd. #23, Santa Clara, CA 95054-3104

List Of Figures

Part

I

Installing Terazona

This part of the Terazona Installation and Configuration Guide demonstrates how to install Terazona in a variety of deployment configurations.

Chapter

1

Installing Terazona 1.4.1

This document contains a step-by-step procedure for installing Terazona v1.4.1. We strongly recommend that you completely uninstall all older Terazona versions before installing this most up-to-date version.

Outlining Terazona

Terazona v1.4.1 consists of multiple software components, which together comprise the various servers and development platforms required for client and server development. and run-time execution.

You can install the following components from the installation CD:

- Server components, including:
  - Authentication Server
  - Auditing Server
  - Messaging Server (iPush)
  - ZAC Server
  - Dispatcher Server
  - Sphere Server
  - Game State Server (GSS)
  - Game Database Server
  - Audit Database Server
  - Licensing Server (WIBU-KEY)
  - Game Guild Server (GGS)
- Developer Components
- Client Applications
- Server Components

The decision to Audit Terazona operations is optional, and you can decide to selectively Audit some, all, or no Entities. However, the Audit Database Server must be at least created and installed within the Terazona server cluster. However, unless you do decide to turn on the Auditing function, the Audit Database Server will not affect Terazona's run-time behavior.

Configuring the Windows Pre-Install Environment

The Terazona Server 1.4.1 development and runtime environments require Windows 2000 Server or Professional with Service Pack 4. The Terazona Client Applications can execute on Windows 2000 Server or Professional. They can also execute on Windows XP with Service Pack 1.

Minimum hardware requirements for Terazona v1.4.1 installation are as follows:

- Pentium III (600MHz or faster)
- 300MB disk space for a complete installation (that is, Server Components, Client Development, Server Development)
- 256MB RAM or higher

Before you run the Terazona installer you must first install the following third party tools:

- Java (JRE) – Terazona Server machines require Sun Microsystem's Java Runtime Environment (JRE) version 1.4.2. The executable file to install the JRE is bundled on the installation CD as **`\3rdParty\Java\Windows\j2re-1_4_2-windows-i586.exe`**. You can also download the JRE from our ftp site or from Sun Microsystems' website.
- Java (JDK) – Terazona Client development environments require Sun's Java Development Kit (JDK) version 1_4_2. The JDK is only available for download from Sun Microsystem's website.

A typical developer configuration is for one or more Terazona Servers to execute on a single machine that is also used as a Terazona Client development environment. In this case, install the JDK.

- Microsoft DirectX (Runtime) – DirectX Runtime version 8.1 is required for Clients to execute ZonaBattle or any of the Win32 graphical demos. You can download the DirectX Runtime 8.1 from Microsoft. Windows XP ships with DirectX Runtime 8.1.
- Microsoft DirectX (SDK) – The Microsoft DirectX SDK is also required if you are installing the Terazona Client development environment, and is available for download from Microsoft.

- Database Server – Terazona currently supports the following database client/server configurations:
  - Microsoft SQL Server 2000 SP3a must be installed on your designated database server before you can begin installation. To provide the required JDBC support, Microsoft SQL Server 2000 Driver for JDBC Service Pack 1 must be installed on each server machine that will require access to either the Game Database or the Audit Database.
  - MySQL 4.0.16 must be installed on your designated database server before you can begin installation. Terazona currently supports the default MySqld-nt (Windows) using MyISAM tables.

    To provide JDBC support, MySQL Connector/J 3.0 must be installed on each server machine that will require access to either the Game Database or the Audit Database.

The machines that require a JDBC client driver are the ZAC, Sphere Server, Game State Server, Game Guild Server and Audit Server machines. The Installer will prompt you to ensure you have downloaded appropriate database drivers.

We currently recommend avoiding a heterogeneous database architecture and using a single matched database client/server configuration throughout your Terazona cluster for all databases. Choose a single database architecture for all Terazona databases.

- WIBU-Systems –– Terazona v1.4.1 is shipped with the WIBU-KEY copy protection system. The machine that runs as the WIBU-KEY server is known as the Licensing Server. Currently only Windows is supported as a host platform for the WIBU-KEY server. Prior to installation, you must deploy such a Licensing Server as a physical WIBU-BOX, accessible within your network. WIBU Software Protection System version 4.00 is located on the installation CD, at **`3rdParty\Wibu\Windows`**.

After installation you can set the WIBU-KEY as a service and link any additional installation machines to the machine where the WIBU-KEY is installed, as described in *Linking Windows WIBU-KEY Client Machines on page 27*.

- Adobe Acrobat Reader – You must have Acrobat Reader version 5.0 (or later) installed in order to read the **`PDF`**-based Terazona documentation. You can download the latest version from Adobe's website.

Installing Terazona

The installation CD contains the installation application, which is a self-installing Java Archive (**`.jar`**) file (built with InstallShield MultiPlatform Edition).

Before installing a multi-server Terazona cluster, please familiarize yourself with the required Terazona servers. Because you must enter specific IP addresses for certain servers during the install, you should create an installation map and pre-allocate certain machines and IPs.

During the course of the installation process, you will decide which components you wish to install – server components, client development, or server development. If you are installing Terazona server components (Messaging Server, ZAC Server, and so on), you must know the IP addresses and/or virtual names of the target installation machines prior to installation. If you are installing a database server for a Terazona client, you will need to know your target database information (name, password, etc.)

1. Ensure that Java, DirectX, and WIBU-KEY are installed and deployed, as per the pre-installation requirements.
2. Ensure that your **`%JAVA_HOME%`** environment variable is set correctly:
   - Terazona Servers - JRE 1.4.2 directory.
   - Terazona Client development environments - JDK 1.4.2 directory.
3. Login as a user with administrator/superuser privileges on your target installation machine.
4. Navigate to the installer file, **`ZonaServer1.4.1.jar`**, located at the root directory of the installation CD (or within the FTP Site). Click or double-click the jar file to cause it to execute.

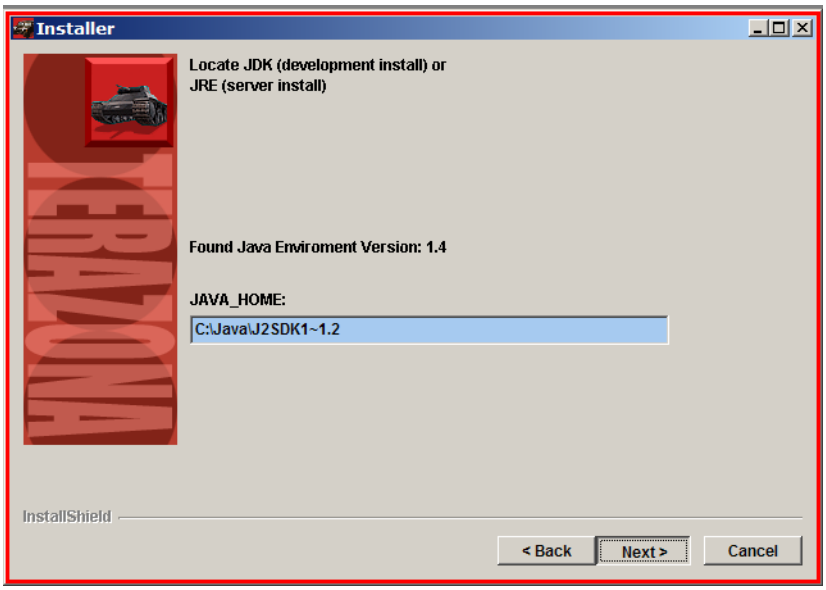
The Terazona load screen displays initially:

Figure 1-1. Terazona Load Screen

5 Next, a welcome screen displays. Click the Next button to get started. A License Agreement will display. Please read the License Agreement, then click the Next button.

6 A confirmation page for your Java installation directory appears. Once the directory appears, click Next.

Figure 1-2. Specifying the JRE Location

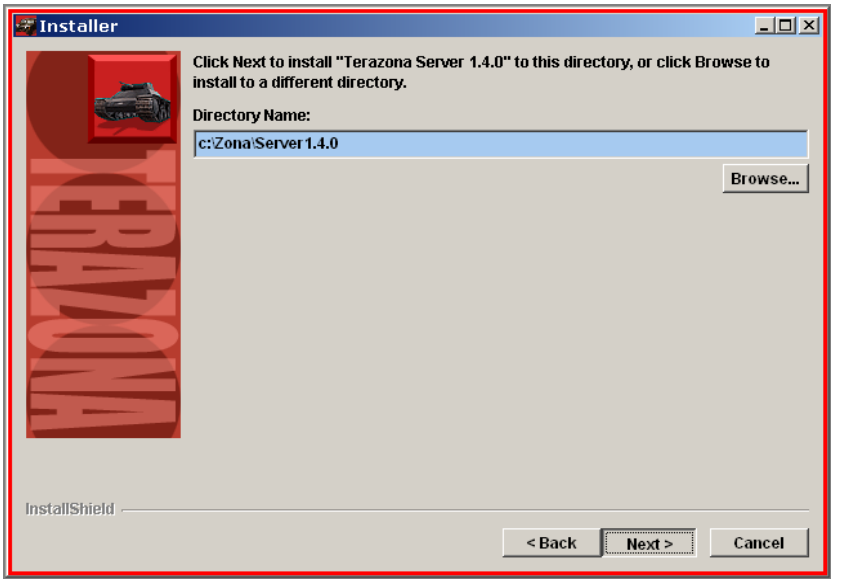


Within Windows environments, the Java install directory name must conform to a DOS-style 8.3 naming convention and contain no spaces. Additionally, the directory path must contain no spaces. White space in the filename or directory path may cause some of the Terazona support utilities to behave unpredictably. The tilde (**~**) abbreviation operator is acceptable to build conforming pathnames, as are dashes (**-**) and underscores (_).

Terazona Servers - JRE 1.4.2 directory.
Terazona Client development environments - JDK 1.4.2 directory.

7 Enter the directory where you want to install Terazona v1.4.1. Click Browse to navigate through the directories on your system. Once you have selected a directory, click Next.

Figure 1-3. Defining the Terazona Install Directory

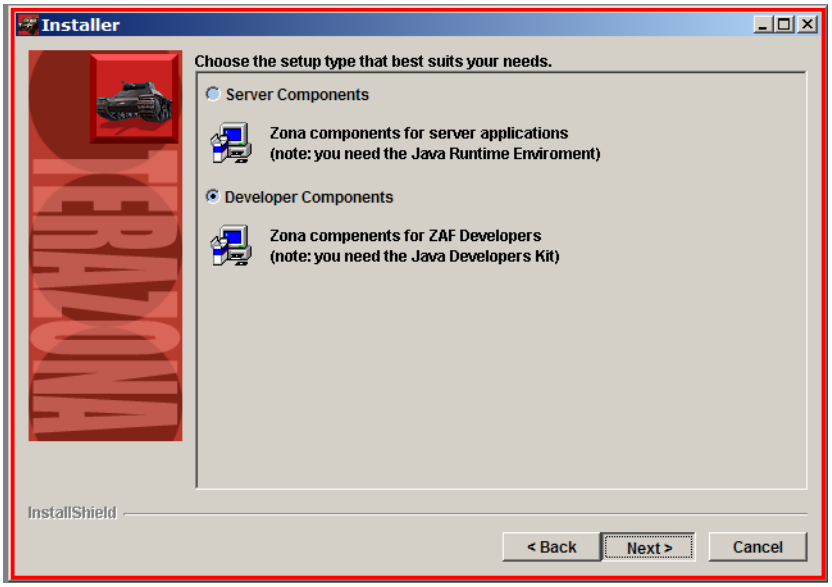


Within Windows environments, the Terazona install directory name must conform to a DOS-style 8.3 naming convention and contain no spaces. Additionally, the directory path must contain no spaces. White space in the filename or directory path may cause some of the Terazona support utilities to behave unpredictably. The tilde (**~**) abbreviation operator is acceptable to build conforming pathnames, as are dashes (**-**) and underscores (_).

8 Select the type of installation you wish to perform and then click Next:

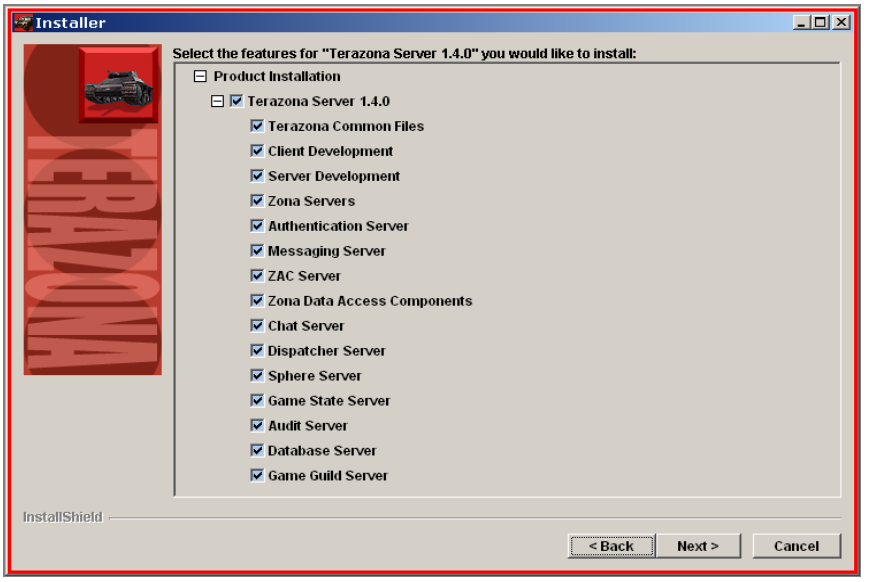
- Server Components – Installs the Terazona servers (subsequent panels display each server type as an option), as well as allowing the option to install client development components (also selected on subsequent screens).
- Developer Components – The most complete install; creates a Terazona Client development environment. This options installs every component required for both client and server development.

Figure 1-4. Defining the Install Type



9 Click Next to display the component selection screen.

Figure 1-5. Selecting the Terazona Features



If you are installing the Server Components, select the ones that you wish to install from the list. If you wish to install the Server Components, select those as well.

Each Terazona server uses a separate messaging service. The Messaging Server, also known as an iPush Server, acts as a communications pipe or information bus between servers.

The Zona Administration Controller (ZAC) Server is the central administration server which keeps track of all Terazona processes. Terazona clients connect to the Dispatcher Server, which is a connection point for all users logging into the system. It provides load balancing for players and routes their requests to a Game State Server.

Game State Servers (GSSs) manage the connections and flow of information between the clients. The Game State Servers are designed to accommodate several hundred concurrent users. To support more users, you may need to install additional GSSs as necessary (that is, re-run the installer). The Sphere Server is responsible for balancing the load between different Game Servers. Game Guild Servers (GGSs) manage the flow of information between members of Game Guilds. In a minimum server system configuration the ZAC, Dispatcher, Game State, and Game Guild servers can run on the same machine.

The Auditing Server must be installed within the Terazona server cluster but you do not have to activate the Auditing function unless desired. We recommend deploying the Game Database and the Auditing Database on separate database servers to optimize performance. See *Scoping the Auditing Server on page 63* for details.

A option for the Licensing Server, or WIBU-KEY machines, is not displayed as part of the install process. As part of the pre-install process you should have designated a machine as the WIBU-KEY server and deploy the WIBU-KEY server software and hardware dongle on that machine.

If you are installing Terazona v1.4.1 for the first time, we recommend you install one Messaging Server for the Dispatcher and ZAC servers, and one Messaging Server for the Game State Server and Sphere Servers. This enhances system performance.

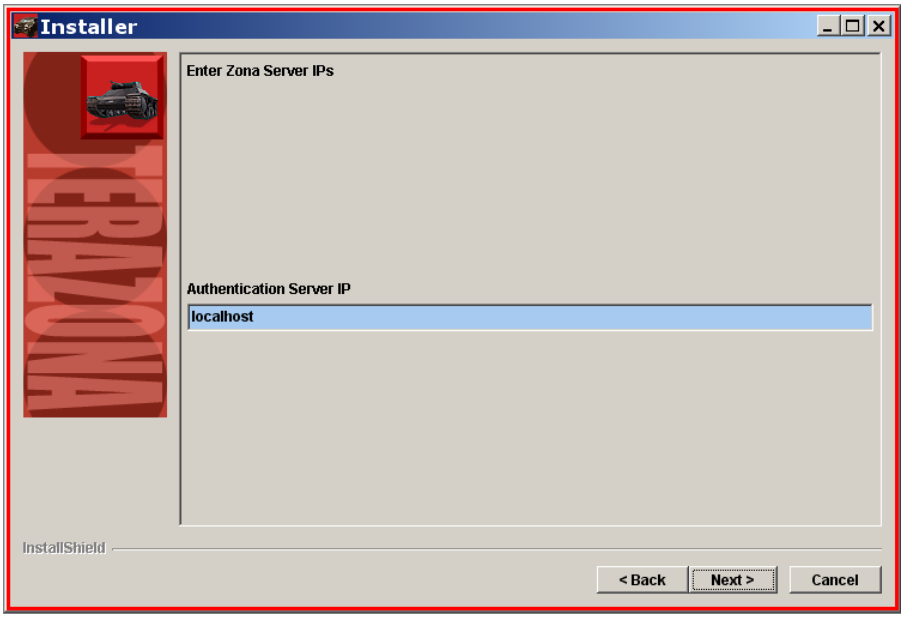
You will have to re-run the installer to install a new Messaging Server on a new machine. Each machine (specifically, each NIC) supports only one Messaging Server. The Messaging Server currently deploys only on Windows machines.

If you are installing the Sphere, Chat, or GSS servers, you must install the Terazona Database Access Components on those machines. If you are installing the Database Servers, the Terazona Database Access Components will be installed automatically on this machine. Terazona Common Files are always installed, regardless of which installation type you select.

To designate this machine as an Authentication Server, select the "Authentication Server" option. You will be presented with extra screens where you enter the IP addresses of machines within the Terazona cluster. See *Configuring Terazona Security on page 57* for details. The Authentication Server currently deploys only on Windows machines.

10 Install the Authentication Server and specify its IP address.

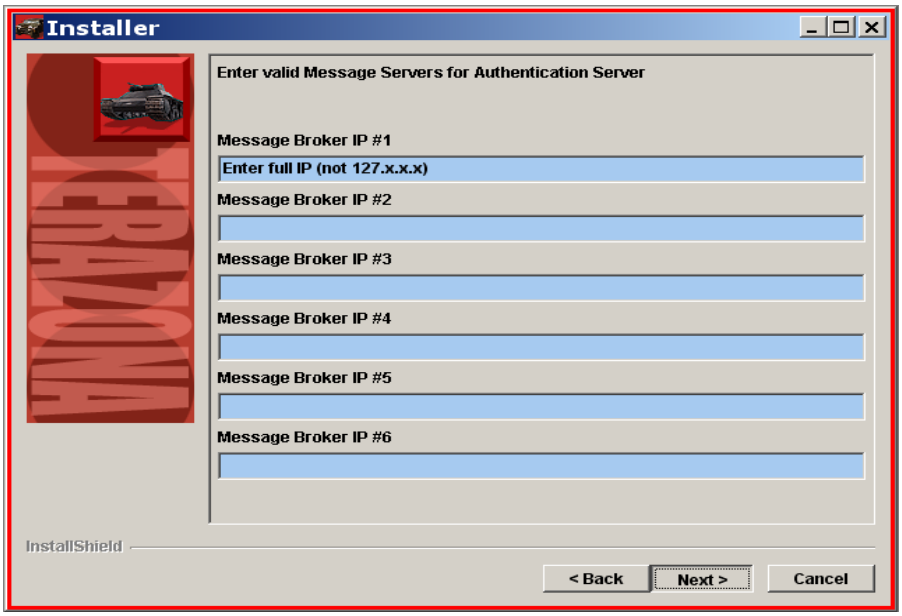
Figure 1-6. Entering the Zona Server IPs - Authentication Server



The Authentication Server currently deploys only on Windows machines.

11 Identify the Messaging Server IP addresses for the Authentication Server.

Figure 1-7. Specifying the Authentication Messaging Server IPs



The above IP addresses for each iPush server correspond to each specific Messaging server that you install. So for example, if you assign iPush Machine X to your ZAC and Dispatcher servers, during installation you must enter that IP address above and also enter that IP address (or corresponding DNS name) and port number of Machine X in the ZAC and Dispatcher server screens, when prompted.

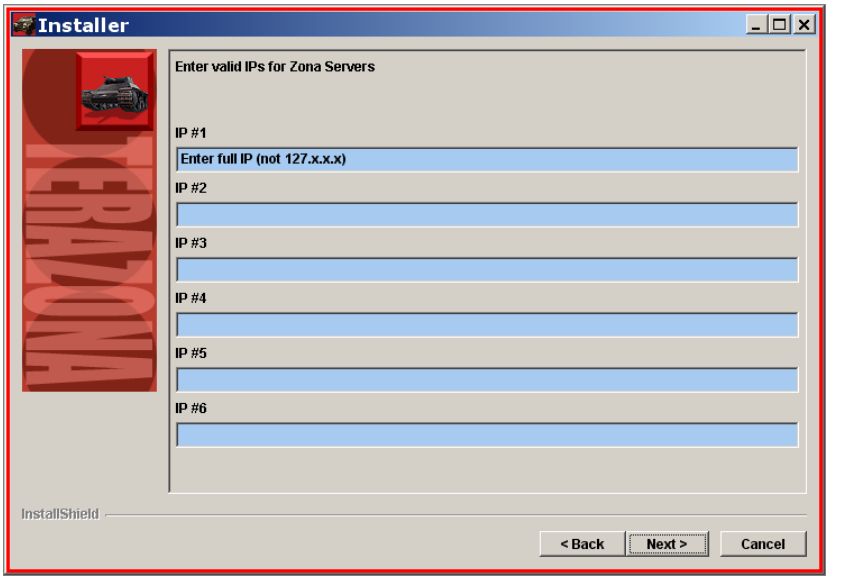
If you assign a different iPush Machine Y for your GSS and Sphere servers, during installation you must enter that IP address above and also enter that IP address (or corresponding DNS name) and port number of Machine Y in the GSS and Sphere server screens, when prompted.

You must add the real IP address that is bound to each machine's NIC (and not the classic pseudo-IP address for a localhost of **`127.0.0.1`**). On Windows machines, at a command prompt use the **`ipconfig`** command to identify your IP address. For more details, see *Configuring Terazona Security on page 57*.

A dialog will warn you if you enter an incorrectly formatted IP address. However, this is a syntax checker only and will not warn you if an IP is unreachable, firewalled, or otherwise unsuitable.

12 The Trusted Server IPs dialog displays. Enter the IP addresses of "trusted" Terazona machines. See *Understanding Terazona Security on page 57* for how to manually add extra Trusted Servers after the install process.

Figure 1-8. Entering the Trusted Server IP Addresses

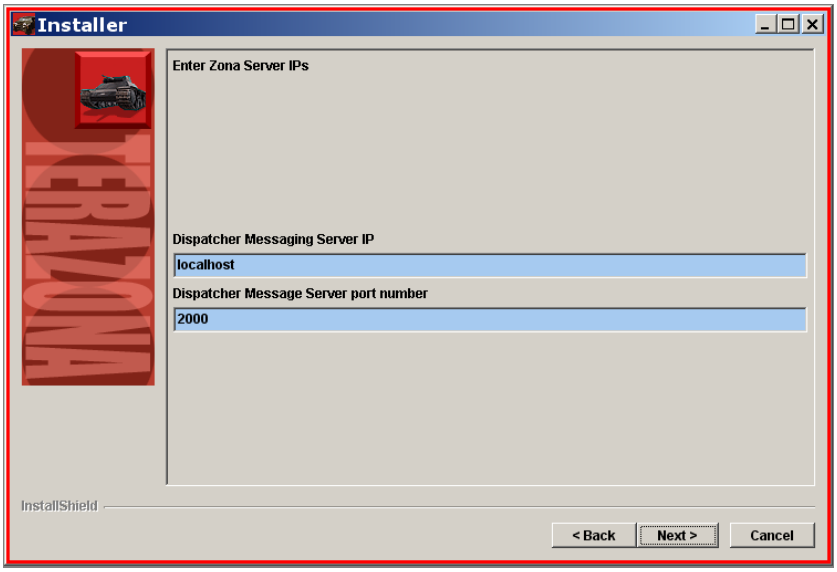


You must add the real IP address that is bound to each machine's NIC (and not the classic pseudo-IP address for a localhost of **`127.0.0.1`**). On Windows machines, at a command prompt use the **`ipconfig`** command to identify your IP address. For more details, see *Configuring Terazona Security on page 57*.

A dialog will warn you if you enter an incorrectly formatted IP address. However, this is a syntax checker only and will not warn you if an IP is unreachable, firewalled, or otherwise unsuitable.

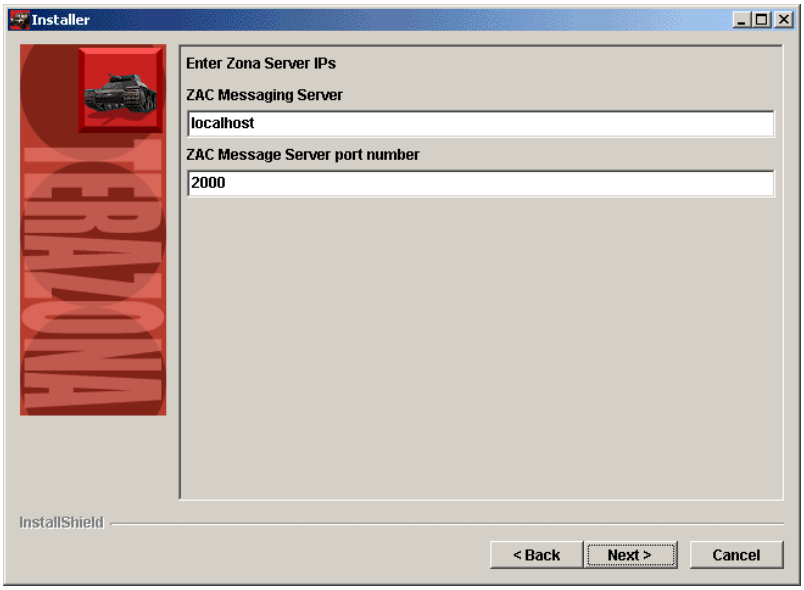
13 Specify the Dispatcher Messaging Server IP address.

Figure 1-9. Specifying the Dispatcher Messaging Server IP



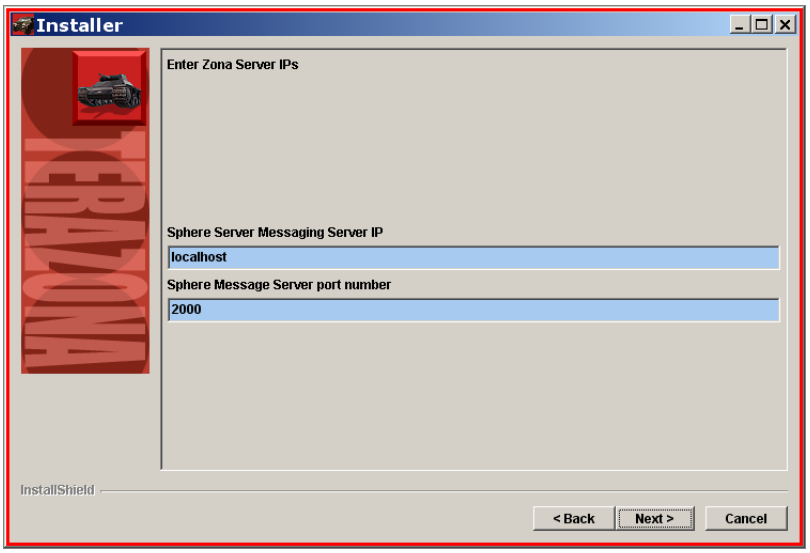
14 Specify the ZAC Messaging Server IP address.

Figure 1-10. Entering the Zona Server IPs - ZAC Messaging Server



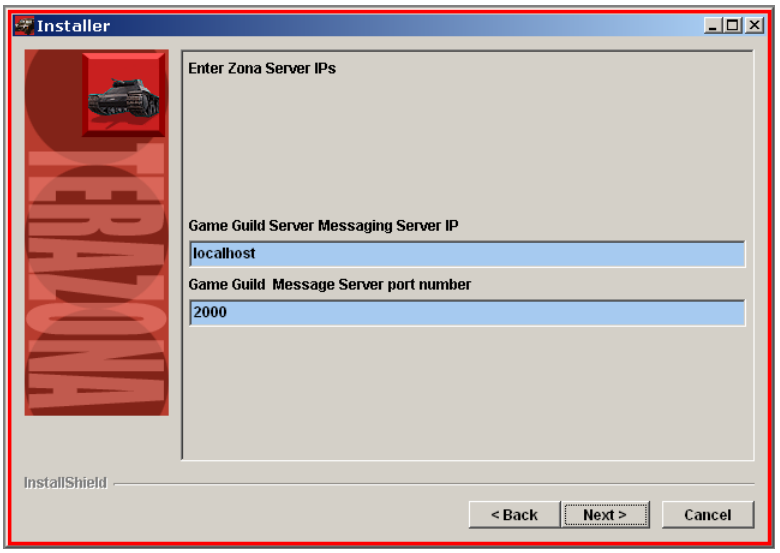
15 Specify the Sphere Server IP address.

Figure 1-11. Entering the Zona Server IPs - Sphere Server



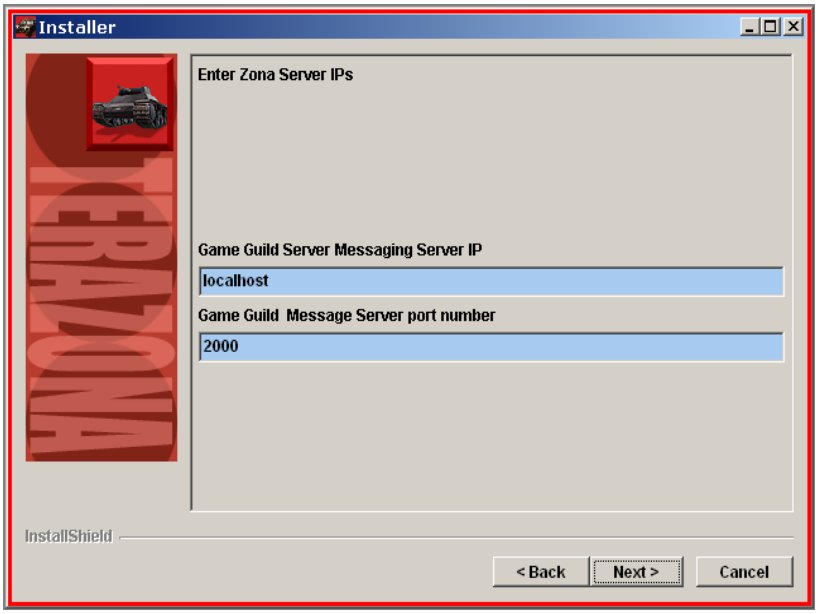
16 Specify the Game Guild Server Messaging Server IP address.

Figure 1-12. Entering the Zona Server IPs - Game Guild Server



17 Specify the GSS Messaging Server IP address.

Figure 1-13. Entering the Zona Server IPs - GSS Messaging Server



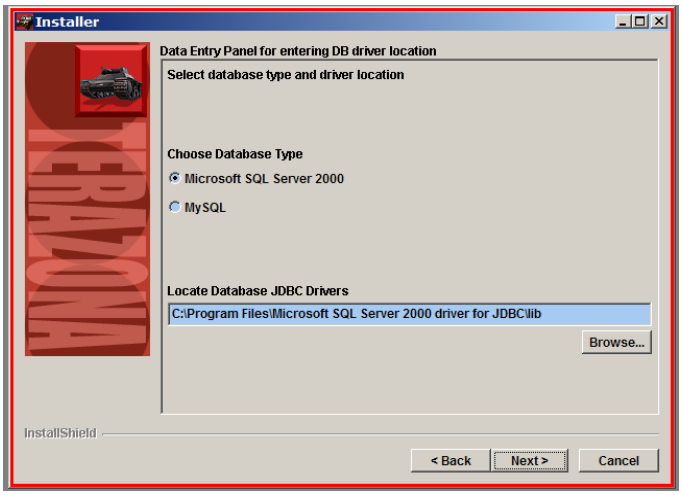
18 You must have installed an appropriate JDBC driver on each server machine that will require access to either the Game Database or the Audit Database. These machines are the ZAC, Sphere Server, Game State Server, Game Guild Server or Audit Server machines. The Installer now prompts you to ensure you have downloaded the correct driver. If you have not yet installed this driver, please do so.

Figure 1-14. Ensuring You Have Installed the Correct JDBC Driver

- Microsoft SQL deployment requires SQL Server 2000 Driver for JDBC Service Pack 1 version 2.2.0029 on all clients.
- MySQL deployment requires MySQL Connector/J version 3.0 on all clients.

19 You should manually enter the JDBC driver location. Click the **Browse...** button to locate this driver yourself. After the correct JDBC driver location has been entered, click the **Next** button to continue.

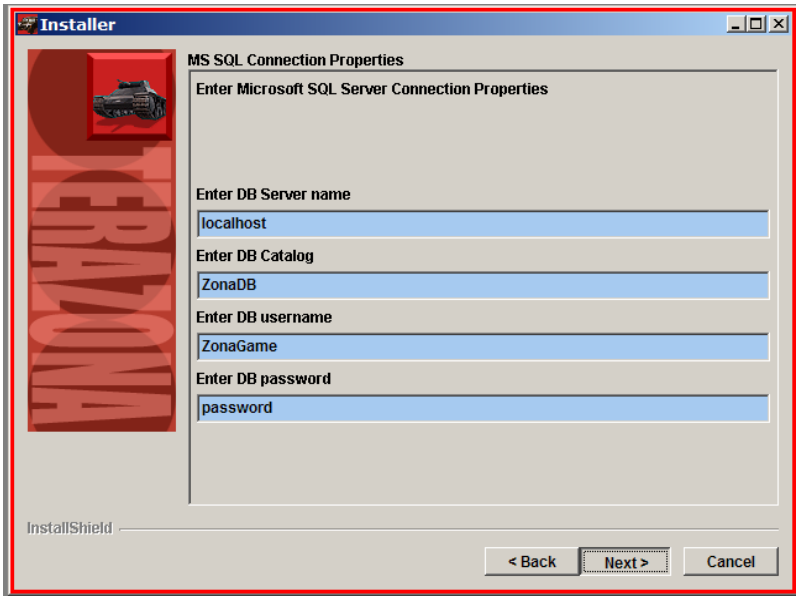
Figure 1-15. Selecting the Database Type and Location



These installation instructions assume that you have chosen to install the MS SQL drivers. If instead you choose to install a different JDBC driver, then the Installer screens will be slightly different and will reference your chosen JDBC driver and database platform.

20 If you selected to install the Database Server and Terazona Database Access Components, you will now need to enter your Database Server name. Leave the DB Catalog field set to **`ZonaDB`**, the DB username field set to **`ZonaGame`**, and the DB password field set to **`password`**.

Figure 1-16. Entering the Database Connection Properties

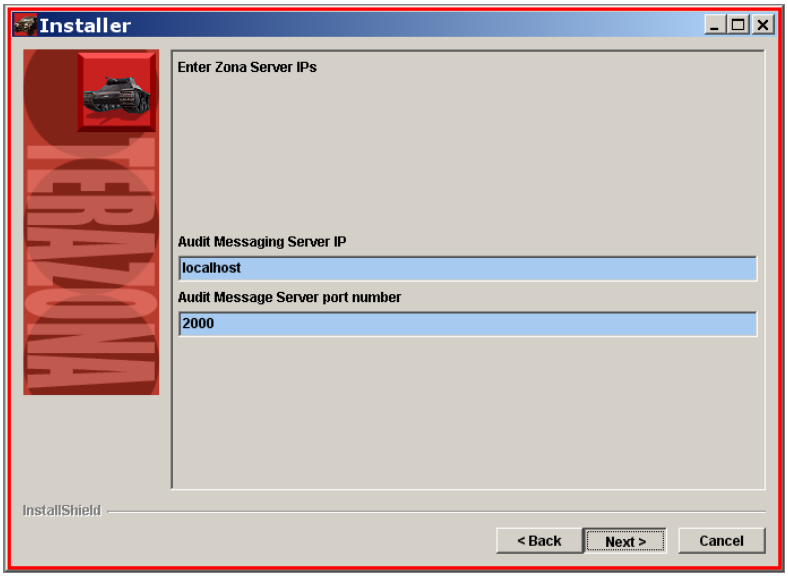


Your Database Server is installed on the machine hosting your SQL Server, and will be used by the other servers. Batch files and SQL scripts are created in the **`%ZONA_HOME%\sql\MSSql`** directory (if MS SQL was selected), and are used to create a database catalog (called **`ZonaDB`** in the default configuration) to store Terazona objects. This is known as the Game Database. For more details on creating the Game Database and Terazona objects and users, see *Creating the Game Database on page 32*.

21 Confirm your installation settings and then click Next. If you selected this option, the Auditing Server configuration screens now display.

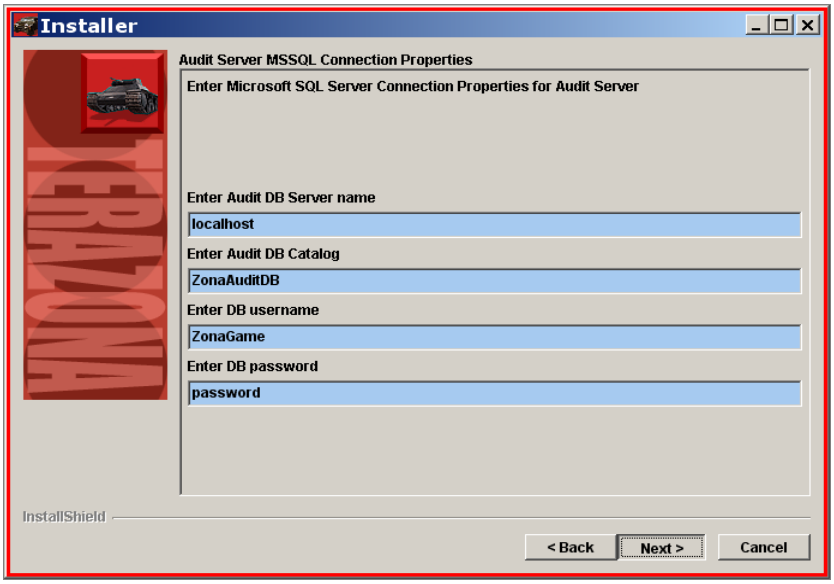
22 Specify the Auditing Server IP address.

Figure 1-17. Entering the Zona Server IPs - Audit Messaging Server



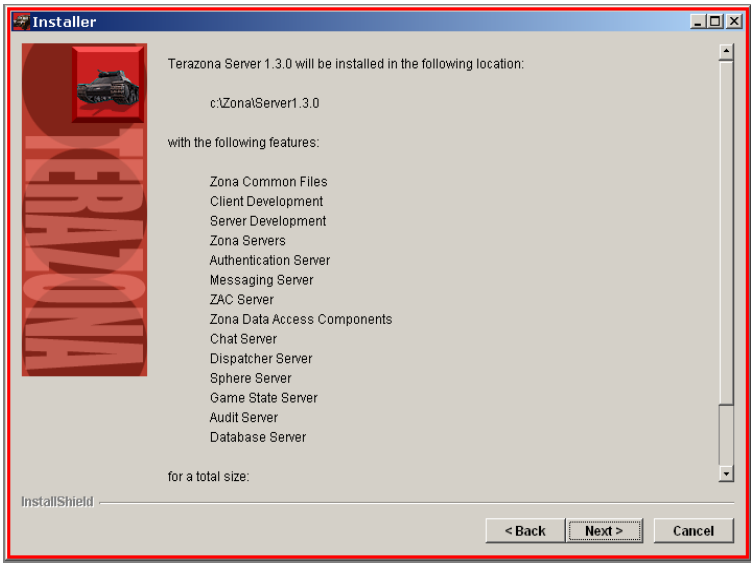
23 The Auditing Server MSSQL Connection Properties dialog displays.

Figure 1-18. Entering the Auditing Server Database Connection Properties



24 When the Installer has finished running, it displays “Finish”. Click the **Finish** button to complete the installation.

Figure 1-19. Checking the Terazona Installation Options



Configuring the Post-Install

When you install the ZAC, Dispatcher, GSS, and Sphere servers (using the same iPush server), your deployment preferences are stored on each server in the **`%ZONA_HOME%\config\zona.xml`** file. This details the configuration properties of the installation – port numbers, database name, and so on. Once you have installed Terazona v1.4.1 on your system, you must then:

1 Configure your WIBU-KEY copy protection system.

2 Ensure that several environment variables have been set.

3 Create a database schema for the Terazona Game Database, and create required Users for the game.

4 Create a database schema for the Audit Database Server.

5 Build the Samples, then configure and run the clients and servers.

Configuring the WIBU-KEY

When you install Terazona, some of the server files are license-protected using encryption. To validate your license and decrypt the server files during run-time execution, the installer deploys the WIBU-KEY application.

Each Terazona server cluster requires at least one machine designated as a WIBU Server. This is the Licensing Server. The WIBU Driver and WIBU-KEY USB dongle are installed on the Licensing Server. All other machines running Terazona server processes must have the WIBU-KEY run-time driver client software installed. The WIBU-KEY software on these client machines must be configured to search for the WIBU Server, or Licensing Server.

On Windows machines, the Installer uses a default value for the target installation directory (that is, **`c:\program files\wibukey`**). WIBU-KEY installs several shortcuts in the Program Manager (**Start > Program**) and also inserts an application icon in the Control Panel (S**tart > Settings > Control Panel**).

The WIBU-KEY application works together with a hardware-based USB dongle. This dongle is configured to support your Terazona license terms, and notifies the WIBU-KEY software to enable Terazona functionality throughout your cluster. The dongle can be installed locally or remotely on a network-reachable machine. The machine where the dongle is physically installed is the Licensing Server.

During server startup, if WIBU-KEY is not running or cannot contact a Licensing Server (or if it detects a problem with your license terms) then you will see WIBU-KEY exceptions in the server console displays. The servers will not initialize successfully.

Once you have installed WIBU-KEY on your desired Licensing Server machine (for example, Machine X), you must start the Network Server service on Machine X. Other machines running the WIBU-KEY client software can validate their server components during startup.

Additionally, you must set the following WIBU-KEY configuration settings:

- Linking Machines
- NT/Windows Service Setting
- Timeout Setting

You should also know how to cancel a WIBU-KEY session if a client crashes before the time out period.

Installing Your WIBU-KEY License

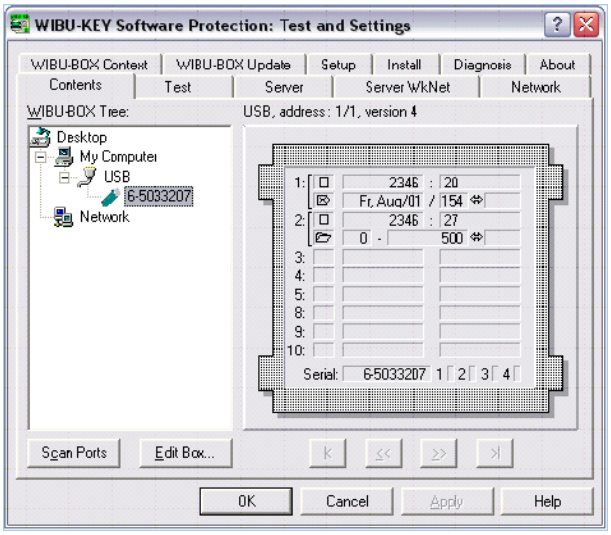
You do not need to take special steps to install a license. Terazona ships with the WIBU dongle pre-configured to support your customer-specific license. You must install this dongle on a machine and designate that machine as the WIBU-KEY Licensing Server. You can then link other machines throughout your organization to this WIBU-KEY Licensing Server, up to the limits imposed by your license terms.

Verifying Your WIBU-KEY License

To verify your WIBU-KEY License (on the machine with the WIBU-KEY dongle physically installed):

1. Select **Start > Settings > Control Panel**.
2. Double-click the WIBU application icon, and then click the Contents tab.
3. The contents of the WIBU-KEY displays in the right-hand panel:

Figure 1-20. WIBU Key Contents



Linking Windows WIBU-KEY Client Machines

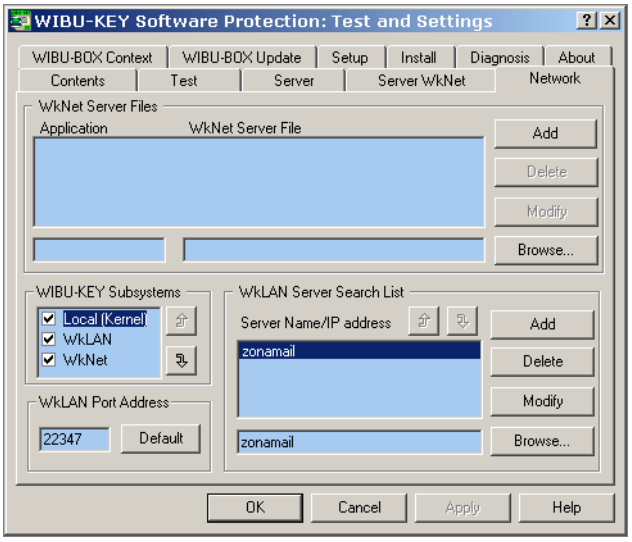
Suppose you installed WIBU-KEY on Machine X, and your server components are installed on Machine Y. You will need to link WIBU-KEY to Machine Y by configuring Machine Y as follows:

[**Machine Y**]

1. Select **Start > Settings > Control Panel**.
2. Double-click the WIBU application icon, and then click the Network tab.
3. In the WkLAN Server Search List section, delete all items in the Server Name/IP Address list.
4. Click Browse to scan for WIBU servers. When you find Machine X, click Add to add it to the search list.

Machine Y is now set up to look for the WIBU key installed on Machine X, and you can start the server on Machine Y.

Figure 1-21. Linking the Machines For WIBU-Key Configuration



You must manually add the server's hostname or IP address into the WkLAN Server Search List. You can use the **Browse...** button to search the network for WIBU servers. Failure to add the server explicitly will cause WIBU-KEY to hang during Terazona startup.

Configuring the WIBU-KEY Windows Service Setting

On Windows machine(s) where WIBU-KEY is installed, we strongly recommend that you enable the WIBU-KEY application to automatically start as a Windows service when the machine is powered on. To do this:

1. Select **Start > Programs > WIBU-KEY > Network Server.**
2. Select Install as service, then click OK.
3. Restart your machine for the setting to take effect. Note that should you ever move your WIBU-KEY to a different machine, you will need to reconfigure this setting.

Figure 1-22. Installing WIBU-Key as a Windows Service

Before deploying WIBU-KEY, you must complete the preceding *Linking Windows WIBU-KEY Client Machines* process of linking your machines to the WIBU-KEY License Server machine where the dongle is physically attached.

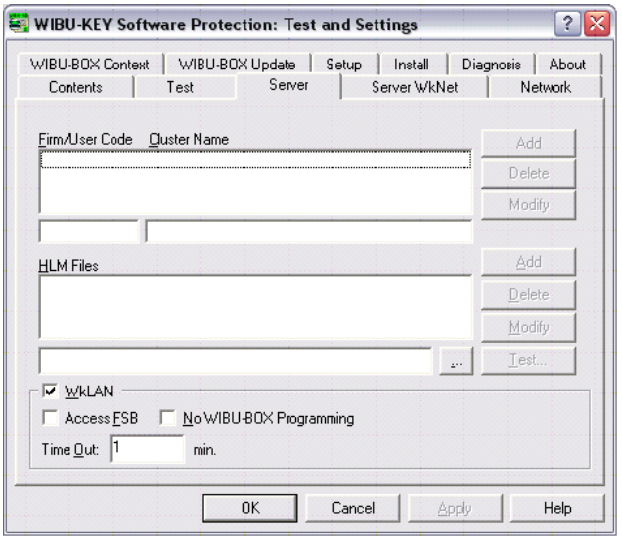
Setting the WIBU-KEY Time Out

We recommend that you set your Licensing Server's WIBU-KEY application to automatically time out clients after a certain period of inactivity (this is because the WIBU-KEY system can only execute a limited number of processes and crashed processes "block" the available process slots until timed-out). For development machines you should set the timeout to a low value, perhaps 60 seconds. To do this:

1 Select **Start > Settings > Control Panel**. Double-click the WIBU application icon, and then click the **Server** tab.

2 Enter a time out value of 1 minute, and then click **OK**.

Figure 1-23. Setting the WIBU-Key Timeout



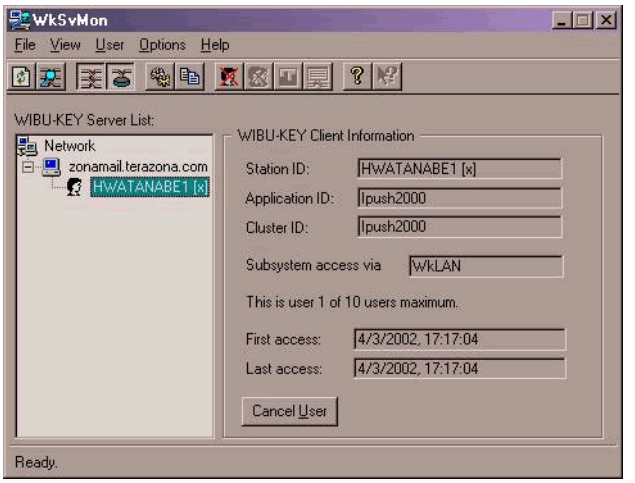
Production servers require a longer WIBU-KEY timeout value than development servers. We recommend setting a timeout period of 3-5 minutes. You may want to adjust this for your own game parameters and system performance.

Cancelling a WIBU-KEY Client Session

If you are experiencing problems running Terazona servers, the client machine may have accidentally crashed, but the Licensing Server may still think the client is alive until the timeout period is reached. To cancel a client session, using the Licensing Server's WIBU-KEY application:

1. Start the Server-Monitor application (**`WkSvMon`**) from the WIBU start menu.
2. Expand the WIBU-KEY Server List. A list of connected clients appears.
3. Select the faulty client machine and click Cancel User. The client session will be deleted, thereby enabling a new connection.

Figure 1-24. Cancelling a Session



You can also reset a WIBU-KEY session by restarting the Licensing Server's WIBU-KEY application and re-initializing the encryption system. If WIBU-KEY is running as a normal application, use the standard Windows GUI to close the application and restart. If WIBU-KEY is running as a Windows service, use the standard Windows Administrative Tool "Services" to stop and then re-start the WIBU-KEY services. Alternatively, within a command prompt type "**`net stop WkSvW32`**" followed by "**`net start WkSvW32`**".

Verifying the Environment

During installation, Terazona v1.4.1 will attempt to set two environment variables, one for the location of your Java JDK and one for the location of your target installation directory.

Verifying the Windows Environment Variables

To ensure that these variable have been set:

1. Navigate to the environment variables screen:
   Control Panel > System Properties > Advanced > Environment Variables.
2. Locate **`%JAVA_HOME%`** (for example, **`C:\jdk1.4.2`**) and **`%ZONA_HOME%`** (for example, **`C:\zona\server1.4.1`**) in the listing of system variables.
3. If either or both of these two variables do not appear, click New and add the name and value (that is, the pathname) for the variable.
4. To use MySQL, ensure that the MySQL bin directory is added to the Windows path System Environment Variable. For a default Windows MySQL install, the MySQL bin directory is **`C:\mysql\bin`**.

Within Windows environments, the variables must conform to a DOS-style 8.3 naming convention and contain no spaces. Additionally, the directory path must contain no spaces. White space in the filename or directory path may cause some of the Terazona support utilities to behave unpredictably. The tilde (**~**) abbreviation operator is acceptable to build conforming pathnames, as are dashes (**-**) and underscores (_).

Creating the Game Database

You must create a SQL Server database schema on the machine where you installed the Game Database Server.

1 To do this, navigate to the **%ZONA_HOME%\sql\mssql** (MS SQL) or **%ZONA_HOME%\sql\mysql** (Windows MySQL) directory and run:

```
CreateZonaSchema host sa pass
```

where:

host - the hostname with the target database

sa - the database user with authority to create Tables in this database

pass - the password for the authorized user

This creates a new database called **ZonaDB** and grants full administrative access to two new users – **ZonaDeveloper** and **ZonaGame**. No user accounts are created.

Creating the Audit Database

You must create a SQL Server database schema on the machine where you installed the Audit Database Server. Note that your Audit Database and Game Database schema and tables can exist on the same machine, but this will affect performance if you activate the Auditing functionality. Production machines should separate the Audit Database and the Game Database onto different database server machines.

1 To do this, navigate to the **%ZONA_HOME%\sql\mssql** (MS SQL) or **%ZONA_HOME%\sql\mysql** (Windows MySQL) directory and run:

```
CreateZonaAuditSchema host sa pass
```

where:

host - the hostname with the target database

sa - the database user with authority to create Tables in this database

pass - the password for the authorized user

This creates a new database called **ZonaDB** and grants full administrative access to the two new users created earlier by **CreateZonaSchema** – **ZonaDeveloper** and **ZonaGame**.

Initializing the Databases

After creating the databases, you must create some necessary user tables:

1 Execute one of the **setup.bat** sample initialization batch files. The samples are located here:

```
%ZONA_HOME%\samples\
```

Creating Additional Users

CreateDemoUser will produce an error unless Zona Modeler has been previously called at least once to initialize user tables. Calling the sample applications' **setup.bat** initialization batch files accomplishes this by invoking Zona Modeler.

Use **%ZONA_HOME%\bin\createDemoUser.bat** utility to create additional users. The syntax is:

```
CreateDemoUser <user_name> <password> [<user_role> [<count>]]
```

There are two required parameters:

user_name – The username to create.
password – The password for the created user.

There are two optional parameters:

user_role – The privilege level for the created user. These values are one of:

Table 1-1. User Privilege Roles

User Role	Value
UNKNOWN	0
SYSTEMADMIN	1
GAMEMASTER	4
NPC	8
PLAYER	**16 (*default*)**

count – The number of users to create. Usernames will be auto-generated incrementally based on the supplied **user_name**. To use **count**, you must supply a valid **user_role** value.

Building the Samples

To build the Terazona demonstration application samples, you must execute **`setup.bat`** batch filess to perform required configuration:

1. To initialize the demo applications, execute the **`setup.bat`** batch file within each sample project directory. This invokes Zona Modeler to create sample-specific metadata and schema and any required user accounts (by calling **`createDemoUser`**). Additionally, Zona Modeler generates sample-specific C++ and Java source code and compiled Java class files that define the network game objects required by each sample. Modify the setup parameters to suit your development requirements.
2. Next you must build the sample applications. Within each sample project directory, execute the **`MakeRelease.bat`** batch file. For debugging, you can use the **`MakeDebug.bat`** batch file.

If you selected the Developer Components install option, then all the samples are deployed pre-compiled. Use Step 3 to re-compile the samples if and when you make changes to their source code or schema.

Running the Terazona Servers

There are several possible configurations for Terazona. Among these are:

Table 1-2. Terazona Configurations

Configuration	Description
Single Machine	All servers execute on a single machine. This is common for development environments.
Multiple Machines	Some or all of the Terazona servers can be distributed across multiple machines. This is common for production environments.

Running Terazona on a Single Windows Machine

This is a common development configuration. To run all your servers on one machine, ensure that the **`%ZonaHome%\config\zona.xml`** file includes the following line:

```
java.naming.provider.url, equal http://localhost:2000
```

On Windows machines, run the following batch files, in order, from:
Start > Programs > Terazona Server 1.4.1:

1. **Servers > Authentication server**
2. **Servers > Messaging server**
3. **Servers > ZAC server**
4. **Servers > Dispatcher server**
5. **Servers > Sphere server**
6. **Servers > Game State Server**
7. **Servers > Game Guild Server**
8. **Servers > Audit Server**
9. **Development > Zona Modeler Interface**

Running Terazona on Several Machines

This is a common production configuration. To run multiple Game State Servers on multiple hosts, all coordinated by one Sphere Server and Messaging Server on a separate machine, you initialize one machine to act as the "administrative" machine (with no Game State Server process) while on the others you initialize only the Game State Server process.

Initializing the Administrative Servers

On Windows, run the following batch files, in order, from **Start > Programs > Terazona Server 1.4.1**:

1. **Servers > Authentication Server**
2. **Servers > Messaging Server**
3. **Servers > ZAC Server**
4. **Servers > Dispatcher Server**
5. **Servers > Sphere Server**
6. **Servers > Game Guild Server**
7. **Servers > Audit Server**
8. **Development > Zona Modeler Interface**

Initializing the Game State Servers

To initialize additional Windows servers to function as Game State Servers:

1. Run the Installer on each destination machine.
   Select the GSS install option only and enter the address of the Messaging Server.
2. Run the batch file **Start > Programs > Terazona Server 1.4.1 > Game State Server**

Repeat the above steps to set up multiple Game State Servers on multiple hosts.

Configuring Firewalled Servers

When Terazona is implemented within a firewalled or Network Address Translation (NAT) environment, you must modify the Dispatcher's **`zona.xml`** file manually. This is located under the config directory on Dispatcher Server. This file enables you to translate which internal IP address(es) map to external IP address(es) for GSS servers.

The entry has to be made in the following XML tag in **`zona.xml`**:

```
<Dispatchers>
    <DispatchersCommon>
      <Ipmap internal="192.168.1.1" external="x.x.x.x"/>
    </DispatchersCommon>
</Dispatchers>
```

- **`internal`** corresponds to the internal IP address of the message broker host
- **`external`** corresponds to the external IP address of the message broker host.

The message broker host is the host of the GSS to which the clients will connect after being redirected.

Running the Samples

After you have set up the servers and installed the client applications, the sample applications are now ready to run. You can use an example client, such as TrackerClient or the ZonaBattle Client. However, before you can run a specific client, you must first configure this file:

```
%ZONA_HOME%\config\zona.xml
```

To develop your own applications or control which Plugin gets loaded by the Server, you should change this value:

```
pluginDynamicLibName="ZonaBattle_ServerPlugIn"
```

Change **`ZonaBattle_ServerPlugIn,`** to the name of your compiled DLL. This will change the name of the main Server plugin.

For further details, see the "Development Environment" chapter of the *Terazona Developer's Guide*.

Identifying the Samples

This table describes the installed sample applications in **%ZONA_HOME%\samples**:

Table 1-3. Sample Applications

Sample Application	Description
%ZONA_HOME%\samples\ Authentication\AuthSamplePlugin	Reference example of how to integrate Terazona with an external Authentication/Billing system.
%ZONA_HOME%\samples\DogClock	An example that demonstrates how to simulate and vary the passage of time across a Terazona cluster and hence within a virtual world with different time zones.
%ZONA_HOME%\samples\GameGuild	An example that demonstrates Terazona's group-based entity state update technology known as Game Guilds.
%ZONA_HOME%\samples\MineHunt	An example that demonstrates the interaction between separate Clients and the NPC Server. This is a good place to learn about how to transfer objects between Clients and Servers.
%ZONA_HOME%\samples\tiletest	A simple game that demonstrates Character initialization and Client-Server interaction.
%ZONA_HOME%\samples\trackerclient	A simple example that demonstrates Character initialization and Client operations.
%ZONA_HOME%\samples\zchatter	An example that demonstrates Guild Chat operations. Can be used as a basis for creating Player Chat applications.
%ZONA_HOME%\samples\zonabattle	A very complex demo game using Terazona C++ API. First-person shooter (FPS), rapid fire, many entity updates. NPC Server.

The clients can also be started from the **Start** menu. Note that the above examples refer to the path to client applications if the Client Development environment was installed.

Creating Terazona Applications

The Client API (CAPI) is a C/C++ interface enabling clients to exchange messages with other clients through a hierarchal grid of Terazona game servers known as a Terazona Cluster.

The bulk of the API resides in the class ZonaServices. By instantiating a ZonaServices object, developer-made clients can logon, enter the game, and send and subscribe to messages. Principally, the API delivers information from ZonaServices to the clients using callback objects, enabling developers to define the callbacks and the actions to take when receiving the data.

A simple example of a client logging on, subscribing and receiving messages is illustrated in the C++ TrackerClient example, found here:

```
%ZONA_HOME%\samples\TrackerClient\
```

The TrackerClient is a small, well-documented C++ example installed with the source code.

A more complex example, TileTest, demonstrates client-server interaction and validation within a simple game, and can be found here:

```
%ZONA_HOME%\samples\TileTest\
```

ZChatter is a demonstration of Terazona's player Chat system, and can be found here:

```
%ZONA_HOME%\samples\ZChatter\
```

The most sophisticated bundled example is ZonaBattle, a reference implementation of a fast-action motorized first-person shooter game, and is found here:

```
%ZONA_HOME%\samples\ZonaBattle\
```

Note that the installer places the source code examples on your machine only when the developer install option is selected during installation. There is no source provided with the default install for ZonaBattle. Please contact support@zona.net for details of how to obtain ZonaBattle source code.

You will get the files for the above examples if you choose the Terazona Client development environment option during installation.

Using the Online Documentation

Further documentation of the CAPI exists in HTML format here:

```
%ZONA_HOME%\doc\api\client\index.html
```

The HTML documentation gives descriptions of the classes and methods making up the C API.

This documentation is also compiled into Microsoft CHM HTML Help files here:

```
%ZONA_HOME%\doc\Help\Client.chm
```

Uninstalling Terazona

We recommend that you completely uninstall all older Terazona versions before installing the most up-to-date version.

Uninstalling Terazona on Windows

1 Run the following command:

Start > Programs > Terazona Server 1.4.1 > Uninstall Terazona Server 1.4.1

The designated Terazona version will be uninstalled.

Getting Additional Information

For the most up-to-date information on Terazona v1.4.1, check the Release Notes documentation. You can also email Terazona support at support@zona.net or check our website at http://www.zona.net/support/ for further information.

Part

II

Maintaining Terazona

This part of the Terazona Installation and Configuration Guide demonstrates how to maintain, audit, and update Terazona.

Chapter

2

Modifying the Terazona License Terms

This document contains information about maintaining and updating your Terazona license terms. If you received the Community Edition of Terazona, then you will probably wish to upgrade your licence terms to operate within a production environment.

Modifying Your Terazona License Terms

The WIBU-BOX stores a customer-specific encryption key that controls the licensing terms for the following Terazona system properties:

- Maximum number of concurrent GSSs
- Maximum number of concurrent Client connections
- Expiration date

Altering the WIBU Key (and thus your license terms) is a four-stage process:

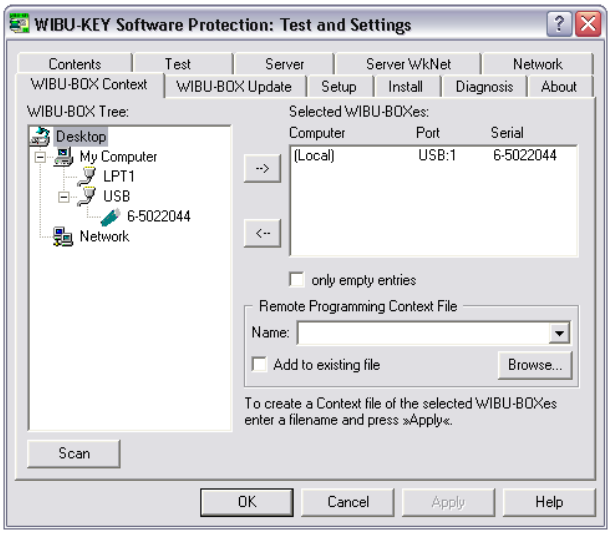
1. Contact us to discuss your new license terms.
2. Extract an RTC (remote context) file from the WIBU-BOX.
3. Send this RTC file to us.
4. Based on your license requirements, our engineers will use this RTC context file to create an RTU (remote programming update) file that will amend your license terms. We will then send you this response file.
5. You use this RTU response file to update your WIBU-BOX configuration.

Extracting an RTC License Context File from the WIBU-BOX

1. Select **Start > Settings > Control Panel**.
2. Double-click the WIBU application icon, and then click the WIBU-BOX Context tab.
3. If there is more than one WIBU-BOX deployed within your network, highlight your intended choice within the Selected WIBU-BOXes panel.

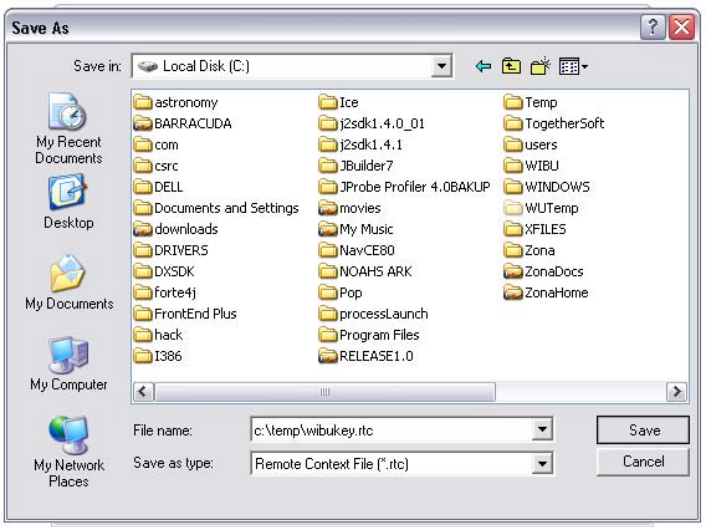
4 The context of the WIBU-BOX displays in the right-hand panel:

Figure 2-1. WIBU-Box Context



5 Click the **Browse...** button. A standard file location selector dialog displays:

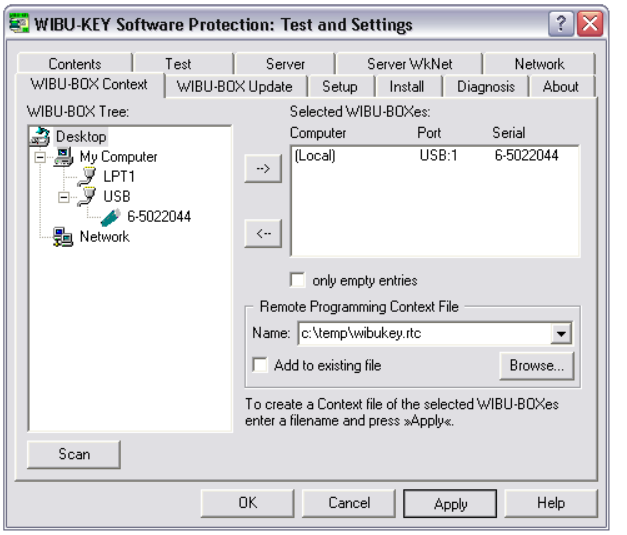
Figure 2-2. WIBU-BOX RTC Save Location Dialog



6 Navigate to your desired save location for the RTC context file

7 Name your RTC context file and click the **Save** button. The WIBU-KEY Context redisplays with your selected save location specified:

Figure 2-3. WIBU-BOX RTC Save Location Specified



8 Click the Apply button to extract the RTC context file. A dialog box appears to indicate success:

Figure 2-4. WIBU-BOX RTC Extraction Notification Dialog

9 Click the OK button. You have successfully extracted an RTC context file from the WIBU-BOX.

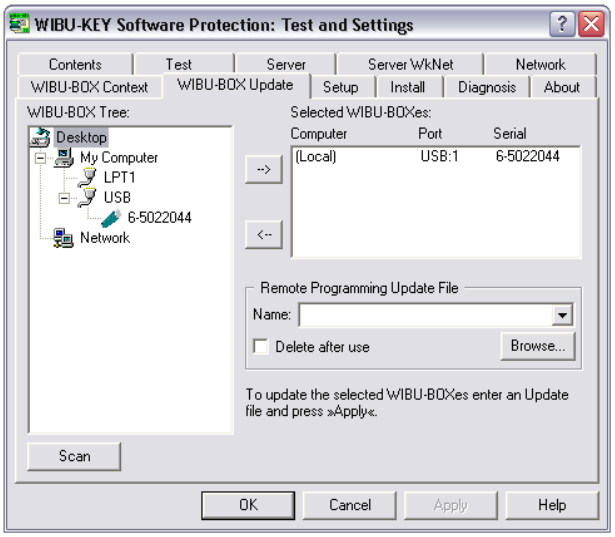
Applying an RTU License Update File to the WIBU-BOX

1 Select **Start > Settings > Control Panel**.

2 Double-click the WIBU application icon, and then click the WIBU-BOX Update tab.

3 If there is more than one WIBU-BOX deployed within your network, highlight your intended destination within the Selected WIBU-BOXes panel.:

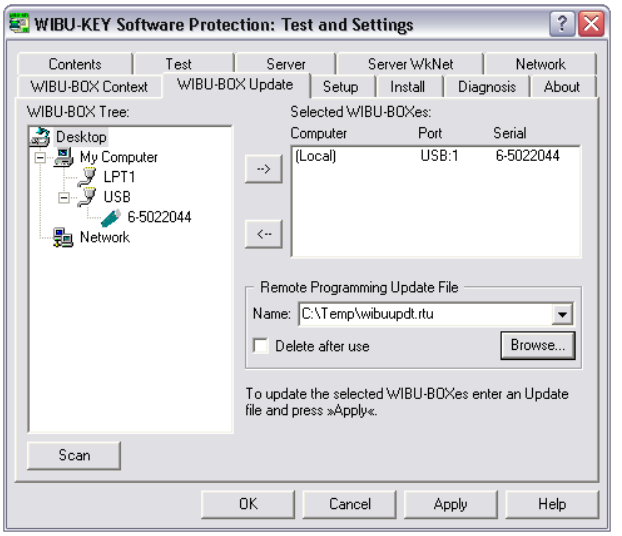
Figure 2-5. WIBU-BOX Update



4 Click the **Browse...** button to select the RTU file that you received from us.

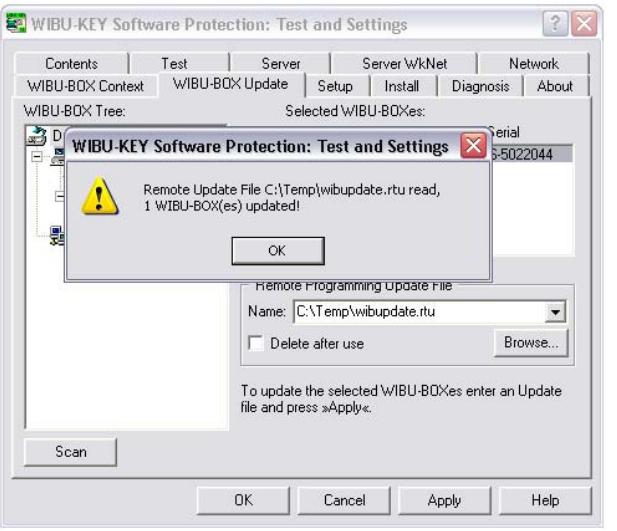
5 Within the file selector dialog that appears, highlight the desired RTU file and click the **Open** button. The WIBU-BOX Update panel redisplays with your desired RTU file selected:

Figure 2-6. WIBU-BOX Update with Desired RTU File Selected



6 To write the contents of the RTU file to the WIBU-BOX, click the Apply button. A dialog box appears to indicate success:

Figure 2-7. WIBU-BOX RTC Extraction Notification Dialog



7 Click the OK button. You have successfully applied an RTU file to your WIBU-BOX. This means that your Terazona license terms have now been updated.

Chapter

3

Configuring Terazona Authentication and Security

Terazona features two main approaches to protecting system integrity: authentication and security. This document contains information about configuring the Authentication Server and the **`ZonaServers.ini`** security file.

Understanding the Authentication Server

The Authentication Server manages connections between customers' commercial databases (loaded with sensitive, financial and billing information) and Terazona's game servers, specifically its Messaging Servers. The Authentication Server forms a middleware layer with a standardized API that customer-specific applications or web services can access without revealing their system internals or sensisitve cusomer billing data to any Terazona servers.

The Game Database (named **ZonaDB** in a default install) and the Billing Database (BillDB) are by design physically and logically seperate for security and privacy concerns. You can wrap or interface the Authentication Server with your own custom sign-on applications. During authentication, you authenticate a player's session requests against your own BillDB (which can be a physical database, a web service, or some LDAP object). Following successful authentication, you hand off that player session to the Terazona Messaging Servers to mediate the player's in-game experience.

Game roles are controlled within the **BillDB.USERDIRECTORY**. To manually upgrade user roles, you must edit this table. The default player permission value is **0**.

The design approach of the Authentication Server Plugin is this:

1. Obtain user's name (**USERNAME**) and password credentials.
2. Query BillDB whether to log in this user.
3. If allowed, then query BillDB again to obtain the user's role.
4. Log this user into Terazona with the specified role, using the unique ASCII-defined **USERNAME** as a common key for both tables.
5. If the user account already exists within **ZonaDB.USERTABLE**, use that. Otherwise, create a new entry for this **USERNAME**.

Roles generally specify to which messaging channels a user can subscribe. Therefore, the Authentication Server talks directly to the Messaging Servers.

Implementing the Authentication Server

The Authentication Server (AuthServer) is located here:

```
%ZONA_HOME%\bin\AuthServer.exe
```

Run this executable file on the machine designated as the Authentication Server. Once running, the AuthServer process checks within the **%ZONA_HOME%\bin** directory for a DLL that implements the Authentication logic. The default Authentication DLL is:

```
%ZONA_HOME%\bin\AuthSamplePlugin.dll
```

This sample Authentication Plugin uses a table in the Game Database called **`ZonaDB.userdirectory`** to store its account data. The sample code contains ODBC-specific database access code for this table. This should be replaced by your own authentication system. This Plugin provides an interface where you can implement an API to communicate with billing servers.

The Authentication Server will automatically maintain the **`BillDB.USERDIRECTORY`**. If the Plugin authenticates a user account that is not found in the Game Database, then the Authentication Server will create a suitable user account in the Game Database.

Configuring the Authentication Server

When you implement your own Authentication DLL, you need to update two configuration files on the Authentication machine with the new DLL's name and location. The two files are:

```
%ZONA_HOME%\bin\auth.ini
  UserAuthPlugin=.\AuthSamplePlugin.dll
...
%ZONA_HOME%\config\zona.xml
  <LibName authDynamicLibName="AuthSamplePlugin.dll" …
```

Update the DLL name reference to reflect your customizations. For example, you can modify these files thus:

```
%ZONA_HOME%\bin\auth.ini
  UserAuthPlugin=.\MyAuthPlugin.dll
...
%ZONA_HOME%\config\zona.xml
  <LibName authDynamicLibName="MyAuthPlugin.dll" …
```

Using the Authentication Server API

The Authentication Server provides two new API functions that developers use to implement Terazona authentication. These functions are defined here:

```
%ZONA_HOME%\include\AuthZonaPlugin.h
```

The functions available for authentication are:

```
int initialize()
```

Used by the Authentication Server to inform the external authentication system that it is beginning operation.

```
int userLoginAuth(char* username, char* password)
```

Enables the developer to query an external authentication system. This function returns true if the user is authenticated. The Authentication Server will then allow the player's Client to log into the GSS.

```
void userLogout(char* username)
```

Called by the Authentication Server when a user logs out. Client code can thus clean up a user's account profile in the Authentication system when this call is made.

```
void finalize()
```

Used by the Authentication Server to inform the external authentication system that it is ceasing operation, pending shutdown.

The Authentication Server validates all attempted calls against a list of error codes defined in this file:

```
$ZONA_HOME\include\AuthZonaPlugin.h
```

To use these functions, link to this library:

```
%ZONA_HOME%\lib\AuthZonaPlugin.lib
```

Examining the Authentication Plugin Demo

The source code for this installed AuthSamplePlugin demo is found here:

```
%ZONA_HOME%\samples\Authentication\AuthSamplePlugin
```

This example uses the **`BillDB.USERDIRECTORY`** for authentication.

Understanding Terazona Security

Terazona provides a network security model in addition to any firewall policies that you operate. The Terazona model is based on the idea of Authenticated Clients and Trusted Clients/Servers.

Authenticated Clients are those external client sessions that the Authentication Server has approved and subscribed to specific messaging channels within the Terazona cluster. These are usually player sessions.

Trusted Clients are those machines within a Terazona LAN cluster that effectively bypass the Authentication Server and are automatically subscribed to the Terazona Message Servers. The NPC Server is a Trusted Client.

Implementing Terazona Security

The Authentication Server distinguishes between requests from external, non-trusted clients and internal, Trusted Clients using an IP mapping file. This file is located on the Authentication Server, here:

```
%ZONA_HOME%/config/ZonaServers.ini
```

This file is used by the Authentication Server to distinguish between internal, LAN machines and external non-LAN machines for the purpose of Authentication and message routing.

Configuring Terazona Security

You use the **`ZonaServers.ini`** file to specify the LAN IP addresses (not hostnames) of all those machines that comprise your Terazona cluster, including any machines that run as NPC Servers.

This file's entries looks like this (IPs are for example purposes only):

```
#This is a list of Zona Server IPs
#All your Zona Servers must have an entry
#in this file
10.10.1.1
172.16.1.1
192.168.1.1
```

During installation, you are prompted to supply the IP addresses of your Terazona cluster machines. To add more machines to the cluster after the initial install, you can manually edit the **`ZonaServers.ini`** file to add these IP addresses.

Because the Terazona system explicitly trusts all machines with IPs that correspond to entries in the **`ZonaServers.ini`** file, exercise caution when configuring your firewall policies. You should filter IP addresses, and guard against hack attacks such as IP spoofing.

Using Localhost IP Addresses For Single Development Machines

Many development machines run all Terazona servers on one machine. With this configuration, you must add the real IP address that is bound to that machine's NIC (and ***not*** the classic pseudo-IP address for a localhost of **`127.0.0.1`**) to **`ZonaServers.ini`** during installation. This is because during the authentication process, the Authentication Server receives the authentication request as originating from that IP address.

Chapter

4

Auditing Terazona

This document contains information about auditing Terazona. This involves recording in-game entity state changes and game activities into the Audit Database in a normalized format that can be queried to create reports and analyze game trends.

Auditing Terazona with the Auditing Server

The Auditing Server (AS) provides a comprehensive auditing facility for Terazona clusters. The AS rationale is to provide an offline, digested report of important game activities. We recommend deploying the Game Database and the Auditing Database on seperate database servers to optimize performance.

Auditing enables Terazona administrators and Game Masters (GMs) to watch trends in player actions and behaviors, to analyze the record of all Chat activities, and to provide an interface for data mining applications that can help to extrapolate future game trends or developing imbalances.

The Developer Guide describes how to select Entities for Auditing, and how to specify which Properties to Audit. In addition to specifying which Entity Properties to Audit from within the ZMUI, to enable Auditing, you must:
Manually edit the **`zona.xml`**:

```
<Service chat="false" gamestate="false" guild="false"/>
```

Change these values to **`true`**:

```
<Service chat="true" gamestate="true" guild="true"/>
```

Run the Auditing Server

Understanding the Auditing Server Rationale

The AS records Game State Data (GSD) about Players, Characters, Entities, and important transactions and exchanges of properties and state data. The AS uses the concept of Selector Ids to specify the type of GSD transaction that has occurred.

The goal of the AS is to record all game activities that have a permanent impact. The AS is not intended to be used to record short duration transient or repeated "trivial" activities.

For example, the AS records what spells have been cast, and their effects on game Characters. The AS does not store simple motion deltas; instead it records a Character's location at the moment a significant transaction or event occurs that affects their GSD.

Using the Auditing Server Selectors

The AS uses the concept of Selectors and Properties as flags and tokens about game activities that are stored in the ZonaAuditDB for future reconstruction of game activities.

These are the current supported GSD Selector Ids:

- **PLAYER_LOGIN**
- **PLAYER_LOGOFF**
- **CHARACTER_CREATION**
- **CHARACTER_SELECTION**
- **CHARACTER_UPDATE**
- **CHARACTER_DELETION**
- **CHILD_ENTITY_CREATION**
- **CHILD_ENTITY_DELETION**
- **CHILD_ENTITY_TRANSFER**
- **ENTITY_DATABASE_UPDATE**

The AS records all Sphere Chat messages, storing them directly in the ZonaAuditDB. The messages are stored with their Recipient data and a Flag that indicates their status:

- **UNDOWNLOADED**
- **UNREAD**
- **DELETED**
- **REPLIED**

The AS also records all Guild Activities. These are the current supported Guild Activity Selector Ids:

- **CREATION**
- **DELETION**
- **MEMBER_ADDITION**
- **NEW_MODERATOR**
- **MEMBER_REMOVAL**
- **INVITATION_SENT**

The Guild Properties are also stored in the ZonaAuditDB. These are the available Guild Properties:

- **PERSISTENT_MEMBERSHIP**
- **EMAIL_FOLLOW_UP**
- **PERSISTENT_MSG**
- **DESTROY_WITH_OWNER**
- **DONT_SEND_MEMBERSHIP_CHANGES**
- **MEMBERS_CAN_POST_MSG**

- **MEMBERS_CAN_DELETE_ANY_MSG**
- **MEMBERS_CAN_DELETE_OWN_MSG**

The Auditing Server currently does not support the auditing of Game Guilds. Audit support for Game Guilds will be added in a future release.

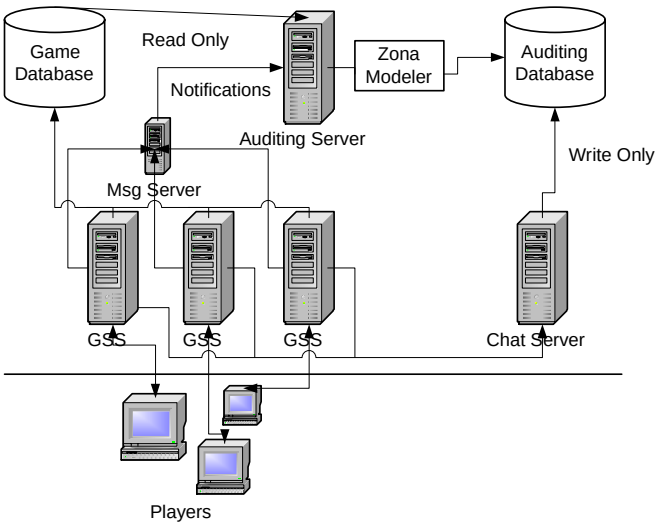
Scoping the Auditing Server

The AS is an optional install within the Terazona server cluster. You can choose whether or not to install the AS within the initial install sequence. The AS uses its own dedicated Messaging Server to minimize its impact on the other Terazona servers.

The AS uses a dual audit channel approach: the GSD is auditing using one event queue while the Chat Data is audited directly into the Audit Database (AuditDB) using a separate channel. This decoupling helps ensure that the access to the Game Database can be optimized independently from the access to the Game Guild Server.

The AS does not "listen" to the Dispatcher, but instead listens only to output from the GSSs. This reduces server inter-dependencies. In terms of server communications, the AS looks like this:

Figure 4-1. Auditing Server Communications Schematic



The AS requires Zona Modeler to provide database schema serialization and deserialization.

Managing the Auditing Server

The install process for the AS deploys the ZonaDaemon on the selected machine. The Zona Admin Controller (ZAC) therefore recognizes the AS and displays standard management diagnostics.

Understanding the GSD Audit Sequence

The Audit Sequence for GSD uses a notification queue and functions as follows:

1. GSSs send timestamped data to the Game Database for storage. Simultaneously, a timestamped notification that indexes this data is sent to the AS. The notification contains the timestamp, the Entity Id, and a Selector Id.
2. At regular periodic intervals the AS polls the Game Database:
   a. The AS checks the timestamps of the Game Database Entity Records.
   b. For notifications where the Selector ID is **`ENTITY_DATABASE_UPDATE`**, the AS checks the timestamps. If the timestamps match (or the Game Database is "older") then the AS discards all earlier notifications for that Entity Id.
   c. If the notification timestamp for a given Entity Id is older than the timestamp in the Game Database then the AS postpones any action on notifications for this Entity Id until it checks the timestamps during the next audit cycle.
3. At the end of an Audit Cycle, the AS commits certain transactions directly to the ZonaAuditDB that do not involve modifying Entity Records (that is, binary game data). These have the following Selector Ids:
   - **`PLAYER_LOGIN`**
   - **`CHARACTER_DELETION`**
   - **`CHARACTER SELECTION`**
   - **`CHARACTER_DELETION`**
   - **`PLAYER_LOGOFF`**
4. For transactions that modify Entity Records, the AS copies the entire binary record for that Entity from the Game Database and hands it off to the Zona Modeler (ZM). The ZM then normalizes this binary data (using user-defined, game-specific schema) and inserts the normalized data into the GSS Data table of the Audit Database. The transactions that trigger this ZM normalization process are:
   - **`CHARACTER_CREATION`**
   - **`CHARACTER_UPDATE`**
   - **`CHILD_ENTITY_TRANSFER`**
   - **`ENTITY_DATABASE_UPDATE`**

5 After several unsuccessful attempts (the current default is three times) to extract data from the Game Database for a specific Entity Id, the AS will give up and discard the notification. You can manually change this value by editing the zona.xml file authentication parameters, as described in *Configuring the Auditing Server on page 66*.

This algorithm has the effect of discarding most continuous, trivial changes (such as motion deltas) in favor of recording only a snapshot location of an Entity at the moment of the Audit cycle. If there are many Entity updates within the Audit interval, only the most recent Entity Object update gets Audited.

The notification queue for game state data looks like this:

Table 4-1. Game State Data Notification Queue

TimeStamp	Selector(Event)	EID	UserID	ParentID	Event Type
10:22:01	PLAYER_LOGIN		98		
10:22:02	CHARACTER_DELETION	115			
10:22:03	CHARACTER_CREATION	115			
10:22:03	CHARACTER_SELECTION	115			
10:22:03	CHARACTER_UPDATE	115			
10:22:10	CHILD_ENTITY_CREATION	120		115	
10:22:20	CHILD_ENTITY_TRANSFER	120		116(new)	
10:22:25	CHILD_ENTITY_DELETION	121			
10:32:55	ENTITY_DATABASE_UPDATE	115			
10:32:55	ENTITY_DATABASE_UPDATE	115			
10:32:55	ENTITY_DATABASE_UPDATE	115			
10:33:00	PLAYER_LOGOFF		98		

Understanding the Chat Audit Sequence

The Audit Sequence for Chat Data is simpler than that for GSD Auditing. All Chat events (for both Sphere and Guild Chat) are recorded directly into the ZonaAuditDB, bypassing the Auditing Server notification queue and the ZM. The Game Guild Server can be configured to send the data to the ZonaAuditDB either pre- or post- content filtering.

Configuring the Auditing Server

The Auditing Server is configured through the following typical entry in the **zona.xml** file:

```
<AuditServers>
    <AuditServersCommon>
        <Service chat="true" guild="true" gamestate="true"/>
                        <Chat filtered="false"/>
                        <Process interval="5000"/>
    </AuditServersCommon>
    <AuditServer id="ZonaAuditServer">
        <MBs>
            <MB id="localhost" priority="primary"/>
        </MBs>
    </AuditServer>
</AuditServers>
```

The entries within the **AuditServersCommon** block select which data are recorded to the ZonaAuditDB.

- **gamestate="true|false"** indicates whether Game State Data should be audited.
- **chat="true|false"** indicates whether Sphere Chat data should be audited.
- **guild="true|false"** indicates whether Guild Activities data should be audited.
- **Chat filtered="true|false"** indicates whether to audit Chat and Guild data pre- or post-content filtering.
- **Process interval="5000"** indicates the length of the Audit cycle, in milliseconds (ms). For shorter intervals (that is, smaller integer values) the Auditing Server will perform less "compactification" of rapid Entity Object updates in its notification queue. This reduced compactification will cause more Entity Object data to be read from the Game Database; the quantity of transactions flagged for storage within the ZonaAuditDB will increase; and finally there will be an increase in CPU normalization load. Setting the Audit Cycle to a low interval (that is, a finer granularity) will thus impact the performance of the Game Database and the ZonaAuditDB.

The **<AuditServer id="ZonaAuditServer">...</AuditServer>** section does not require end-user modification at this time.

Analyzing the Auditing Server Data

The ZonaAuditDB has four tables:

1. Game Session Data - **`ZonaEntityAudit`**
2. GSS Data - ***`UserDefinedEntityName`*****`Audit`**
3. Chat Messages - **`ChatAudit`**
4. Guild Activities - **`GuildAudit`**

`UserDefinedEntityName` is the Entity name as defined within Zona Modeler. This name is used by the Auditing Server with the **`Audit`** suffix appended to create unique table names.

The **`ZonaAuditDB.ZonaEntityAudit`** table name idoes not change.

Using a games-specific, developer-defined XML Schema translation file, Zona Modeler normalizes the binary Entity Object data from the Game Database into discrete ZonaAuditDB records for storage within the GSS Data Table. The ZonaAuditDB is automatically configured during the installation of the Auditing Server. Zona Modeler creates the GSS data table when it is installed.

Player activities that do not result in an Entity Object modification (and therefore do not require the normalization of binary data into the AuditDB) are stored within the **`ZonaAuditDB.ZonaEntityAudit`** table.

Auditing GSS Data

The **`ZonaAuditDB.ZonaEntityAudit`** table looks like this:

Table 4-2. ZonaAuditDB.ZonaEntityAudit Table - Game Session Data

Time Stamp	UID	EID	Selector(Event)	PID	ET
10:22:57	98		`PLAYER_LOGIN`		
10:23:33	98	115	`CHARACTER_DELETION`		
10:33:59	98	115	`PLAYER_LOGOFF`		

After the GSD Audit scheduling algorithm has discarded redundant Entity Object updates, the data sent to Zona Modeler for incorporation within each **`ZonaAuditDB.`*`UserDefinedEntityName`*`Audit`** table look like this:

Table 4-3. ZonaAuditDB.*UserDefinedEntityName*Audit - GSS Data Table Input

Time Stamp	UID	EID	Selector(Event)	PID	ET	*User Defined*	*User Defined*
10:30:15	98	115	`CHARACTER_CREATION`		Default	Binary Data	Binary Data
10:30:16	98	115	`CHARACTER_SELECTION`		Default		
10:30:16	98	115	`CHARACTER_UPDATE`		Default	Binary Data	Binary Data
10:31:43		90	`CHILD_ENTITY_CREATION`	110	Default		
10:32:55		90	`CHILD_ENTITY_TRANSFER`	115	Default	Binary Data	Binary Data
10:32:58		114	`CHILD_ENTITY_DELETION`		Default		
10:32:58		115	`ENTITY_DATABASE_UPDATE`		Hidden	Binary Data	Binary Data

When calculating PID, please note that for **`CHILD_ENTITY_CREATION`**, EID is a new child entity id and PID is its parent id; while for **`CHILD_ENTITY_TRANSFER`**, EID is a acquired child entity id and PID is a new parent id.

Zona Modeler will convert the Binary Data "blobs" into normalized data for storage within each **`ZonaAuditDB.`*`UserDefinedEntityName`*`Audit`** table using the user-defined normalization schema specified within the Zona model file definition for each Entity.

Auditing Chat Messages

The Sphere Chat Message data is sent directly to the Chat Messages table in the **`ZonaAuditDB.ChatAudit`**. The table data looks like this:

Table 4-4. ZonaAuditDB.ChatAudit - Chat Data Table

Time Stamp	Sender	Recipient	Guild	Sphere Recipients	Subject	Body	Flag
10:33:15AM	130	150				Hello	
10:34:22AM	145			101, 102, 103, 104,		How	
10:35:50AM	121		1001			Get a	
10:35:51AM	122	-1				System	

A **`Recipient`** value of **`-1`** indicates that this was a system-wide chat message, received by all Characters.

Currently, the maximum quantity of Sphere Chat recipients handled by the Auditing Server for is 500. Recipient IDs above the 500 limit will be silently ignored by the Auditing Server.

Auditing Guild Activities

The Guild Activities data is sent directly to the Guild Activities table in the **`ZonaAuditDB.GuildAudit`**. The table data looks like this:

Table 4-5. ZonaAuditDB.GuildAudit - Guild Activities Table

Time Stamp	Guild	Selector	Requester	Entity	Properties
10:33:15AM	10	**`CREATION`**	330		**`MEMBERS_CAN_POST_MSG`**
10:34:20AM	10	**`DELETION`**	330		**`MEMBERS_CAN_POST_MSG`**
10:35:32AM	10	**`MEMBER_REMOVAL`**	330	322	**`MEMBERS_CAN_POST_MSG`**
10:36:01AM	10	**`INVITATION_SENT`**	330	333	**`MEMBERS_CAN_POST_MSG`**

Chapter

5

Logging and Monitoring Terazona

Terazona server cluster operations can be monitored and logged during run-time execution. This is distinct from Auditing - which refers to the recording of game world entity state changes.

Logging with Terazona

Terazona enables you to log application messages, warnings, or critical errors with ajustable levels of precision.

Configuring Logging Using the zona.xml File

Terazona's logging services can be configured within the **`zona.xml`** file:

```
<Log debugLevel="5" fileName="ZonaServer.log"
                              isConsole="true" isFile="true"
                          isNetwork="false" size="123456"/>
```

The entries within the **`Log`** block control certain Logging parameters:

- **`debugLevel="1|2|3|4|5|6|7"`** indicates the level of logging detail, with 1 indicating the maximum level and 7 the minimum level of logging detail.
  - Levels **`1,2,3`** generate the maximum quantity of log messages. Setting these levels can affect Terazona's performance and they are recommended for development, not production, installations.
  - Level **`4`** generates a large quantity of log messages and is not recommended for production installations.
  - Level **`5`** generates application information, warning, and error log messages and is the recommended default setting. This log level is known as **`INFO`**.
  - Level **`6`** generates application warning and error log messages. This log level is known as **`WARNING`**.
  - Level **`7`** generates only critical application error log messages. This log level is known as **`ERROR`**.
- **`filename="UserDefinedText"`** specifies the name of the file used to store the log messages. When the maximum **`size`** allowed for the file is reached, Terazona archives the existing log file by renaming it with a prefix and creates a new log file.
- **`isConsole="true|false"`** indicates whether to display or hide the log messages within the application console window.
- **`isFile="true|false"`** indicates whether the log messages should be written to the **`filename`** file, or discarded.
- **`isNetwork="true|false"`** indicates whether the log messages should be sent across the network to be viewed remotely by the Zona Admin UI. Setting this option to **`true`** with lower **`debugLevel`** settings such as **`1`**, **`2`**, or **`3`** can result in increased network traffic that may affect performance. We recommend setting this option to true only for higher **`debugLevel`** settings such as **`5`**, **`6`**, or **`7`**.

- **`size="IntegerValue"`** specifies the maximum logfile size, in KB. Excessively large KB settings are discouraged.

Configuring Logging Using the Console

You can also dynamically configure Terazona's logging services debugLevel settings during execution using the Terazona server console.

1. Activate the GSS Console.
2. At the command prompt, type:

   ```
   loglevel value
   ```

- **`logLevel="1|2|3|4|5|6|7"`** indicates the level of logging detail, with **`1`** indicating the maximum level and **`7`** the minimum level of logging detail.