ClearTrust SecureControl

Installation and Configuration Guide

Version 4.5

August 2000

ClearTrust SecureControl, Release 4.5

Securant Technologies, Inc., One Embarcadero Center, 5th Floor, San Francisco, CA 94111

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. I.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License.]

The reason behind this being stated in this direct manner is past experience in code simply being copied and the attribution removed from it and then being distributed as part of other packages. This implementation was a non-trivial and unpaid effort.

Year 2000 Compliance Statement

Securant is committed to producing the highest quality software products and services. We recognize the issues and potential problems associated with the storage and calculations of dates with two digit year fields (Year 2000 problem). All dates used within our products utilize a standard four digit year or system specific representation.

When installed and implemented in accordance with our current written specifications, ClearTrust meets or exceeds the following rigorous requirements:

(i) records, stores, processes, calculates, transmits, displays, and presents calendar dates on or after (and, if applicable, spans of time including) January 1, 2000.

(ii) the use of dates before, on or after January 1, 2000 will not adversely affect performance with respect to date-dependent data, computations, output, or otherfunctions.

(iii) will not abnormally end or provide invalid or incorrect results as a result of date-dependent data; and

(iv) accurately recognize, manage, accommodate, and manipulate date-dependent data including leap years.

.....

Chapter 1
Getting Started

Securant Technologies' ClearTrust SecureControl is a robust, enterprise-class security solution that is designed to be deployed in a wide range of environments. The system includes several components that interoperate securely over TCP/IP networks. The product has been built using state-of-the art programming methodologies and languages (Java, C/C++) and leverages infrastructure services, such as CORBA ORBs for inter-component communication. (For a more complete overview of the ClearTrust system architecture and its functionality, see Chapter 1, "ClearTrust SecureControl Overview" in the ClearTrust SecureControl *Policy Adminstration Guide*.)

The ClearTrust solution provides authentication, access control, authorization, single sign-on, delegated administration, policy assessment, intrusion detection, and auditing. This guide will help you install and configure the core components to deliver these security features to your organization.

Installation and Configuration Overview

Given that no two enterprise environments are alike, ClearTrust SecureControl is designed to be highly customizable and extensible. All components run on either Windows NT or on Solaris.

Minimum Installation Process

At a minimum, you will need to install the following core components for any ClearTrust SecureControl implementation:

- SecureControl Servers (Authorization Server, Entitlements Server, and Entitlements Database) and the SecureControl Entitlements database. These components provide the core services of the ClearTrust solution. See Chapter 2, “ClearTrust SecureControl Servers” for information about installing and configuring.
- SecureControl Web Server Plug-in on each Web server site that you want to protect. Securant provides Web Server Plug-ins for Apache, Microsoft IIS, and Netscape. See Chapter 3, “ClearTrust Web Server Plug-ins” for instructions. You can configure the Web Server Plug-ins to support Single Sign-on (SSO)—the more sites that you secure, the more your users will appreciate this feature—and you can also enhance the security of the SSO implementation by enabling encryption support. These topics are also covered in Chapter 3; see “Overview of Single Sign-on (SSO)” for more information.
- SecureControl Manager, the Java-based client application that security administrators will use to configure access privileges the users, groups, and realms; to distribute administrative privileges to user communities; to configure Basic Entitlements and Smart Rules; and to identify and configure the Web servers and applications that you wish to secure. This tool provides a single interface through which your enterprise-wide security policy is configured and managed. Chapter 7, Installing the ClearTrust Management Tools, provides the details. For information about using the tool for administering security, see the *Policy Administration Guide.*

You can also customize the Web forms, CGIs, and Servlets included with ClearTrust, and you can create your own. See Chapter 4, “Customizations” for information.

With the ClearTrust Servers, Management tool, and Web Server Plug-in installed, you can begin creating users and securing resources. The *Policy Administration Guide* provides complete information.

Optional Installation, Integration, Configuration

Depending upon the requirements of your enterprise environment, you may want to install one or more of the following ClearTrust components:

- LDAP Replication server to integrate user accounts from LDAP v3 compliant directory services into your Entitlements database. If you install LDAP Replication Server, you’ll also need to install its companion management tool, the LDAP Replication Manager. (As with the

SecureControl Manager, you can install the LDAP Replication Manager on either Solaris or Windows NT.) Chapter 5, “Integrating LDAP Directories and the Entitlements Database” for additional information.

- Redundant Authorization Servers, running on either Windows NT or Solaris. ClearTrust SecureControl provides fail-over among multiple Authorization Servers to ensure robust run-time operations and avoid a single point of failure. The basic installation of the ClearTrust Servers (first item under “Minimum Installation Process” on page 11) provides two Authorization Server processes running on the same machine to provide redundancy, but you can also implement as many stand-alone Authorization Servers as you need for load-balancing purposes. See “Chapter 6, “Redundant Authorization Servers” on page 111” for details.
- ClearTrust SecureDetector, an integrated component of the ClearTrust solution, that monitors application-layer (rather than network-layer) threats to your environment. See Chapter 7, “Installing the ClearTrust Management Tools” on page 121 for information about installing and configuring SecureDetector, and see the *Policy Administrator’s Guide* for information about using SecureDetector to detect and take action against threats.
- Additional custom applications created by using the ClearTrust SecureControl API. Such applications can interface directly to the ClearTrust SecureControl servers to authenticate users or application processses. Examples of uses include batch loading users from other systems or custom integrations with work-flow applications. See the *Developer’s Guide* for additional information.
- Additional custom Web server plug-ins that you develop yourself, to extend the functionality of the ClearTrust Web server Plug-in. See Chapter 1, “Using the ClearTrust Web Server Plug-in API” in the ClearTrust Developer’s Guide for information about using the ClearTrust Plug-in API.

This is just a brief overview. The ClearTrust solution includes numerous examples to get you started, including CGIs and Servlets, sample code, and the like that you can customize for your own environment. You’ll find these on the ClearTrust SecureControl CD. Be sure to look through all associated Readme files for additional information.

About this Guide

Conventions

The following conventions are used in this guide:

TABLE 1-1: Typographical Conventions

Convention	Meaning
`courier text`	Courier font denotes filenames, directory names, pathnames, or commands.
`italic text`	Italic font denotes variables or field values in command strings and the titles of other guides.
`UPPER CASE`	Upper-case denotes environment variables or Oracle commands, including SQL commands.
`[ ]`	Optional information appears in brackets.
`...`	Ellipsis in code listings denote continuation of a file or command.

Intended Audience

This *Installation and Configuration Guide* is designed to help system administrators, security admstrators, database administrators—or anyone else involved with implementing or integrating security components in a distributed network environment—get started using Securant's ClearTrust solution.

Related Documents

For more information, see the following guides, available on the Securant ClearTrust installation CD:

- *Policy Adminstration Guide*
- *Developer's Guide*

Your Comments Are Welcome

Please send comments, corrections, and suggestions about this guide to *pubs@securant.com*.

Chapter 2
ClearTrust SecureControl Servers

This chapter details the system requirements for running ClearTrust SecureControl and provides instructions for installing the ClearTrust SecureControl servers on either the Windows NT or Solaris operating system. It includes the following sections:

- "Installing ClearTrust SecureControl on Solaris"
- "Installing ClearTrust SecureControl on Windows NT"
- "Next Steps"

TABLE 2-1: ClearTrust SecureControl Server Requirements Summary

	Solaris	Windows NT
Operating system	Sun Solaris 2.6	Windows NT Server 4.0
Patches	Sun Solaris 2.6 Recommended Cluster	Service Pack 6
Hardware platform	Sun SPARC Ultra	Pentium II, 300-Mhz (or faster)
Storage (minimum)	1.2-Gbyte	500-Mbyte (installation only)
Peripherals	CD-ROM drive	CD-ROM drive
Memory	256-Mbyte RAM	256-Mbyte RAM
Supporting software	JRE 1.2.2	JRE 1.2.2
Additional software	SUNWarc, SUNWbtool, SUNWhea, SUNWlibm, SUNWlibms, SUNWsprot, SUNWtoo, SUNWmfrun	na
Database server	Oracle 7.3.4 for Sun Solaris 2.x (included) or Sybase Adaptive Enterprise Server 11.9.2	na; Oracle 7.3.4 embedded in product
Utilities	GNU Zip (gzip), tar, host terminal access or X server software (for example, Hummingbird)	na
Network	TCP/IP connectivity and access to DNS	TCP/IP connectivity and access to DNS

Installing ClearTrust SecureControl on Solaris

Installing ClearTrust SecureControl on the Solaris operating environment involves a wide range of administrative tasks, including installing and configuring a database management system. Some of the tasks require you to be logged on the Solaris system as super user (root), some tasks require you to be logged on as owner of the database installation (user *oracle* or user *sybase*.)

IMPORTANT NOTE: Pay close attention to how you're logged on at any given time to ensure that you install all components with the appropriate permissions. You should only be logged on as the `root` user when you create directory structure and prepare mount points, and when you create your database user accounts (for Oracle or Sybase installation) and the user account for the ClearTrust SecureControl software installation (*ctrust*).

The instructions in this section take you through the process of installing ClearTrust SecureControl on the Solaris. In general, the process involves:

- Preparing the Solaris operating environment for database installation
- Installing the database
- Setting environment variables for Solaris and database interaction
- Configuring and tuning the database instance
- Installing the ClearTrust Servers

Before you begin, review the "ClearTrust SecureControl Server Requirements Summary" on page 15 (Table 2-1) and "Before You Begin" on page 16.

Before You Begin

As with any other Solaris applications you'd install, you should plan out mount points, directory structure, pathnames, and other such details before you begin. Verify that your system meets the minimum requirements listed in Table 2-1. If you plan on setting up ClearTrust SecureControl to use automated e-mail for system alerts, you'll need to know the host name of your SMTP server.

You'll need `root` access for several portions of the installation, starting here:

1. Log on to the Solaris server as user *root*.

2 Verify that Solaris version 2.6 with the recommended Solaris 2.6 patch cluster is installed by using the `showrev` command with the patch argument:

```
$ showrev -p
```

If the patch is not installed, you should apply it now. You can obtain it from Sun's Solaris patch site at http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access. See the Readme included with the patch cluster for instructions.

3 Run the package information command and verify that all packages listed in Table 2-1 ("ClearTrust SecureControl Server Requirements Summary" on page 15) are installed:

```
pkginfo -i
```

4 Create a Solaris user account under which to install and run the ClearTrust SecureControl servers. Securant recommends using *ctrust* as the name for this user account. Whatever the name you choose, give the *ctrust* user permission to run the gzip and tar utilities.

5 Create a directory for the ClearTrust SecureControl (*securant)* installation. Use naming conventions consistent with other user home directories on your Solaris system. For example:

```
/usr/app/securant
```

With this preliminary setup and system requirements validation complete, you can begin preparing Solaris more specifically for the database you plan to install. Continue with the instructions for "Installing Oracle Database Server" or "Installing the Sybase Adaptive Server Enterprise" on page 28 as appropriate.

Installing Oracle Database Server

These instructions presume that you are familiar with Oracle Database Server in the Solaris environment. Specific steps may vary, depending upon version releases and patches. Be sure to refer to the *Oracle Installation Guide* for additional information on system requirements, product requirements, disk space, and memory requirements.

In addition, for detailed information about any of the Oracle installation steps, be sure to refer to the Oracle documentation, specifically:

- Oracle *Administrator's Reference for Sun SPARC Solaris 2.x*
- *Oracle Installation Guide* for information about setting the environment and creating an OFA (optimal flexible architecture) directory structure.

You can acess these and other Oracle manuals online at Oracle Technology Network (http://technet.oracle.com).

To set up Solaris for the Oracle database:

1. Log in as the `root` user and configure the Solaris kernel for shared memory; shared memory must be large enough to accommodate the Oracle system global area (SGA). You can examine the `/etc/system` file to see the existing shared memory configuration in the file (if it's already been configured).

   Oracle Corporation recommends setting the shared memory parameters as high as possible for the operating system (but not too high or the machine won't boot up). Oracle's installation manual for Oracle 7.3.4 states that the total allowable shared memory is determined by the formula `shmmax*shmseg`. However, the actual values will depend upon your hardware—RAM installed, for example, and your init*sid*.ora configuration. The recommended minimum values listed in Oracle's documentation for Release 7.3.4 are based on the assumption that the default init*sid.*ora is used to run a single instance (see "Shared Memory Values for Oracle" on page 162 for the recommended settings).

   The ClearTrust SecureControl installation process includes modifying several settings in the database initialization file, and the recommendations for shared memory settings are based upon the higher values. An example of shared memory settings as listed in an `/etc/system` file could be:

   ```
   set shmsys:shminfo_shmmax=32769156
   set shmsys:shminfo_shmmin=1
   set shmsys:shminfo_shmmni=100
   set shmsys:shminfo_shmseg=20
   set semsys:seminfo_semmns=200
   set semsys:seminfo_semmni=100
   ```

2. Add the `dba` group to the `/etc/group` file by using the Solaris operating system groupadd administration utility:

   ```
   $ groupadd -g 101 dba
   ```

   If the "dba" group already exists on your Solaris system and you want to simply add the new oracle user account to the existing group, be aware of the DBA privileges that you'll in effect be giving to group members and on what Oracle systems. If you want to isolate the privileges to the ClearTrust schema to a select sub-set of DBAs (in an environment with multiple Oracle systems), you

can create another name for the dba group, in which case the Oracle Installer will re-link the Oracle 7 Server during installation (the dba group name is hard-coded into the file $ORACLE_HOME/rdbms/lib/config.s).

3. Create an *oracle* software owner account that includes properties for user ID (UID), group ID (GID; same as that in previous step); home directory, and the logon shell. For example:

   ```
   $ useradd -u 101 -g 101 -c oracle -d /export/oracle -s
   /usr/bin/csh
   ```

   When you finish this step, the system's `/etc/group` file will contain a completed entry for the dba group and will include the *oracle* user:

   ```
   dba::101:oracle
   ```

4. Determine how you will layout the Oracle software and datafiles on your system. See Oracle installation manuals for information and guidelines for configuring an OFA (optimal flexible environment) compliant directory structure on Solaris, and then create directories for the Oracle installation accordingly. For an OFA-compliant directory structure, you'll create four directories: one for the Oracle software, and three for the datafiles.

5. Prepare mount points by entering these commands:

   ```
   chown -R oracle mount_point
   chgrp -R dba mount_point
   chmod -R 755 mount_point
   ```

NOTE: step 6, 7, and the table need some help. can we provide a simple consistent "for instance" example of what these might look like? and then carry through each step..

6. Locate (or create) the local /bin directory (typically found at `/usr/local/bin`. You'll need to know the location when you set environment variables. (After installation, the `root.sh` script will place the oraenv (or coraenv for the C shell) in this directory so that regardless of changes to the PATH environment variable, the `oraenv` (or `coraenv`) scripts will continue to function.)

7. Set environment variables for the *oracle* software owner. For the Bourne and Korn shell, set variables in the .profile file; for the C shell, set variables in the .cshrc file (see the Table). (When you change any settings in these files, be sure to source the file (`$ ..profile` or `% source .cshrc`) so that the *oracle* user's session invokes the settings for the current session.)

TABLE 2-2: Example shell scripts for *oracle* user.

Bourne or Korn shell (.profile file)	C shell (.cshrc file)
`PATH=$PATH:/sbin:/bin:/usr/openwin/bin:/usr/local/bin`	`setenv PATH /bin:/usr/bin:/usr/local/bin:$ORACLE_HOME/bin`
`ORACLE_BASE=/export/app`	`setenv ORACLE_BASE /export/app`

Bourne or Korn shell (.profile file)	C shell (.cshrc file)
ORACLE_HOME=/export/app/oracle/product/734	setenv ORACLE_HOME /export/app/oracle/product/734
ORACLE_ASK=NO	setenv ORAENV_ASK NO
ORACLE_SID=CT	setenv ORACLE_SID CT
TERM=vt100	setenv TERM vt100
ORACLE_TERM-vt100	setenv ORACLE_TERM vt100
umask 022	umask 022

8 Mount the Oracle software CD. (Some versions of Solaris may not require this step. For example, if you're using Solaris Volume Management software, the CD-ROM is mounted automatically to the `/cdrom/oracle` when you insert the disk into the drive.) To manually mount the CD-ROM, enter:

```
$ su root
# mkdir /cdrom
# mount -r -F hsfs device name/cdrom
# exit
```

As shown in the example above, you must be logged on as root to mount or unmount the drive.

Be sure to unmount the drive later, when the installation is complete, before removing the CD-ROM from the drive.

You've now completed preparing the Solaris operating environment for the Oracle database server installation, and you can install the Oracle database. The Oracle installation process is guided by the scripts that Oracle Corporation provides with its software, so choices may vary based on the version of the Oracle Installer you are using. These instructions focus on choices that will effect ClearTrust SecureControl. For additional information about the Oracle installation, consult the Oracle manuals or your Oracle on-site DBA.

During the installation, you'll be prompted to provide names for the database instance, the ClearTrust schema, and other database system objects. You can accept the defaults (many of which are simply "ct_*objectname*"), or you can provide new names if you like.

Securant recommends naming your database instance *CT*, and the instructions below and elsewhere in this guide are based on the assumption that you've done so. Replace this name throughout if necessary.

To install the Oracle Database Server:

1. Log in as user *oracle*.
2. Launch the Oracle Software Installation script from the mounted CD-ROM:

```
cd /cdrom/oracle/orainst
./orainst
```

The Oracle installation script begins. You'll be prompted to select various options as choices as the script proceeds. Use the settings shown in the table, unless you have reason to do otherwise. By choosing the default installation type, the Oracle Installer prompts you for the values of ORACLE_BASE, ORACLE_HOME, and ORACLE_SID, and then uses values for numerous system log files based on these values. See the Oracle Installation manual for complete information.

TABLE 2-3: Oracle Installation Prompts

Prompt	Selection
Installation type	Default
Installation activity choice	Install, Upgrade, or De-Install Software
Installation Options	Install New Product—Create DB Objects
Installation Locator	
Products selected in the Software Asset Manager	Oracle7 Server RDBMS 7.3.4 PL/SQL SQL*Net SQL*Plus TCP/IP Protocol Adapter
Tablespaces	Accept the default location and sizes for the tablespaces, rollback segments, and log files you created multiple mount points
Group to act as dba	dba
Group to act as operator	dba
Storage type	Filesystem-based database
Distribute control files over 3 mount points?	No (unless you created multiple mount points for OFA-compliant structure during Solaris setup)

Prompt	Selection
Mount point	/oracle (if you followed earlier steps; otherwise, enter the directory name that you created for Oracle)
Character set	US7ASCII
Set password for internal users	No
MTS configured and SQL*Net Listener auto start	No

The installation should complete without errors. The Oracle installation script creates an `oraenv` (or `coraenv` for the C shell) command file that contains values for the Oracle environment variables. You should make sure this file is located in the *oracle* user account's `local/bin` directory, separate from the Oracle software home directory. In addition, the Oracle installation script creates sample user startup files—`.login` (or `.cshrc` files, for the C shell) and `.profile` (Bourne and Korn shell) files—that automatically set the ORACLE_SID, ORACLE_HOME, PATH, and TERM variables for the *oracle* software owner. Add the oraenv or coraenv command line to the startup file of the .login (or .profile) for the oracle user account.

To setup the environment variables:

1. Log in as root.
2. Execute the root.sh script:
3. Source ORACLE cshrc file
   cd $ORACLE_HOME/orainst
   ./root.sh
4. Verify that environment variables are correct.
5. Enter the path of the local/bin. The installation script prompts you to confirm that the root home dir doesn't match the Oracle home directory. This isn't a problem, so you can continue.
6. Enter 'y' to continue.

   At the end of the installation process, the Oracle Installer script automatically starts the database instance. At any time, you can confirm that the Oracle instance is running by checking for "*oracle*_ct.." processes:

   ```
   ps -ef | grep ora
   ```

You can also startup and shutdown the Oracle database instance using the Oracle Server Manager (or SQL*Plus) as often as you like without creating errors. As the database starts, you'll see system global area (SGA) and other settings display, along with a notification that the database has started.

To configure the Oracle database server:

1. Log in as *oracle*.
2. Start the database instance by launching the Oracle Server Manager, located in the `oracle/bin` directory:

   ```
   svrmgrl
   ```

3. When the Server Manager prompt displays (`SVRMGRL>`), connect to the database, shutdown, and then startup:

   ```
   SVRMGR> connect internal
   SVRMGR> shutdown
   SVRMGR> startup
   ```

   As the database instance starts up, you'll see the SGA (system global area) parameters display. Verify that the database started without errors.
4. Exit from the Server Manager:

   ```
   SVRMGR> exit
   ```

5. Start the Oracle listener:

   ```
   lsnrctl start
   ```

   The listener should start, displaying the SQL*Net configuration parameters without any errors.
6. To verify that the database is up running, logon to the database using SQL*Plus:

   ```
   sqlplus system/manager@$CT
   ```

   You should should connect without errors to the database and display the SQL*Plus prompt:

   ```
   SQL>
   ```

7. Change the Oracle default `system` password to something more secure.

   ```
   sqlplus system/manager@$CT
   alter user system identified by new_password;
   ```

8. Change the Oracle default `sys` password to something more secure.

   ```
   sqlplus sys/change_on_install@$CT
   alter user sys identified by new_password;
   ```

9 At the SQL*Plus prompt, select rows from the database using the following command:

```
SELECT object_name FROM user_objects;
```

10 The command should execute and return a listing of database objects.

11 With the SQL*Plus prompt still displayed, you can now run the script to create the tablespace for ClearTrust. From another shell window, navigate to the /scripts sub-diretory on the ClearTrust SecureControl CD and copy the cttablespace.sql file from the CD to the same directory in which you're running SQL*Plus, or create a path to the file on the CD. Execute the script at the SQL*Plus prompt:

```
sqlplus system/system_user_password@CT
SQL>@cttablespace
```

The CTtablespace script will prompt for sizes and locations for several tablespaces, indexes, and other database structures. Accept the defaults, or note the values and locations you enter here; you'll need to know these details later during the ClearTrust server installation, and the values and locations must match what you enter here exactly.

Oracle Installer Prompt	Variable
Absolute mount	/export/home/oracle
pointORACLE_SID for ClearTrust	CT
Size of index tablespace	50
Size of data tablespace	200
Size of SecureDetector index tablespace	50
Size of SecureDetector data tablespace	200
Name for ClearTrust schema owner	ct_owner
Password for ct_owner	your_choice_just_remember_it
Temporary tablespace for ct_owner	temp
Data tablespace	ct_data
Index tablespace	ct_index
SecureDetector	ct_sd_data
SecureDetector index tablespace	ct_sd_index

12 When the script finishes executing, the SQL*Plus prompt re-displays. You can verify that the ClearTrust SecureControl schema and Oracle listener are up by issuing the following command:

```
connect ct_owner/ct_owner_password@CT
```

In a second, you should be connected to the ClearTrust instance, and the SQL> prompt should re-display.

Tuning Oracle's Configuration for ClearTrust

The initial database instance created for the ClearTrust SecureControl Entitlements database is based on Oracle's default initialization file ($ORACLE_HOME/dbs/`init.ora`), which is not intended for large production environments. You must modify the initialization file to create a more robust database instance.

To modify the configuration for the Oracle database installation:

1 Use a text editor to open the initialization file for the ClearTrust SecureControl instance. If you followed the defaults and recommendations, the filename is `initCT.ora` file, located in the $ORACLE_HOME/`dbs` directory.

2 Comment and un-comment lines as needed to implement the LARGE or LARGER settings for block buffers, shared pool size, and other settings that comprise the system global area. Specifically, you should enable the large or larger settings for db_buffers, shared_pool_size, and log_buffer.

3 If necessary, add the open_cursors and sort_area_size parameters to the end of the `initCT.ora` file, and set them to 500 and 1048576, respectively:

```
open_cursors=500
sort_area_size=1048576
```

4 For systems using replication you must also add the following to the `initCT.ora` file:

```
job_queue_processes=2
```

Changes you make to the Oracle initialization file won't take effect until you re-start Oracle (shutdown and then startup), as follows:

5 Start Oracle Server Manager:

```
svrmgrl
```

6 When the Server Manager prompt displays, shutdown and re-start the Oracle database instance by entering these commands:

```
SVRMGR> connect internal
SVRMGR> shutdown
SVRMGR> startup
SVRMGR> exit
```

7 The database should start without any errors; the database's new SGA parameters will display during startup.

With the database now up and running with the larger SGA settings, you can now modify the size of the rollback segments: If these segments aren't properly sized, your Entitlements database may one day run out of expansion space. Because you can't resize the rollback segments while they're in use, you must create new rollback segments (to the size you wish), take the existing segments offline, and move the larger sized segments in place, as instructed below.

To increase the size of the rollback segments for production:

1 Start up the server manager and connect internally.

```
svrmgrl
```

```
SVRMGR> connect internal
```

2 Create a new tablespace for the new rollback segments (do not enter carriage returns for the line wrap):

```
SVRMGR>CREATE TABLESPACE rbs_big datafile
'/app/oracle/oradata/ct/rbs_big01_ct.dbf' size 80M ONLINE;
```

3 Create new rollback segments with chosen storage parameters. This example uses fairly large values:

```
SVRMGR> CREATE ROLLBACK SEGMENT rb01 TABLESPACE rbs_big
STORAGE (INITIAL 1M NEXT 1M optimal 2M);
SVRMGR> CREATE ROLLBACK SEGMENT rb02 TABLESPACE rbs_big
STORAGE (INITIAL 1M NEXT 1M optimal 2M);
```

```
SVRMGR> CREATE ROLLBACK SEGMENT rb03 TABLESPACE rbs_big
STORAGE (INITIAL 1M NEXT 1M OPTIMAL 2M);
```

```
SVRMGR> CREATE ROLLBACK SEGMENT rb04 TABLESPACE rbs_big
STORAGE (INITIAL 1M NEXT 1M OPTIMAL 2M);
```

4 Bring the new rollback segments on-line:

```
SVRMGR> ALTER ROLLBACK SEGMENT rb01 ONLINE;
SVRMGR> ALTER ROLLBACK SEGMENT rb02 ONLINE;
SVRMGR> ALTER ROLLBACK SEGMENT rb03 ONLINE;
SVRMGR> ALTER ROLLBACK SEGMENT rb04 ONLINE;
```

5 Edit the init*CT*.ora file to enable the new rollback segments and remove (or comment-out) references to the segments currently in use. Again restart the database using these commands:

```
SVRMGR> connect internal
SVRMGR> shutdown
SVRMGR> startup
SVRMGR> exit
```

The database should start without errors and the database's SGA parameters should be displayed.

6 Verify that the rollback segments are online and ensure that the new segment names rb01-rb04) are present:

```
sqlplus system/password@CT
```

```
SELECT SEGMENT_NAME, OWNER, TABLESPACE_NAME, STATUS FROM
dba_rollback_segs;
```

7 Exit SQL*Plus. Start up the Server Manager and take the previous RBS tablespace off-line:

```
svrmgrl
SVRMGR> connect internal
SVRMGR> ALTER TABLESPACE RBS offline;
```

8 Drop the old rollback segments (r01, r02, r03, r04) to free space and minimize confusion: your initial rollback segments are now offline and won't be used.

```
SVRMGR> DROP ROLLBACK SEGMENT r01;
SVRMGR> DROP ROLLBACK SEGMENT r02;
SVRMGR> DROP ROLLBACK SEGMENT r03;
SVRMGR> DROP ROLLBACK SEGMENT r04;
```

9 Resize the temporary tablespace data file:

```
SVRMGR> ALTER DATABASE DATAFILE
'/app/oracle/oradata/ct/temp01.dbf' resize 80M;
```

10 Alter the default sizing for temporary segments by typing:

```
SVRMGR> ALTER TABLESPACE TEMP DEFAULT STORAGE (INITIAL 2M NEXT
2M PCTINCREASE 0);
```

11 Exit the Server Manager by typing `exit` at the `svrmgr>` prompt.

You have successfully installed and configured your Oracle database. You can now start the Oracle database.

To startup Oracle Database Server:

1 Use Server Manager to startup the database:

```
svrmgrl
SVRMGR> connect internal
SVRMGR> startup
```

2 As the database instance starts, you'll see messages about system global area configuration and the like. When the startup completes, you can close out the Server Manager session and launch the Oracle TNS listener process:

```
SVRMGR> exit
lsnrctrl start
```

Now both the Oracle database server and listener process are running. You'll need to do this whenever you start the ClearTrust SecureControl Servers.

See "Installing the ClearTrust SecureControl Servers" on page 35 to continue the installation process.

Installing the Sybase Adaptive Server Enterprise

This section tells you how to install the Sybase Adaptive Server Enterprise 11.9.2 database. These instructions presume you are familiar with Solaris and with server software installation. If you are using the Oracle database, see "Installing Oracle Database Server" on page 17.

Setting up the Operating System for Sybase

Before you begin this procedure, consult the *Sybase Installation and Configuration Guide* for information about system requirements, product requirements, disk space, and memory requirements. You should also closely review the *Sybase Installation and Configuration Guide* if you're not an experienced Sybase DBA. Sybase manuals are available online at http://sybooks.sybase.com/asg1192e.html.

To set up the operating system for Sybase installation:

1 Log in as *root*.

2 Add user *sybase*. For example:

```
Useradd -u 102 -g 102 -c sybase -d/sybase-s/usr/bin/hash
```

3 Create a directory for the Sybase installation. This directory must exist before you begin installing Sybase; the Sybase installation routine won't create this directory for you, and if it doesn't exist, Sybase will exit with an error. Securant recommends using `/sybase` as the mount point if possible—the actual Sybase installation directory may be linked to /sybase.

To setup the environment:

4 Log in as user *sybase*. Set the environment variable `sybase` to the directory in which you plan to install the Sybase software. For example, in a Bourne shell you might enter:

```
export SYBASE=/app/sybase
```

while in a C shell, you might enter:

```
setenv SYBASE /app/sybase
```

This environment variable is used by the Sybase installation and setup utilities. For additional information about the Sybase environment variables you need to set, see the *Sybase Installation and Configuration Guide*. Follow the installation requirements detailed in Chapter 2 of the *Sybase Installation and Configuration Guide* to ensure a successful installation.

To install and configure the Sybase Adaptive Enterprise Server:

1 Log in as *sybase*.

2 Use the sybsetup utility to unload or copy the Sybase software from the distribution media to the machine.

```
/cdrom/cdrom0/sybsetup
```

3 The script prompts you for the location of the installation source. Enter the absolute path of the `sybimage` file located in the installation CD root directory

4 Follow the instructions to install the server. Install only the Adaptive server and any language modules you need. You can also install the `sybsetup` as part of your Sybase installation. No other Sybase components are necessary.

5 Return to the `sybsetup` main menu and launch the `srvbuild` program by choosing the "Build new server" option. Choose to build the Adaptive server only. Parameters for this build process are completely system dependent. Securant recommends consulting your database administrator to obtain appropriate values.

The `srvbuild` utility enables you to specify and configure the master database. Here's an example of a path name and device name you might use to setup the server:

`/sybase/master.dat` (for master database path)
`/sybase/tempdb/systemproc.dat` (for the systemprocs path)

6 The `srvbuild` utility prompts you to select localization. Run localization to adjust the language options and sort order. The appropriate settings are *Dictionary Order*, *Case Insenstive*, and *Accent Insensitive*.

The default installation of Sybase allocates only 2MB for the tempdb, which is not enough for SecureControl, so you should increase the size of the tempdb to 20MB by using the following command:

```
alter database tempdb log on device_name = 20
go
```

7 For performance reasons, keep database devices and log devices separate.

Preparing to Install the ClearTrust SecureControl Database

Prior to installing the ClearTrust SecureControl database, you must prepare the Sybase server. Make sure Sybase Adaptive Enterprise Server is running by entering this command:

```
/sybase/install/showserver
```

If Sybase isn't running, you can start the server by entering:

```
/sybase/install/startserver -f
/sybase/install/RUN_servername
```

The ClearTrust SecureControl CD-ROM contains a SQL script for setting up the ClearTrust Secure database. The script contains all the information needed to create a 250-MB Sybase database device named `ct`, with specific device names and pathnames. The file is located on the ClearTrust CD-ROM at:

```
/cdrom/cdrom0/DB/setupsybase.sql
```

Open this script and edit all the variables as needed to ensure a succesful database creation in your environment. This table lists the settings in the current version of the `setupsybase.sql` script; always double-check the contents of this script file before running, even if you think you want to accept the defaults.

TABLE 2-4: Sybase Database Creation Script

Sybase Variable	Defaults
Path for Sybase installation	/sybase

Sybase Variable	Defaults
Sub-directory for installation process	/sybase/tempdb
Default name of ClearTrust database	ct
Default size of `ct` data tablespace	250
Database device name	ctdev
Virtual device number (vdevno) for database device	4
Log device name	ctlog
Virtual device number (vdevno) for log device	5
Database segment (1 of 4)	CT_DATA, ct, ctdev
Database segment (2 of 4)	CT_INDEX, ct, ctdev
Database segment (3 of 4)	CT_SD_DATA, ct, ctdev
Database segment (4 of 4)	CT_SD_INDEX, ct, ctdev

When you're finished validating and modifying the script, you can run it by entering:

```
/sybase/bin/isql -Usa -Psapassword -Sservername -
i/cdrom/cdrom0/DB/setupsybase.sql
```

The setupsybase.sql script creates a 250-Mb database device that contains a 200-MB database identified as "ct," the ClearTrust SecureControl database. You can modify this script to change the specifics (name, size, and so forth).

Customized Preparation

8 To prepare a customized ClearTrust SecureControl database, execute the system procedure by launching the Sybase `isql` utility:

```
$SYBASE/bin/isql-Usa-P -Sdatabaseservername
```

```
/sybase/bin/isql -Usa -Ppassword -Sdatabaseservername -
i/cdrom/cdrom0/db/setupsybase.sql
```

```
password refers to the SA account password
```

```
databaseserver refers to sybase server name
```

9 Use the Sybase system procedure `disk init` to create a database device. At the > prompt, type in a SQL statement and system procedure with its parameters.

```
disk init
name=”device_name”,
physname=”physicalname”,
vdevno=virtual_device_number,
size=number_of_2k_blocks
```

For performance reasons, keep database devices and log devices separate.

For example, the following statements and system procedure creates a database device of size 250MB and a device for the Transaction Log.

```
Use master
go
disk init name=”ctdev”,
physname=”/sybase/tempdb/ctdev.dat”,vdevno=4,size=128000
go

disk init name=”ctlog”,
physname=”/sybase/tempdb/ctlog.dat”,vdevno=5,size=38400
go
```

10 Create a database on the database device. Use the device you created previously as the primary storage for ClearTrust SecureControl:

```
create database database_name
[on{default|database_device}[=size_in_MB]
[,database_device [=size_in_MB]]...]
[log on database_device[=size_in_MB]
[,database_device[=size_in_MB]]...]
[with override]
[for load]
```

For example:

```
use master
go
create database ct on ctdev=225
log on ctlog=60 with override
go
```

The size for the database creation is in Mbytes. `ct` is the database instance that will be set in the `Default.conf` file of Clear Trust:
`securecontrol.db.instance=ct`

11 Sybase logs the Transactions of the SecureControl database. Once the Transaction Log is full, Sybase stalls all other requested Transactions. There are two solutions to this problem. For the more sophisticated solution, refer to the section on Transaction Logs in your Sybase Manual. For a simpler solution, you can turn on 'trunc log on chkpt' by:

```
sp_dboption ct, trunc, true
go
```

12 Create database segments:

>sp_addsegment *segname*, *dbname*, *devname*

For example:

```
Use ct
go
sp_addsegment CT_DATA, ct,ctdev
go
sp_addsegment CT_INDEX, ct,ctdev
go
sp_addsegment CT_SD_DATA, ct,ctdev
go
sp_addsegment CT_SD_INDEX, ct,ctdev
go
```

You must create a minimum of four segments: two for data and the other two for indices. You supply the segment names (segname), which must match the values that you later specify in the ClearTrust SecureControl server install process.

13 Create Logins and Users for the database using the addlogin system procedure. The syntax is as follows; note that Sybase requires passwords of at least six alphanumeric characters:

```
sp_addlogin loginame,passwd[,defdb[,deflanguage[,fullname]]]
```

For example, at the SQL prompt:

```
use ct
go
sp_addlogin 'CT_OWNER','ct_owner',ct
go
sp_addlogin 'CT_ADMIN','ct_admin',ct
```

```
go
sp_addlogin ‘CT_USER’,’ct_user’,ct
go
```

To add a user or alias login name, use the `adduser` system procedure:

```
sp_adduser loginame[,name_in_db[,grpname]]
```

For example:

```
use ct
go
sp_adduser ‘CT_ADMIN’,’ct_admin’
go
sp_adduser ‘CT_USER’,’ct_user’
go
```

14 Assign **`sso_role`** and **`sa_role`** to CT_OWNER (or the owner of the database):

```
sp_role{“grant”|”revoke”},
{sa_role|sso_role|oper_role},loginame
```

For example:

```
sp_role’grant’,sso_role,CT_OWNER
go
sp_role’grant’,sa_role,CT_OWNER
go
```

15 Assign **`sa_role`** to CT_ADMIN (or the admin of the database):

```
sp_role’grant’,sa_role,CT_ADMIN
go
```

16 Assign **`oper_role`** to CT_USER:

```
sp_role’grant’,oper_role,CT_USER
go
```

17 Transfer database ownership to CT_OWNER:

```
sp_changedbowner loginame[,true]
```

For example:

```
use ct
go
sp_changedbowner CT_OWNER
go
```

If the database returns this message:

```
The proposed new db owner already is a user in the database
```

then CT_OWNER already owns the database. Proceed to "Installing the ClearTrust SecureControl Database."

You have successfully installed and configured your Sybase database.

Set the Sybase System Administrator Password

```
Before continuing, you should create a new password for the
sa (Sybase system administrator) account. This account was
created during the Sybase software installation; by default,
the initial value for the password is NULL. To ensure the
security of your database installation, you should log in as
sa and set a pasword with the sp_password procedure:
```

```
$SYBASE/bin/isql -Usa -P -Sservername
1>sp_password null, new_password
2>go
```

See "Installing the ClearTrust SecureControl Servers" to continue the installation process.

Starting Sybase Adaptive Server Enterprise

To start Sybase:

1. Enter the startup command sequence for Sybase as follows, where */sybase* is your Sybase installation directory and *servername* is the name of your database server:

   ```
   /sybase/install/startserver -f
   /sybase/install/RUN_servername
   ```

   As the database instance starts, you'll see some messages to that effect. You can install the ClearTrust SecureControl servers or start them if they're already installed.

Installing the ClearTrust SecureControl Servers

This section provides step-by-step instructions for installing the ClearTrust SecureControl servers on Solaris, using the script provided on the ClearTrust SecureControl CD.

These instructions presume you've already installed and configured a relational database management system as described in "Installing Oracle Database Server" on page 17 or "Installing the Sybase Adaptive Server Enterprise" on page 28, and that the database and its listeners have been started.

To run the ClearTrust SecureControl Solaris server installation script:

1. Log in to Solaris as the user *ctrust* (presuming you followed the steps under "Before You Begin" on page 16. Whatever the name, the user must have permission to write in the directory where you intend to install the ClearTrust SecureControl software.)
2. Navigate to the `/scripts` sub-directory on the ClearTrust SecureControl CD-ROM and start running the script from the command line (the script is text-based; you cannot run in a windowing environment):

   ```
   ./install.script
   ```

   In a second, you'll see a display prompting you to log in as the user that will own ClearTrust server components.
3. Press Enter to continue with the installation. The installation script prompts you for the name of Securant home name.
4. Enter the name of the directory into which you'd like to install ClearTrust; the default is `/opt/securant`. Depending upon your Solaris configuration, you may need to change the name. You'll then be prompted for the type of database you'll be running.
5. Enter 'O' for Oracle or 'S' for Sybase.

   If you enter O, the script checks for the Oracle installation and the prompts you to enter the name of your ORACLE_HOME directory. Enter the directory you defined during the Oracle installation. The script then prompts for the Oracle base. Enter the directory you defined as ORACLE_BASE during the database installation.

   If you enter S, you'll be prompted for the home diretory of the Sybase directory. Enter the home directory for the Sybase database.
6. The script then prompts for the name of the ClearTrust database; the default is CT.
7. The installation script checks for the Oracle system identifier (Oracle_SID) and then prompts you for the name of the Oracle database instance or for the name of the Sybase database.

8. Enter the name of the ClearTrust database instance (for Oracle users this is simply `ORACLE_SID`). The script displays the entries you've made so far and prompts you to continue; it might look somthing like this:

```
SECURANT_HOME: /opt/securant
DB_HOME: /mnt/jill1/app/oracle/product/7.3.4
ORACLE_BASE: /mnt/jill1/app/oracle
DB_INSTANCE: CT
JAVA_HOME: /opt/securant/jre
```

9. Press Enter to continue, and you'll be prompted to enter y or n to continue with the installation. (You can enter Ctrl-C to quit if the entries aren't correct.)
10. Enter 'y' to continue with the installation. The script prompts for the root directory for the ClearTrust installation.
11. Enter the path to the root directory on the ClearTrust installation disk. The script prompts again to continue with the install.
12. Enter 'y' to continue with the installation. The script provides a status message when it finds the software distribution and displays a progress message as the software is coiped to the destination directory. At the end of the process, the script prompts you to install the Securant ClearTrust API client distributions.
13. Enter 'y' to install the ClearTrust API Client software. The script notifies you of the progress as the API files are copied to the destination. When finished, it prompts you to install the ClearTrust Entitlements Manager software.
14. Enter 'y' to install the ClearTrust Manager software. The script notifies you of the progress as the files are copied; when finished, it prompts you to add the ClearTrust environment variable to the *ctrust* user's profile.
15. Enter 'y' to setup the *ctrust* user's C shell (.cshrc) environment for ClearTrust. The shell is modified. The script then prompts to create the ClearTrust configuration file.
16. Enter 'y' to create the configuration file. The next several prompts will prompt you for values for the database objects that you created during database installation. You must ensure that the values for the ClearTrust index, the database schema and username, event log name, and other database-configuration names match those you specified in the CTtablespace.sql script (if you installed the Oracle database), or, if you installed the Sybase database, values must match the names of the four Sybase segments you added when preparing the database.

    The data you entered will be displayed as the default selection, so in most cases you can simply accept the defaults.

17. After you've entered (or accepted) values for data tablespace (ct_data), ClearTrust index (ct_index), event log name (ct_sd_data), database schema (ct_owner), database administrator (ct_admin), ClearTrust user (ct_user), and their respective passwords, the script will then prompt you to create an internal index for the Entitlements Server; the default is Yes (will create an internal index).
18. Press Enter to accept the default and create an internal index. The script then prompts you enter the type of database (Oracle or Sybase); the default is Oracle.
19. Press Enter to accept the default if you've installed the Oracle database; or enter Sybase if you've installed the Sybase database server.

    If you select Oracle, you'll be prompted for a SQL*Net database identifier; the default is CT.

    If you entered Sybase, you'll be prompted for the hostname of the Sybase server, the database name (default is *ct*), and the connect string that the JDBC driver will use to connect to the Sybase database. Enter the JDBC name as:

    `jdbc:sybase:Tds:`*`servername:`*`4100:`*`ct`*

    The port number can be found in the `interfaces` file; convert the hexadecimal number (default is 4100). The script then prompts for the Voyager ORB namespace server.
20. Enter the fully qualified domain name of the machine on which you're installing ClearTrust (hostname.domain-name.com). The script then prompts for a port number for the Voyager namespace server; the default is 5600.
21. Enter a new value or press Enter to accept the default. The script then prompts for the name of the Entitlements Server; the default is "EntitlementsServer."
22. Enter a new name or simply press Enter to accept the default name. The script then prompts for the API server port number; the default is 5601.
23. Enter a new port number if appropriate or press Enter to accept the default. The script then prompts for sizing parameters for various logs, including API transaction and error logs; the default size is 1000K.
24. Press Enter to accept the default size, or enter a new size for all the logs. You'll then be prompted to size the number of active connections to the Authorization Server; the default is 5 active connections.
25. Enter a new value or press Enter to accept the default. You'll then be prompted for statement groups per connection; the default is 1.
26. Enter a new value or press Enter to accept the default and continue. The script then prompts for TCP server thread pool size; the default is 20 threads.

27 Enter a new value or press Enter to accept the default and continue. The script prompts for the maximum number of TCP connections allowed on the server; the default is 100 connections.

28 Enter a new value or press Enter to accept the default value and continue. You'll then be prompted to set the Authorization mode to active or passive.

The default mode is ***active***; in active mode, users can access resources unless they have been specifically declined access. On the other hand, ***passive*** mode is stricter in that access to every resource is checked, and only those who have been given explicit access to a resource will be able to access the resource.

Press Enter to accept the active (default) mode. The script prompts you to identify whether the Authorization server will be using SSL (secure sockets layer) for encrypted communications; the default is "yes," meaning SSL will be selected.

29 Enter "no" if you won't be using SSL, or simply press Enter to accept the default setting. The script prompts you to select whether SecureDetector logging will be enabled or not; the default is no, SecureDetector logging will not be enabled.

30 Press Enter to accept the default, or enter Yes if you plan on installing the SecureDetector component and wish to enable logging. The script prompts for the Authorization Server cache size; the default is 8000 entries for the Master Authorization server.

31 Press Enter to accept the default size, or enter the appropriate value for the number of entries you anticipate needing stored in the cache at any given time. (In an organization with hundreds of thousands of user accounts, you may wish to increase this number.) You'll be prompted for a User value for Authorization server cache; the default is 4000.

32 Press Enter to accept the default and continue. The script prompts for the number of Application Functions to be stored in the Authorization server cache; the default is 2500.

33 Press Enter to accept the default and continue. The script prompts for the number of Smart Rules to be stored in the Authorization server cache; the default is 250.

34 Press Enter to accept the default and continue. The script prompts for the number of User Properties to be stored in the Authorization server cache; the default is 0.

35 Press Enter to accept the default or enter a value for the number of User Properties you expect to implement. The script prompts for the number of Non-access Functions to be stored in the Authorization server cache; the default is 500.

36 Press Enter to accept the default. The script prompts for the user activity log level; the default is 10.

37 Press Enter to accept the default, or enter a 0, 20, or 30 to accept a different log level—the higher the number, the more information in the log. You'll then be prompted to enter the size of the user activity log; the default is 1000K.

38 Press Enter to accept the default, or enter a new value to increase or decrease the size of the log from this default. The script then prompts for the location (as a fully-qualified domain name) of Authorization server's dispatcher process.

39 Enter the fully-qualified domain name (*hostname.company-name.com*) of the machine on which ClearTrust is being installed. The script prompts you for the port number of the Encryption Key Server; the default is 5606.

40 Press Enter to accept the default, or enter a new value for the port number if you have reason to change it. The script prompts for the port number of the Authorization Server registration process; the default is 5607.

41 Press Enter to accept the default, or enter a new value for the port number if you have reason to change it. The script prompts for Authorization Server list port; the default is 5608.

42 Press Enter to accept the default, or enter a new value for the port number if you have reason to change it. The script prompts for a value for the lifetime of the session key; the default is 1 hour.

43 Press Enter to accept the default, or enter a new session key lifetime value. The script prompts for the type of encryption (if any) you wish to implement for the session key; the default is 3DES (Triple DES (data encryption standard).

44 Press Enter to accept the default, or enter a "b" to implement Blowfish encryption algorithm. Alternatively, you can enter "c" to send the key as cleartext (not encrypted; not recommended). The script prompts for the format for Plug-in communications; the default is IP.

45 Press Enter to accept the default and use IP addresses to map ClearTrust Services and Web Server Plug-ins, or enter "hostname" to use hostname values instead of IP addresses. The script prompts for the email address of the ClearTrust administrator.

46 Enter the email address of the administrator who should receive email notifications of ClearTrust system events, or simply press Enter to continue and leave this entry blank. The script prompts for the SMTP hostname that will support the email notifications.

47 Enter the fully qualified hostname of the SMTP server—for example, smtp-host.your-company.com—or press Enter to leave the value blank and continue. The script prompts for the password policy lifetime (for user passwords); the default is 60 days.

48 Enter a new value to change the maximum lifetime for user passwords, or press Enter to accept the default. The script prompts for your policy setting for user password history; the default is 8.

49 Enter a new value, or press Enter to accept the default of 8 previous passwords in the history file. The script prompts for the user account lifetime; the default is 365 days.

50 Press Enter to accept the default, or enter a new value that you wish to use as the default for all your security policies. The script continues, prompting you for license information.

51 Enter the license information you received from Securant. This is a long text string that includes the IP address of the server onto which you're installing, and was generated by Securant expressly for the server. After entering the license information, you'll be prompted for the license key.

52 Enter the license key information. The script notifies you that configuration information is being saved, and then prompts you to install the ClearTrust tables.

53 Enter Y to create the tables and continue with the installation. The script displays status and warning messages, notifying you that it will over-write any existing ClearTrust database.

If this is a first-time installation, enter *y* to continue. The script displays a status message as it creates the database and displays a notification when the installation is complete.

54 For security purposes, set permissions on the logs directory so that only the ClearTrust user can read/write/update it. After you start the ClearTrust server for the first time, you should also modify the permissions on the log files to 600 (read and write permissions for the owner only).

55 For a Sybase installation, you must run the `sql_server.sql` script, located in the /DB directory of the CD-ROM, as the last step of the installation. Presuming your Sybase files are located in a directory called `/sybase/bin/isql`, the command to execute the script is as follows:

```
/sybase/bin/isql -Usa -Ppassword -Shostname -
i/cdrom/cdrom0/DB/sql_server.sql
```

where *hostname* is the name of the machine on which you have installed the Sybase server software.

The ClearTrust SecureControl software installation on Solaris is now complete. All the entries you made during this process are stored in a default.conf file. (See “ClearTrust SecureControl Default.conf Parameter Settings” on page 137 for a listing of all settings, default values, and what they mean.) You can now start the ClearTrust SecureControl Servers.

Starting the ClearTrust Servers on Solaris

The database server must be up and running before you can start ClearTrust SecureControl servers. Start Oracle (see “Starting the Oracle Database Server” on page 28), or start Sybase (see “Starting Sybase Adaptive Server Enterprise” on page 35). With the database up and running, you can start ClearTrust SecureControl. To start all SecureControl servers at once, enter:

```
secctrl start
```

To start the ClearTrust SecureControl servers individually, use the `secctrl_server` utility (provided in your ClearTrust SecureControl installation) with the name of the server you wish to start as the command line argument. Starting the servers one-by-one can help you isolate problems. See “Troubleshooting” on page 195 for information about error messages and possible workarounds.

To start each server individually, you must do so in this sequence:

1. Start the Entitlements Server by entering:
   ```
   secctrl_server dataserver
   ```
2. Start the Dispatcher by entering:
   ```
   secctrl_server dispatcher
   ```
3. Start the Authorization Servers:
   ```
   secctrl_server authorizer port-number
   ```
   where *port-number* is the Authorization Server listener port

To stop ClearTrust SecureControl server processes:

1. Enter `secctrl stop`
   You can also stop the individual server processes.

Installing ClearTrust SecureControl on Windows NT

The ClearTrust SecureControl Windows NT installation process uses the InstallShield wizard to guide you through a series of dialog boxes. You'll be prompted for much of the information necessary to get the core SecureControl services installed and running. The entire process usually takes less than an hour.

Before You Begin

Verify that your system meets the minimum requirements listed in Table 2-1. Before you begin the installation, make sure you have the following items and information on hand:

- Your ClearTrust SecureControl installation CD.
- The name of an SMTP (email) host on your network and the email address of the person or entity primarily responsible for your ClearTrust SecureControl server. (You don't need these unless you plan to configure automatic notification.)
- The License Info and License Key provided to you by Securant.
- An established Administrator account on the target machine.

Installing ClearTrust SecureControl Servers

The installation process uses the InstallShield program to create the sub-directories (in the `/Securant` directory, by default) and install all components of the system, including:

- Oracle Database Server (/Orant)
- ClearTrust SecureControl API (/SecCtrl/API) for developers to use in creating custom applications
- Policy Administration Guide, Developer's Guide, and Installation and Configuration Guide (/SecCtrl/docs)
- Installation programs (/SecCtrl/installs) for ClearTrust SecureControl Manager, LDAP Replicator, SecureDetector
- Sample data to load into the system and a tutorial (/SecCtrl/Tutorial) to help you learn about using ClearTrust SecureControl to secure resources

To run the ClearTrust SecureControl Windows NT installer:

1. Log on to Windows NT as an Administrator of the machine.

2 Exit any currently running programs.

3 Insert the ClearTrust SecureControl CD-ROM into the drive. The installer program will launch automatically. From the installer program, you can choose which components you want to install. The ClearTrust SecureControl Server is selected by default.

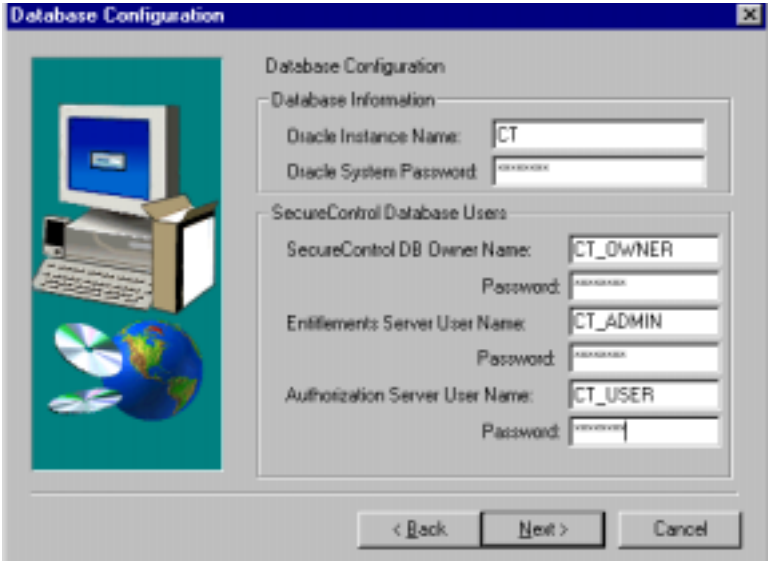
4 Click Install to begin the installation. A verification dialog prompts you to verify that to install the ClearTrust SecureControl server.

5 Click OK to start the InstallShield installer program. The software license agreement displays; if you agree to the terms, click Yes to continue with the installation. You'll be prompted to select a directory for the ClearTrust SecureControl installation.

6 Edit the directory location if you like, and then click Next to continue. A dialog displays prompting you to choose the type of installation—Typical, Compact, or Custom.

7 Make sure the radio button for Typical is selected and click Next to continue. *Typical* installs all the ClearTrust SecureControl server components you need, including product guides (`..\SecCtrl\docs\`) and a tutorial (`..\SecCtrl\Tutotial\index.html`) that will step you through the process of configuring access policies using the SecureControl solution.

8 Click Next to continue with the installation. The Database Configuration dialog displays:



9 Create passwords for the Oracle database and for the SecureControl Entitlements Server, Authorization Server, and SecureControl database owner user accounts. (When creating any password, use a combination of upper- and lower-case letters; include at least one digit or punctuation mark; and avoid dictionary words or real names.)

10 Click Next to proceed with the installation.

11 Enter your mail server name and email address. You may omit these if you do not want ClearTrust SecureControl to send email notifications of significant problems.

12 Click Next to proceed with the installation.

13 Enter the License Info and License Key you obtained from Securant. (Enter these strings exactly as provided, or the SecureControl server won't run. You can copy and paste the information from the file in which they are contained using a text editor, such as Notepad, to ensure accuracy.)

14 Click Yes to restart your computer. After your computer restarts, ClearTrust SecureControl performs some initial configuration. This process may take several minutes.

This installation places the installation programs for all of the other ClearTrust SecureControl components in a directory named Installs.

Starting the ClearTrust SecureControl Servers on Windows NT

You can start all ClearTrust SecureControl Servers at once by selecting the Start all Services menu selection. You can verify that all services have started by check the Services control panel. The four ClearTrust SecureControl services are prefixed "JavaService," that is, JavaService AUTH00, JavaService AUTH01, JavaService DISPATCH, and JavaService ESERVER.

To start all ClearTrust SecureControl servers and the Oracle database:

1 From the Windows NT Start menu, select Programs.

2 Select Securant

3 Select ClearTrust SecureControl

4 Select Start all Services.

5 Verify that all the services have started by checking the Services control panel.

If the services haven't all started, you can start them manually. (Starting the servers one-by-one can help you isolate problems. See "Troubleshooting" on page 195 for information about error messages and startup problems.)

To start the servers manually, you must first start the Oracle services by open the Services control panel:

- Start the OracleStartORCL service
- Start the OracleTNSListener service.

At this point the database should start, and you can now start the ClearTrust SecureControl servers. You must start the servers in this specific sequence:

- Entitlements Server
- Dispatcher
- Authorization Servers

You can start the servers as Windows NT services or as applications.

- Run the **startall.bat** batch file which can be found in the bin subdirectory of the SecureControl home directory.

You can also start all services at the Windows NT command prompt by running the `startall.bat` batch file, located in the `Securant\SecCtrl\bin` sub-directory.

To run the servers as Windows NT services, open the Services control panel and start each of the services in order:

1. JavaServiceESERVER service
2. JavaServiceDISPATCH service
3. JavaServiceAUTH00 service
4. JavaServiceAUTH01 service.

To start the servers as applications, run the **eserver.bat**, **dispatch.bat**, **auth00.bat**, and **auth01.bat** batch files found in the bin directory of your ClearTrust SecureControl installation.

Setting up ClearTrust SecureControl to run as services (rather than applications) can enable you to automate the restart process, just you can with any Windows NT service.

Next Steps

If you've completed the steps in this chapter, ClearTrust SecureControl Servers are now installed on Solaris or Windows NT. To start the system, you must first start the underlying database, and then you can start the ClearTrust SecureControl Servers. At this point, you can:

- Install sample data and step through the ClearTrust SecureContorl Tutorial to begin learning how to use the system from a security administrator's perspective. You'll find everything you need to get started, including a script to load sample data, in the `Securant/SecCtrl/Tutorial` subdirectory.
- Install and configure Web Server Plug-ins on any Web servers that you want to secure, as detailed in Chapter 3.
- Install and configure the ClearTrust SecureControl Manager software and begin setting security policy for your environment, as detailed in Chapter 7.

Finally, be aware that most of the selections you made during installation are located in the default.conf file, and that you can change parameter settings if necessary (port numbers, names, and the like) by manually editing the file in a text editor. See "ClearTrust SecureControl Default.conf Parameter Settings" on page 137 for listings of all parameters.

Chapter 3
ClearTrust Web Server Plug-ins

To protect your Web servers using ClearTrust SecureControl, you must install and configure the appropriate ClearTrust SecureControl Web Server Plug-in on the Web Server. Securant Technologies provides ClearTrust Web Server Plug-ins for each of the leading web servers—Apache, Microsoft IIS, and Netscape—that conform to the architecture of the Web servers into which they install; that is, there's a *module* for Apache; a *filter* for Microsoft IIS; and *plug-ins* for Netscape. You'll find them all referred to simply as "Plug-ins" or "ClearTrust Web server Plug-ins" throughout this guide.

Essentially, these Plug-ins replace or augment the Web server's native security mechanisms. A ClearTrust SecureControl Web Server Plug-in runs in the same process as the Web Server itself, and is invoked whenever the Web Server needs to determine access rights for a particular URI (uniform resource identifier). The ClearTrust Web Server Plug-in forwards access requests to a ClearTrust SecureControl Authorization Server (installed in Chapter 2) then passes the answers it receives back to the Web Server.

In addition, ClearTrust Web Server Plug-ins can be configured to implement Single Sign-on, so that after users authenticate with one Web Server, they don't have to go through subsequent authentication routines—it's handled for them behind the scenes, using an encrypted session cookie generated by the Authorization Server.

You must install and configure the appropriate Plug-in for any Web server that you wish to secure using ClearTrust SecureControl services. This chapter tells you how. It includes the following sections:

- ClearTrust Web Server Plug-in Installation Overview

- Understanding the Configuration Options
- ClearTrust Web Server Plug-ins for Solaris
- ClearTrust Web Server Plug-ins for Windows NT
- Configuring the ClearTrust SecureControl Plug-in
- Name of Authentication Types for Configuration

ClearTrust Web Server Plug-in Installation Overview

The installation process can vary greatly depending on your environment; for example, you may be installing several Web Server Plug-ins on several different Web servers, some on Windows NT based Web servers, some on Solaris Web servers.

In general, here are the steps for installing a Clear-Trust Web Server Plug-in:

1. Generate the secret by using the Keygen utility at the ClearTrust server, and have it available as you begin to install the Web Server Plug-in. See “Using the Keygen Utility to Generate a Shared Secret” on page 58 for details.
2. Install the Web Server Plug-in at the Web server. For Windows NT-based Web Servers, the installation is a standard InstallShield routine that provides many configuration parameters by default. See “ClearTrust Web Server Plug-ins for Windows NT” on page 66.

   For Solaris-based Web Servers, installation is typically a manual process. See “ClearTrust Web Server Plug-ins for Solaris” for details.
3. Configure the Web server for single sign-on by copying the secret (created in Step 1) to the Web server.
4. Configure the Web Server Plug-in for the type of authentication and granularity of security that you want by editing the Plug-in’s Default.conf file (located in the `Securant\ct_root\plugins\`*`web-server-type`* directory. The file contains numerous parameters, but the only ones you usually need to edit are:

   —WebServer name, which must match the name of the Web server you secure using the ClearTrust Manager

   —URIs of the resources you want to secure

   —type of authentication mechanism to use for each resource

   —additional configuration settings such as authorization server mode (standard or distributed) and SSL (secure sockets layer)

See “Understanding the Configuration Options” on page 51 for more information these important configuration parameters before you begin.

5 Verify your configuration settings during Web Server restart by watching the Solaris standard output or the display in the DBWin32 window (a Windows debugging tool). Before customizing or changing your configuration, you should ensure that SSO is running properly.

After installing the Plug-in, you can return at any time and modify your configuration settings. You can also extensively customize the experience for your users by modifying (or creating from scratch) the Web forms, CGIs, and Servlets that comprise the entry points (portal) to your organization’s Web application environment. (See Chapter 4 for details).

TABLE 3-1: ClearTrust Web Server Plug-in Requirements Summary

	Sun Solaris		Microsoft Windows NT	
	Apache	**Netscape**	**Microsoft IIS**	**Netscape**
Supported releases	Apache 1.3	Netscape 2.x, 3.0, 3.5, 3.6	IIS 4.0	Netscape 2.x, 3.0, 3.5, 3.6
Patches required	http://bugs.apache.org/index.cgi/full/6055		Service Pack 5	Service Pack 5
Operating system	**Solaris 2.5.1**	**Solaris 2.5.1**	Windows NT 4	Windows NT 4
OS Patches	**Patch cluster 2.6**	**Patch cluster 2.6**	Service Pack 5	Service Pack 5
Hardware platform	Sun Ultra (or faster)	Sun Ultra (or faster)	Pentium II (300-Mhz or faster)	Pentium II (300-Mhz or faster)
Storage	500-Mb for software install	500-Mb for software install only	300-Mb for software install only	300-Mb for software install only

Understanding the Configuration Options

Whether the configuration is manual, as with the Solaris Web Server Plug-ins, or wizard-driven, as with the Windows NT Web Server Plug-ins, you should have a basic understanding of some of the key parameters that affect software components comprising the ClearTrust SecureControl solution before you begin.

For example, the ports through which the various components on the ClearTrust SecureControl servers communicate with the Web Servers are set in the configuration file of both ClearTrust SecureControl server and Web Server Plug-in; port numbers in both configuration files must match

for the system to work. Table 3-2 details some of the important settings in the Web Server Plug-in's configuration file. All of the settings can be modified at any time; see "Web Server Plug-ins Reference" on page 145 for a complete parameter reference.

TABLE 3-2: Web Server Plug-in Configuration Parameters

Parameter	Description	Cross-reference in default.com
Server dispatcher host	DNS name of the machine that runs the ClearTrust SecureControl Server's dispatcher.	securecontrol.plugin.dispatch_host
Server dispatcher port	Port number of the ClearTrust SecureControl Server's dispatcher.	securecontrol.plugin.dispatch_port
Server dispatcher time-out	Number of seconds the Plug-in will wait from replies from the Server Dispatcher.	Default value is 10. Maps to securecontrol.plugin.dispatch_timeout
Key server port	Port number of the Key Server.	securecontrol.plugin.dispatcher_key_port
Authorization server timeout	Seconds the Plug-in waits for replies from ClearTrust SecureControl Authorization Servers	Default value is 5. Maps to seurecontrol.plugin.auth_server_timeout
Authorization server mode	Determines how Plug-in distributes requests for authorization	Default value is "standard." Maps to securecontrol.plugin.auth_server_mode
Web server name	Name that identifies the Web Server to the ClearTrust SecureControl environment	Default is "WebServer." Change as you see fit. Must match the name you use for the Web Server in the SecureControl Manager application. Maps to securecontrol.plugin.web_server_name
Realm name	Identifies the HTTP authentication realm (not the ClearTrust realm)	securecontrol.plugin.web_server_name
Cookie domain	Specifies scope of ClearTrust SecureControl's SSO	Must be at *company.com* or *organization.org* (or *network.net..)* level or lower. securecontrol.plugin.cookie_domain
Key client secret	Authenticates the Web Server to the Key Server to enable single sign-on.	Stored in the keyclient.sec file on the Web Server. Entered in dialog box in Windows NT Web Server Plug-in installation. Entered manually in Solaris installations.
SSL	Selects secured sockets layer (SSL) encryption.	Must match the setting for securecontrol.net.ssl.use=[yes or no]. The Web Server (Apache, IIS, Netscape) must also be configured with SSL enabled.

The table lists just some of the settings you'll need to consider as you install or configure the Web Server Plug-in. Gather this information before you get started if you don't already know it. Some of the parameters that effect runtime operations and successful implementation are discussed in more detail in this section.

Support for Multiple Authentication Types

User authentication is a fundamental security process in any virtual enterprise network. Granting or denying access to any resource should be based on the validated identity of the User; authentication is the process of validating user identity based on some sort of proof, or credential. Because ClearTrust SecureControl is built on a "pluggable" authentication framework, organizations can implement the authentication mechanisms most suited to their needs, from off-the-shelf mechanisms to their own customized approaches.

ClearTrust SecureControl provides out-of-the box support for a wide range of authentication mechanisms including:

- Basic authentication (BASIC), an internal ClearTrust SecureControl UserID and password combination. For an extra measure of security, ClearTrust SecureControl uses a hashing algorithm on ClearTrust SecureControl passwords, so they are never exposed in cleartext either in transit (across the wire) or when stored in the ClearTrust SecureControl Entitlements Database. If no other authentication mechanisms are configured, this is the default for all ClearTrustSecureControl-protected resources.
- X.509 certificates (CERTIFICATES), supported by the Web Server software browser certificates for authentication. The certificate must be mapped to a User account in ClearTrust SecureControl.
- Entrust certificates (EDIRECT); in this mode, ClearTrust SecureControl implements Entrust's PKI infrastructure. (Cannot be used with SSL)
- RSA SecurID. Users must provide valid SecurID token information from the RSA ACE Server. A user account of the same name must exist in ClearTrust SecureControl.
- Windows NT logon. Users must have a valid Windows NT domain account. The same name must exist in ClearTrust SecureControl.

- LDAP[1] authentication; the LDAP user accounts must be mapped to ClearTrust SecureControl user accounts, either by creating a user account with the same name or using the LDAP Replicator tool to import the DN (distinguished name). You can enable SSL support between LDAP directory servers and Authorization servers.
- Certificate+ authentication lets you configure both a certificate and another authentication mechanism, such as a SecurID credential or a Windows NT domain logon. Certificate+ builds on the Web Server's built-in certificate support, so you must turn on SSL at the Web Server and configure the default.conf file of the Web Server Plug-in for SSL enabled. In addition, you must configure the Web Server to REQUIRE that clients submit certificates. However, in ClearTrust SecureControl, you secure the resource for the additional authentication mechanism only, for example, NT, LDAP, or Basic.
- Custom. You can use the ClearTrust SecureControl Plug-in API to create your own custom authentication routine as a Plug-in Extension (PIX). You can customize the login prompt and create your own error messages and logging. (You can also use create a PIX that integrates with existing legacy authentication mechanisms or any number of other integrations.) For more information, see the *ClearTrust SecureControl Developer's Guide*.

In all cases, the configured authentication mechanism prompts the User attempting to access a SecureControl-protected resource to provide the appropriate identification credentials.

For example, in the case of Windows NT, Basic, and LDAP, the credentials consist of a user name and password, while in the case of RSA SecurID, users will be prompted for the alpha-numeric string value that's automatically generate by the ACE Server. Users must provide the requested proof (or user credentials, that is, user name and password, user name and token, and so on). If the authentication mechanism accepts the credentials, ClearTrust SecureControl then checks for the User's authorizations on the requested resource.

1. Not only can you use LDAP directory services for authentication, you can replicate user account information from LDAP directories into your ClearTrust SecureControl entitlements database. You must map the user object schema from the LDAP directory using the ClearTrust SecureControl LDAP Replicator Tool, as detailed in "Integrating LDAP Directories and ClearTrust" on page 91.

Requirements for External Authentication Mechanisms

For ClearTrust SecureControl to check the credentials of a user against the Windows NT domain controller, an LDAP directory service, or an RSA ACE Server (for SecurID), the user account name in ClearTrust must match exactly the user account name in the external authentication mechanism. That means you must:

- Create new ClearTrust SecureControl accounts
- Synchronize the external accounts with ClearTrust SecureControl by importing these values. You can use the ClearTrust SecureControl API to write an application to do this; see the *ClearTrust SecureControl Developer's Guide* for more information.

Granular Support for Multiple Authentication Types

Granular resource-based authentication allows you to configure different authentication modes for different resources on any given ClearTrust SecureControl-enabled Web Server. ClearTrust SecureControl can support any or all of the authentication mechanisms listed above simultaneously, for different resources. That is, you can configure a global authentication mechanism that will be used for all resources, and you can at the same time specify different authentication mechanisms for select resources, at the URI level.

For example, you could require users to present a digital certificate to access specific financial resources, and you can allow users to access other applications with their Windows NT logon ID and password.

Two parameters in the `default.conf` file work together to ensure that all users always authenticate using the proper credentials for any given resource. These parameters are:

- `securecontrol.plugin.auth_resource_list`
- `securecontrol.plugin.default_auth_mode`

When you configure the Web Server Plug-in for your Web site, you can add as many specific URIs to the authentication resource list as you'd like to configure, and each one can have a different authentication mode attached to it. If you don't specify a particular resource, the default authentication mode (securecontrol.plugin.default_auth_mode) is used.

For example, the two parameter settings in the example below dictate that users trying to access anything under `/apps` would need to authenticate using a Windows NT logon, while users trying to access anything under `/financial` would need a certificate. All other resources would require the native ClearTrust SecureControl authentication.

```
securecontrol.plugin.default_auth_mode=BASIC
securecontrol.plugin.auth_resource_list=/apps/*=NT,
/financial/*=EDIRECT:CERTIFICATE
```

Authentication Types and Logging

Basic authentication is handled by the ClearTrust SecureControl Authentication Servers and is logged by the ClearTrust SecureControl system according to the logging preferences you selected during system configuration. All other forms of authentication are performed at the Web Server. Therefore, authentication failure (except for Basic) is not logged by the ClearTrust SecureControl system.

If you are performing authentication at the Web Server, be sure to provide your own logging functionality using the Plug-In extensions. See the *ClearTrust SecureControl Developer's Guide* for details.

Support for Form-based Authentication Prompt

When users attempt to access a ClearTrust SecureControl-protected resource, they are first prompted for authentication credentials by the Web browser's native authentication dialog box.

Alternatively, you can configure the ClearTrust Web Server Plug-in to use HTML forms. ClearTrust SecureControl includes several HTML forms (and the requisite CGIs) that you can use for Basic, LDAP, NT, and SecurID authentication. In addition, you can customize these forms to your own organization's identity and you can customize what your users see after login by directing them to a specific Web page. (Furthermore, you can personalize the pages for each individual User as detailed in "Customizing the Environment" on page 82.)

With either approach, the Web Server Plug-in processes the User ID and password information using the configured authentication mechanism, be it NT logon, LDAP, SecurID, or the Basic authentication built-in to ClearTrust SecureControl.

To select the forms-based approach, the Web Server Plug-in's default.conf file must have the `securecontrol.plugin.form_based_enabled` parameter set to "yes." See "Configuring the ClearTrust SecureControl Plug-in" on page 71 for further information about additional, related settings that you must make.

Support for Distributed Authorization Servers

The Authorization Server caches information about Users; Authorization Server is the component that the Web Server Plug-in contacts when it checks the authorization level of a given user. One of the parameters in the default.conf affects how the Web Server Plug-in will interact with Authorization Servers at run-time. Specifically, you can configure the Web Server Plug-in to function in *standard* mode or *distributed* mode.

- In standard mode, authentication requests go to a single Authorization Server as long as it's responsive; additional authorization servers will only be used when one fails.
- In distributed mode, authentication requests move in round-robin order among all know Authorization Servers.

Depending on your environment, standard mode may be best because it leverages the Authorization Server's caching mechanism. On the other hand, if you have a widely distributed environment, distributed mode may achieve overall better responsiveness. Nonetheless, this setting will have ramifications on how you configure other aspects of your ClearTrust SecureControl protected environment.

For example, if you set the parameter for distributed mode, you must have additional, stand-alone Authorization servers installed, configured, and running on your virtual enterprise network. For additional background information about these alternative modes and information about installing stand-alone Authorization servers, see "Redundant Authorization Servers" on page 111.

Support for Single Sign-on (SSO)

The ClearTrust SecureControl solution implements a single sign-on mechanism so that your Web users don't have to logon repeatedly during a working session as they access one resource after another. Rather, when

users initially authenticate to ClearTrust SecureControl, they are issued an encrypted session cookie that enables them to access all other resources to which they've been granted access.

As administrator, you can configure the length of time that a working session comprises, the type of encryption (3DES (default) or Blowfish) to use, and many other parameters. Encryption and session life in particular are two parameters that directly enhance the security of this approach. At runtime, when a user tries to access a ClearTrust protected Web site, the encrypted cookie is passed to the Plug-in.

The initial hand-shaking between Web Server and ClearTrust Authorization Server is accomplished by means of a shared secret mechanism. The Plug-in authenticates itself to the Key Server using an initial shared-secret; the Key Server generates subsequent secret keys on a regular basis and makes the new key available to the Plug-in.

To create the shared secret, you use the *keygen* utility, located in the `\Securant\SecCtrl\bin` directory of the ClearTrust SecureControl server installation (on Windows NT) or in the /scripts sub-directory on Solaris. Keygen creates secret keys for the Web Server Plug-ins and stores them in the KeyServer.sec file; the Key Server will read this file to initiate the first authentication.

Using the Keygen Utility to Generate a Shared Secret

To generate a secret key using Keygen, you must be at the server where your ClearTrust SecureControl installation is located.

Keygen is a simple command line application that takes the hostname of the Web server and the fully-qualified hostname (hostname.company.com, for instance) as input and then generates a teYou can copy the text string, including comments, into a text file and move the file to the Web server you're configuring so you can install the key on the Web server when instructed to do so later in this chapter. (Be sure to delete the text file when you're finished for security purposes.)

The two switches you'll use with Keygen are:

-a, to add another Web server to the KeyServe.sec list, and

-e, to extract the key data about a given Web server from the file.

On Solaris, the keygen utility is stored in the scripts sub-directory, which is typically located under the `/scripts` directory of the Securant ClearTrust installation (for example, `/opts/securant/ct_root/scripts`). You run the script, first adding your adding your Web Server and then extracting the secret key, as follows:

```
bash-2.01$ ./keygen -a eldar eldar.securant.com
success!
bash-2.01$ ./keygen -e eldar
eldar 5cYKs4pVRXVL8XcU9PNBuUblOA6rhW56sC4l07Ysm7hk
# The above line should be the first line in the client's
# secret file
# Any succeeding lines (including these) may be left in
# as comments.
```

```
On Windows NT, the keygen utility is located in the
\Securant\SecCtrl\bin directory. Here's an example Keygen run
at the Windows NT command prompt:
```

```
C:\Securant\SecCtrl\bin>keygen -a eldar eldar.securant.com
success!
C:\Securant\SecCtrl\bin>keygen -e eldar
dino bFQNpkSfm1RFRnspDzO0yyNSTlHBydCrWk9zqupa4jZk
# The above line should be the first line in the client's
# secret file
# Any succeeding lines (including these) may be left in
# as comments.
```

If you don't get the "success!" message, make sure you're not trying to generate a key for a Web Server that already exists in the KeyServer.sec (if you are, you'll see an appropriate error message), or that you're not entering an incorrect hostname or domain name—these names must be real in the context of your network. Double-check the hostname and the DNS name and try again. Or if the key already exists, you can simply extract it and install it on the Web server.

ClearTrust Web Server Plug-ins for Solaris

ClearTrust Web Server Plug-ins are available for Apache Web Server (1.3.x) and for Netscape Enterprise Server (2.x, 3.x). Basic requirements for the Web Server Plug-ins for the Solaris environment include:

- Sun Solaris 2.5.1 (or later), with recommended patches from Sun
- SPARC Ultra-class workstation

- 256-Megabytes RAM

The installation files for the Solaris Web Server Plug-ins are located on ClearTrust SecureControl CD. Installing the Plug-ins in a Solaris Web Server involves copying the relevant files to the appropriate directory on your Web server, making changes to the Web Server's configuration so that it engages the Plug-in code, and modifying the Plug-in configuration so that it protects the resources you want, using the authentication mechanisms you choose, as detailed in these instructions.

ClearTrust Web Server Plug-in for Apache

The Plug-in code for Apache is a single custom Apache module, `mod_ct_auth.c`, located in the /Plugins/Solaris/Apache/ct_auth/ directory on the ClearTrust SecureControl CD.

The process for creating an Apache Server that implements the ClearTrust Plug-in is the same as it is for implementing Apache modules in general:

- Modify the Apache configuration file so that it includes reference to the module (ct_auth) in the makefile
- Generate the makefile
- Run make and compile the code
- Modify the httpd.conf file
- Start the server

You can find complete instructions for creating Apache at http://www.apache.org/docs/install.html. These instructions are for a basic Apache Web server incorporating the ClearTrust SecureControl Plug-in only. The instructions presume that you already have an Apache Web server installed, configured, and running, and that the installation is in the `/opt/`*`apache_version`* directory.

Included with the ClearTrust SecureControl `ct_auth` module source code and libary files are samples of Apache's two chief configuration files that have been customized to include the necessary references for the `ct_auth` module. These files are:

- configuration.tmpl, a template that creates the makefile needed to build the Apache server

- httpd.conf, the configuration file that enables (or disables) specific modules and provides directives to Apache server at runtime

You can copy and paste the directives from these files into your existing Apache configuration files and re-compile your Web server, or, if you're starting from scratch, you can use these files as a starting point for a new Apache installation.

The httpd executable that you create will reside in the /src directory of the Web server, under the Apache installation.

System Requirements

You can obtain Apache Server 1.3.x from http://www.apache.org. To compile the source code, you'll also need gcc 2.95.1 (or later), and make 3.77 (or later). You can obtain these from ftp://ftp.gnu.org/gnu.

Basic system requirements include those listed in "ClearTrust Web Server Plug-ins for Solaris" on page 59 (Sun Solaris 2.5.6; Recommended Patches; Sun Ultra workstation with 256-Mb of RAM). In addition, note the following:

Apache Proxy Server To implement the ClearTrust Plug-in in an Apache Web Server configured for proxy or caching, you must also apply a patch to the proxy module to enable SecureControl to set cookies. You can obtain this patch from Apache at http://bugs.apache.org/index.cgi/full/6055.

Be sure to enable the proxy module first and make sure it's working correctly before you add the Plug-in.

Secure Apache Server (Implementing SSL). To implement SSL (secure sockets layer) on your Apache Web Server, make sure the server is configured and running correctly before you add the ClearTrust Plug-in.

See the Readme file for the ClearTrust Web Server Plug-in for Apache for additional details about setting up Apache as a Proxy Server and setting up Apache as a secure server, and see "Apache Web Server and SSL" on page 155 for additional information about incorporating SSL (secure sockets layer) into your Apache server.

Installing the Plug-in for Apache

Before editing any of the configuration and source code files you should make a backup copy of the originals and save them.

To install ClearTrust Web Server Plug-in for Apache:

1. Stop the Apache Web server:

   ```
   cd /usr/local/apache/bin/
   ./apachectl stop
   ```

2. Backup the existing executable by copying it to another location where it will be untouched (in case you decide to restore it later).

3. Make a directory for `ct_auth` in the `modules` sub-directory of your Apache installation and copy the contents of the ct_auth from the CD to this sub-directory:

   ```
   mkdir ct_auth
   cd /opt/apache_version/src/modules/ct_auth
   cp /cdrom/cdrom0/plugins/solaris/Apache/ct_auth/* /ct_auth
   ```

   Alternatively, you can also extract the files from the tar file to this sub-directory:

   ```
   tar xvf /cdrom/cdrom0/plugins/solaris/Apache/securecontrol-
   plugin-sol-ap-4.0.tar
   ```

4. Open the Configuration file (/opt/*apache_version*/src/Configuration) in a text editor and add the EXTRA_CFLAGS and AddModule directives for the ClearTrust module to your Apache configuration file. For example:

   ```
   EXTRA_CFLAGS=-DHTTP_ROOT\"/usr/local/apache\"
   AddModule modules/ct_auth/mod_ct_auth.o
   ```

5. Open `mod_ct_auth.c` in a text editor and modify the path of `sdiclient.a` and `libct45-apache13x-plugin.a` so that pathnames are correct for your system.

6. Open the `httpd.conf` file and add references for the SecureControl module as shown in the sample httpd.conf file. You'll find four sections within this file labeled as "ClearTrust SecureControl Web Server Plug-in Configurations." The first section identifies the location of the ClearTrust Plug-in software; you must insert this line in your Apache httpd.conf file and change it as needed to map to the correct directory for your installation:

   `CTPluginRoot /opt/`*`apache_version`*`/src/modules/ct_auth`

7. The second section of the httpd.conf file adds these lines to the httpd.conf. Copy these into the equivalent location in your httpd.conf file.

   ```
   AuthType Basic
   Require valid-user
   AuthName SCRealm
   ```

Sections three and four of the configurations are optional. Section three is for form-based authentication, and section four enables Apache to function as a proxy server. (See “Using Apache as a Proxy Server” on page 153 for details).

8 To implement custom forms on your Apache Web site, make sure your httpd.conf includes this line in the appropriate section of the file and includes the correct pathname for your installation:

```
Alias /securecontrol
/opt/apache_version/src/modules/ct_auth/securecontrol-docs
```

9 To use SecureControl SSO and the Apache mod_usertrack on the same Apache Web Server, set the CookieTracking directive to “off” in the httpd.conf. (If you’re not using mod_usertrack with the CookieTracking directive, you can skip this step.)

10 Replace httpd with the one you just built:

```
cp /opt/apache_version/src/httpd
```

11 Copy and paste the secret into the `KeyClient.sec` file (located in the KeyClient sub-directory of the Web Server Plug-in’s directory). See “Using the Keygen Utility to Generate a Shared Secret” on page 58 for details. Close the file when you’re done, and delete any extra text files containing this secret that you might have created.

12 Open the Default.conf file (located in the conf sub-directory of the Web Server Plug-in’s directory) and make any necessary edits for Web Server name, default authentication type, and resource list. See “Configuring the ClearTrust SecureControl Plug-in” on page 71 for additional information.
You can also edit this file to enable your own customized forms and administration forms. See “Customizations” on page 77 for additional information.

13 Re-start the Apache Server:

```
./apachectl start
```

The Apache Server should startup and initialize the Web Server Plug-in (ct_auth module). Any specific resources that you have listed in the default.conf will be protected.

ClearTrust Plug-ins for Netscape Enterprise Server

The ClearTrust Web Server Plug-in for Netscape’s Enterprise Web Server version 2.0.1, 3.0, 3.5.x, and 3.6.x is a single dynamically linked library file called `libct45-netscape36x-plugin.so`. To install the Plug-in, you must

make some simple changes to the configuration of the Netscape Enterprise Server itself. These instructions presume that you already have Netscape Enterprise Server installed, configured, and running on your Solaris system.

System Requirements

Solaris 2.5.1, plus patches

Sun SPARC Ultra workstation-class

Installing the Plug-in

Implementing the ClearTrust SecureControl Plug-in involves making manual changes to the obj.conf file. Any time you change the obj.conf file manually, you must shut down the Administration Server; make the changes to the file; and then restart the Administration Server later.

In addition, when you make changes through the Administration Server, the Administration Server removes comments from the obj.conf file. So if you make changes to the your Netscape Web Server *after* you've installed the ClearTrust SecureControl Web Server Plug-in, you may need to restore the Plug-in's configuration information in the obj.conf.

For example, if you install the Web Server Plug-in on a secure Netscape Enterprise Server—one that implements SSL—and you later decide to disable SSL, the Administration Server removes all the directives for the Plug-in, thereby disabling ClearTrust SecureControl as well.

As a precaution, you should always make a backup of the obj.conf file before you make manual edits of any kind.

To install the ClearTrust Web Server Plug-in on Netscape:

1. Shut-down the Netscape Enterprise Server Administration Server. Go to the Administration server (*host-name*:*admin-port-number*/), click on the Admin Preferences tab, and then click the "Shut down the administration server!" button.
2. Make a backup copy of the Netscape `obj.conf` for safekeeping by copying the obj.conf to a new filename, such as obj-conf.bak. Make a note of what you called the file and where it's located in case you want to restore it.
3. Copy the SecureControl Plug-in files from the ClearTrust CD to the Web server:

   ```
   cp -R /cdrom/cdrom0/plugins/solaris/Netscape/* install_root
   ```

where *install_root* is the directory you selected to install the SecureControl Plug-in (for example, /opt/securecontrol/secctrl/plugin/). The system does not default to a specific directory.

Alternatively, you can extract the tar file from the *install_root* location:

```
cd install_root
tar xvf/cdrom/cdrom0/plugins/solaris/Netscape/securecontrol-plugins-sol-nes-4.0.tar
```

The tar file contents will be extracted into the directory.

4. Using a text editor, open up the obj.conf file (located in the `/opt/securecontrol/secctrl/plugin/conf` directory). Directives to implement the ClearTrust SecureControl Web Server Plug-in have been embedded (in bold typeface, preceded by comments) in the Netscape obj.conf file. (Directives without comments are from the standard default version of `obj.conf`.)
5. Copy the ClearTrust SecureControl directives into your existing obj.conf or can either use the `obj.conf` file included in the ClearTrust SecureControl distribution (after making appropriate modifications for your local environment), or you can cut and paste the ClearTrust SecureControl directives into the existing customized `obj.conf` file. Either way, don't change the order of the directives: doing so can lead to incorrect behavior.
6. Open the KeyClient.sec file
7. Copy and paste the secret into the `KeyClient.sec` file (located in the KeyClient sub-directory of the Web Server Plug-in's directory). See "Using the Keygen Utility to Generate a Shared Secret" on page 58 for details. Close the file when you're done, and delete any extra text files containing this secret that you might have created.
8. Open the Default.conf file (located in the conf sub-directory of the Web Server Plug-in's directory) and make any necessary edits for Web Server name, default authentication type, and resource list. See "Configuring the ClearTrust SecureControl Plug-in" on page 71 for additional information.
   You can also edit this file to enable your own customized forms and administration forms. See "Customizations" on page 77 for additional information.
9. Start the Netscape Administration tool. When you restart the Netscape Enterprise Server, the Plug-in will be initialized as part of the Netscape Web Server configuration.

ClearTrust Web Server Plug-ins for Windows NT

Installing the Web Server Plug-ins on Windows NT is an InstallShield driven process. Before you begin, you should know the DNS name of the machine on which the ClearTrust SecureControl Server Dispatcher service is running and you should have the secret key available in a text file for entering into the appropriate dialog box.

The installation dialogs have many default values preconfigured; you can accept or change as needed. (Unless you have reason to change the default port setting and timeout parameters, you can leave them alone.) See “Understanding the Configuration Options” on page 51 for additional background information about the choices you need to make during installation.

Both Microsoft IIS and Netscape follow the same installation process, which typically takes less than an hour. You’ll see two primary installation dialog boxes during the process, starting with the Plugin Configuration dialog:

The dialog box displays default values for most of the fields, which you can likely accept. However, if you changed the port number on the ClearTrust SecureControl servers installation (the Entitlements server), make sure you change the number in this dialog to match.

The fields for which you must provide entries are the hostname, cookie domain, and Key Client Secret. See the Table for details about these settings.

TABLE 3-3: Plug-in Configuration Dialog Options

Parameter	Description	Usage note
Server dispatcher host	Enter the fully-qualified DNS name of the machine that runs the ClearTrust SecureControl Server's dispatcher.	*Hostname.domain-name.domain* For example, hq001.securant.com
Authorization server mode	Default is "standard."	Change to "distributed" only if you'll be configuring additional stand-alone Authorization Servers, as detailed in Chapter 6.
Web server name	The name of the Web Server whose resources you want to secure. Change to something more meaningful for your enterprise if you wish.	Default is "Web Server." The name must match the name you use to identify Web Servers that you secure in ClearTrust, using the Manager application.
Realm name	Used by the Web Server for session cookie creation and distribution. (Has nothing to do with ClearTrust SecureControl Realms, which are user groupings for authorization purposes.)	Default is CTRealm. This is HTTP authentication realm for cookie-passing purposes. All Web Servers that will be involved in SSO should have the same realm name.
Cookie domain	Defaults to the domain name configured in the machine's TCP/IP configuration.	Cannot use .com, .org, or .net (and so on) by itself. Must use a minimum of organizational unit name (company.com, organization.org, network.net) or below.
Key client secret	Authenticates the Web Server to the Key Server to enable single sign-on.	Entered in dialog box in Windows NT Web Server Plug-in installation. Entered manually in Solaris installations.
SSL	Selects secured sockets layer (SSL) encryption.	Must be configured in the Web Server configuration as well.

CGI Configuration

Depending on the version of ClearTrust Web Server Plug-in you're installing, the CGI Configuration dialog box may not display. The parameters set by this dialog are used for Administrative forms and CGIs that are included with the Plug-ins, which provide some basic browser-based administration capabilities. You can also customize the forms. See "Configuring the Administrative CGIs" on page 79 for more information.

Unless you plan to use the Administrative CGIs, you can accept the defaults without making other entries in this dialog box. If you decide later that you want to use the Administrative CGIs, you can always manually edit the `cgi.conf` file, which is where all the parameter values are contained.

TABLE 3-4: CGI Configuration Parameters

Parameter	Description	Usage note
Host	API Server hostname	Enter the name of the machine on which ClearTrust SecureControl servers have been installed.
Port	API Server listener port	Default value is 5601. Only if necessary, change to match the port number on the CleartTrust SecureControl servers (
Administrator	User name of the Administrator that the CGIs will use to log on to ClearTrust SecureControl	Default is “admin.” If you’ve already installed the ClearTrust Manager and deleted this account and
Admin Password	Administrative password for	Default is “admin.” If
Admin Group (VBU)	Administrative group that owns the CGIs	Default is “SecureControl Admin Group”
SSL	Select to enable SSL support of CGI.	You must configure all ClearTrust software components and the resources that are being secured (Web servers) for SSL or none at all.
Password length	Length for Aministrator’s password CGI	Default is 12.

ClearTrust Web Server Plug-in for Microsoft IIS

These instructions are for installing the ClearTrust Web Server Plug-in for Microsoft Internet Information Server (IIS). The instructions presume you have a Microsoft IIS Web Server installed on the target machine. See “ClearTrust Web Server Plug-ins for Windows NT” on page 66 for an overview of the installation process, including screenshots that display as you follow the steps below.

To install the ClearTrust Web Server Plug-in on Microsoft IIS:

1. Exit all currently-running programs.
2. Insert the ClearTrust CD into the CD-ROM drive
3. Launch the ClearTrust SecureControl installer by executing the installer program (`ct-iis400nt-plugin.exe` on the ClearTrust SecureControl CD in the `//installs/plugins/NT/IIS` directory) or by selecting ClearTrust SecureControl IIS Plug-in from the start-up dialog box that displays when you insert the CD into the drive.
4. Click the Install button to continue. In a few seconds, the installer displays a Welcome splash screen.
5. Click Next to continue past the Welcome screen.
6. Read the software license terms and if you agree to the terms, click Yes to continue with the installation.
7. Select a directory for your ClearTrust SecureControl server installation or accept the default (`C:\Securant\SecCtrl\IIS Plug-in`).
8. Click the Next button to continue. A dialog box prompts you to choose whether to set IIS Authorization settings at this point in the installation. ClearTrust Plug-in on Microsoft IIS requires Basic Authorization, so you should set it now.
9. Click the Yes button to have the installer change the settings for you. (You can use Microsoft Management Console to change this later if necessary; see “Microsoft Internet Information Server (IIS)” on page 157.) The Plug-in Configuration dialog displays.
10. The Plug-in Configuration dialog box displays most of the default settings for the Web Server and the ClearTrust SecureControl servers. You’ll need to change any of the defaults if necessary and enter the host name of the ClearTrust SecureControl server (installed in Chapter 2).
11. Enter the value of the secret (created as described in “Using the Keygen Utility to Generate a Shared Secret” on page 58) in the Plug-in Configuration dialog.

For descriptions of all parameters, see “Plug-in Configuration File Options” on page 145.

12 Click Next to continue. The installer displays a listing of all the settings you've chosen. You can go back and change any as needed, or you can accept them and manually change the configuration at a later time.

13 Click Next to continue. The CGI Configuration dialog box displays.

14 To make changes to the installation configuration, click Back and change the settings as desired; otherwise, click Next, and the installation will take place.

15 Continue with the instructions for “Configuring the ClearTrust SecureControl Plug-in” on page 71.

ClearTrust Web Server Plug-in for Netscape Enterprise Server

This section tells you how to install the ClearTrust Web Server Plug-in for Netscape Enterprise Server on Windows NT. These instructions presume you have a Netscape Enterprise Server installed on the target machine. See “ClearTrust Web Server Plug-ins for Windows NT” on page 66 for an overview of the installation process, including screenshots that display as you follow the steps below.

NOTE: Each version of the Netscape web server software has a corresponding version of the ClearTrust Web Server Plug-in. These instructions are for Netscape Enterprise Server 3.6. For Netscape 3.5 installations, after installation you must copy the associated dll and lib files for 3.5.x from the installation disk and overwrite the files supporting 3.6 that the installer places in your `C:\winnt\system32` directory.

To install ClearTrust Web Server Plug-in for Netscape:

1 Shut-down the Netscape Enterprise Server Administration Server. Go to the Administration server (*host-name*:*admin-port-number*/), click on the Admin Preferences tab, and then click the “Shut down the administration server!” button.

2 Exit all other currently running programs. You install the ClearTrust SecureControl Plug-in for Netscape on Windows NT by executing the installer program (`ct-nscpALLnt-plugin.exe`), available on the ClearTrust SecureControl CD in the `//installs/plugins/NT/Netscape` directory.

You can also execute this program by inserting the ClearTrust SecureControl CD into the CD-ROM drive, choosing the ClearTrust SecureControl Netscape Plug-in from the dialog box that displays, and clicking the Install button. The Welcome splash screen displays.

3. Click the Next button to continue. The software licensing agreement displays.
4. Read the software license terms and if you agree to the terms, click Yes to continue. The Setup Type dialog displays.
5. Click on the Typical radio button to select it and then click Next to continue.
6. Select a directory for your ClearTrust SecureControl server installation (the default path is C:\Securant\SecCtrl\Netscape Server Plug-in) and Click Next. You should now see a list of all the Netscape Web Servers on your machine. If no web server names display, the Web server hasn't been setup properly.
7. Select the Web Servers onto which you wish to install the ClearTrust SecureControl Plug-in and click Next. The Plugin Configuration window displays.
8. The Plug-in Configuration dialog box displays most of the default settings for the Web Server and the ClearTrust SecureControl servers. You'll need to change any of the defaults if necessary and enter the host name of the ClearTrust SecureControl server (installed in Chapter 2).
9. Enter the value of the secret (created as described in "Using the Keygen Utility to Generate a Shared Secret" on page 58) in the Plug-in Configuration dialog.

   For descriptions of all parameters, see "Plug-in Configuration File Options" on page 145.
10. Click Next to confirm your settings. The ClearTrust SecureControl Plug-in installer presents all the configuration parameters you entered; you can click Back to change any if necessary. You can also change parameter values after installation is complete by manually editing the default.conf file.
11. Click Next to continue. The CGI Configuration dialog box displays.
12. To make changes to the installation configuration, click Back and change the settings as desired; otherwise, click Next, and the installation will take place.
13. Continue with the instructions for "Configuring the ClearTrust SecureControl Plug-in" in the next section.

    The installation of the ClearTrust SecureControl Plug-in is now complete; the Plug-in's configuration file will contain the parameter values you entered during this process.

Configuring the ClearTrust SecureControl Plug-in

After installing the Web Server Plug-in, you must make a few changes to the Plug-in's configuration file and then re-start the Web Server in order to initialize the plug-in code with the Web Server. Specifically, you must:

- Identify the resources that you want to protect

- Specify the types of authentication mechanisms, both internal (ClearTrust) and external (third-party) that you want protected resources to use
- Configure locations for forms and CGIs for form-based authentication prompts

This section tells you how.

To configure the ClearTrust Web Server Plug-in:

1. Use a text editor to open the default.conf file. You'll find this located in the /conf sub-directory of the Web Server Plug-in. For example:

   `/Securant/SecCtrl/`*`web-server-name`*` Plugin/conf/default.conf`

2. Find the `securecontrol.plugin.auth_resource_list` and enter the configuration for each URI equal to the type of authentication you want to use.
3. Use commas to separate resources, use colons to separate a list of multiple authentication types for a single resource, in the pattern *URI=Auth-type-1:Auth-type-2, URI2=Auth-type-1*

   As an example:

   ```
   securecontrol.plugin.auth_resource_list=/*=NT:BASIC,CERTIFIC
   ATE, SECURID, /apps/*=CERTIFICATE, /financial/*=SECURID
   ```

   This configuration specifies that Users attempting to access resources under `/financial` must authenticate using a SecurID token; that Users attempting to access resources under `/apps/*` must present a certificate; and that Users attempting to access any other resources under the root directory must present a Windows NT domain logon account and password or a ClearTrust SecureControl User account and password.

   If the User hasn't already presented with another accepted authentication type in the list, he or she is prompted for the first authentication type in the list. See Table 3-5 for additional information about authentication types.

NOTE: Be sure to map the appropriate User account or certificate into the ClearTrust SecureControl system as needed to support the external authentication mechanism. For example, if you NT as an authentication type, you must create User accounts in ClearTrust SecureControl that matches names in the Windows NT domain controller. (Ask Securant Professional Services for scripts and applications that can automate this process.)

4. Secure the resources you listed in the default.conf file (step 3) using the ClearTrust SecureControl Manager. Be sure the name of the Web Server you create in ClearTrust Manager is the same as the name entered in the default.conf file.

5 By default, the ClearTrust Web Server Plug-in uses the form-based authentication prompts (the HTML pages and CGIs) that ship with the product. This feature is enabled by this parameter setting in the default.conf file:

```
securecontrol.plugin.form_based_enabled=YES
```

You can use the Basic Authentication prompt that's native to the Web browser by changing the Yes to a No. If you do, Users will see this type of prompt:

6 If you leave the form-based authentication prompts enabled and you want to display a particular page to users when they successfully log in (other than the default pages provided), you can provide the appropriate path to this parameter:

```
securecontrol.plugin.login_home_location=/index.html
```

In this example, the index page for the Web server's root directory will display upon successful authentication. (For additional customizations, including information about personalizing the page for users based upon preferences stored in ClearTrust, see Chapter 4.)

With these configuration details completed, you're ready to startup the Web server; see "Verifying Configuration Parameters" for the concluding steps of Web Server Plug-in setup.

TABLE 3-5: Name of Authentication Types for Configuration

Options	Description	Usage Note
BASIC	ClearTrust SecureControl UserID and password combination.	Authentication events logged to ClearTrust SecureControl.
CERTIFICATE	X.509 digital certificates	User account name in ClearTrust SecureControl must match user name in LDAP directory.

Options	Description	Usage Note
CUSTOM	Custom authentication mechanism that you design and code, implementing the ClearTrust SecureControl Plug-in API	Must use ClearTrust Plug-in API to create a Plug-in Extension (PIX) that provides the custom functionality you want.
EDIRECT	Entrust Direct digital certificates	Do not use with SSL enabled.
LDAP	LDAPv3 compliant directory services	Must have an LDAP directory service available. User account name in ClearTrust SecureControl must match user name in LDAP directory.
NT	Windows NT domain controller based authentication	Windows NT environments only. User account name in ClearTrust SecureControl must match user name in NT domain controller.
SECURID	RSA SecurID authentication mechanism; uses RSA SecurID token card in conjunction with RSA ACE Server	Must configure RSA ACE Server. User account name in ClearTrust SecureControl must match user name in ACE Server.

Verifying Successful ClearTrust SecureControl Plug-in Configuration

Once you have installed and configured the ClearTrust SecureControl Plug-in, follow these steps to verify that the Plug-in is working correctly:

1. Start the ClearTrust SecureControl servers (the Entitlements, Authorization, and Dispatcher servers).
2. Start the Web Server and verify configuration parameters as described in the next section. See “Troubleshooting Single Sign-On Initialization” on page 176 for additional information if the if SSO doesn’t start successfully, or if you see other error messages.
3. Request a page on that server via a Web browser and perform final verification. See “Runtime Test” on page 76 for details.

Verifying Configuration Parameters

When the Web Server starts and the Plug-in initializes, informational messages are printed to the standard output. By reading the output, you can tell at a glance if the Web Server Plug-in was successfully initialized. If your Web Server is running on Solaris, you will see the output in the shell in which you started the Web Server. If your Web Server is running on

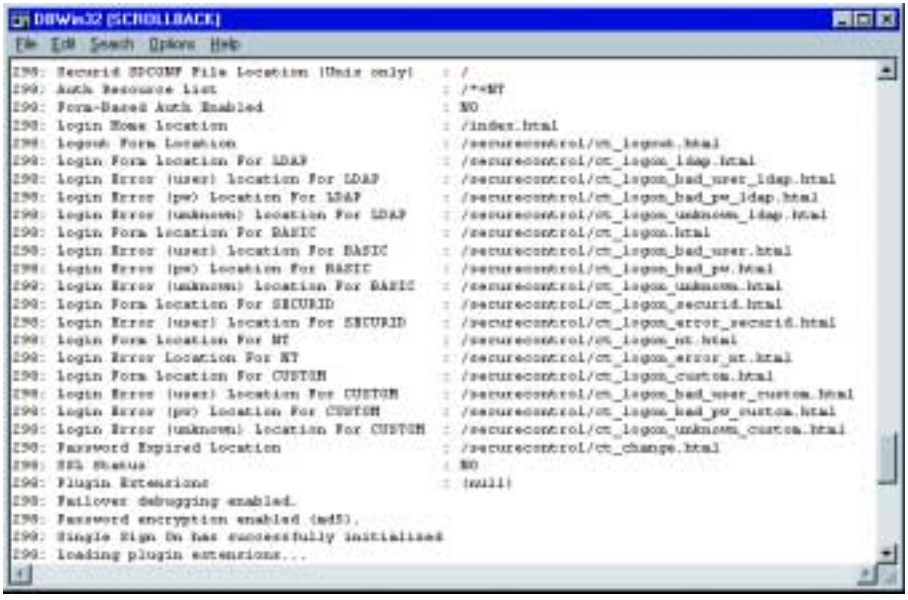
Windows NT, you need to start a separate program called DBWin32 to view the output. DBWin32 is available in the `util` subdirectory of the Plug-in directory in ClearTrust SecureControl distribution.

To verify the configuration parameters:

1. Start DBWin32 (or open a new shell).
2. Start the Web Server.
3. Read the output and verify that the configuration parameters have the values you expect. (If they don't, you can manually edit the Web Server Plug-in's default.conf file.)
4. Look for this message at the end of the display:

   `SSO successfully started`

   Here's a screenshot of a successful DBWin32 display:



If you don't see the `SSO successfully started` message, your Web Server Plug-in has not successfully initialized. See "Troubleshooting ClearTrust Web Server Plug-in Installations" on page 176 for additional information.

Runtime Test

Once the Web Server is running and you've confirmed that SSO is working, you can perform a runtime test.

To perform a runtime test:

1. Open your Web browser and enter the URL of a page from the Web Server. Depending upon where you are in the overall configuration of your ClearTrust SecureControl protected environment, you should see one of the following results:
   - The page displays. You can expect this result if your ClearTrust SecureControl servers are in active mode and the page is unprotected (meaning you haven't configured User accounts and secured the resources using the ClearTrust Manager application).
   - The page doesn't display, but a `permission denied` message does display. You can expect this result if your ClearTrust SecureControl servers are in passive mode and the page is unprotected.
   - A Username and Password logon prompt displays in your browser. You can expect this result if you've used ClearTrust Manager software to secure this resource.

   Verify that both protected pages and unprotected pages are working as you expect in the Web Server.

Next Steps

If you've completed the steps in this chapter, you should have a running Web server that incorporates the ClearTrust SecureControl Web Server Plug-in. If you haven't done so already, you should install the ClearTrust SecureControl Manager and configure the Web server resources, applications, and users. See "Installing ClearTrust SecureControl Manager" on page 122 for details.

Chapter 4
Customizations

This chapter provides information about various customizations and personalizations that you can implement in your ClearTrust SecureControl protected Web Servers. Many of these are provided as samples to get you started developing and creating your own customized and personalized forms and Web server applications.

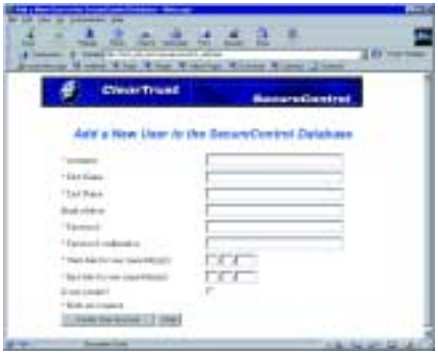
Web-based ClearTrust Administration

ClearTrust SecureControl includes several Web forms that you can use for several administrative tasks, including:

- Adding new users
- Changing passwords
- Resetting passwords

Add New User Form

The ct_add.html (and new-user.cgi) work together to let Administrators create new users the ClearTrust entitlements database:



Change Password Form

The ct_change.html and change-password.cgi let any user access change his or her password. You could add link to this form from your home page to give your Users convenient access, and you can customize the entire layout of the HTML form itself for your organization (that applies to all these HTML forms).

Reset Password Form

The ct_reset.html and reset-password.cgi let administrators generate new passwords automatically for users who have forgotten theirs. When you enter a valid User name, the new password is generated automatically (a random string of the length specified in the properties field) and displayed on a subsequent HTML page. You can give access to this form to local support staff, for example, to enable them to easily reset passwords for users.

Configuring the Administrative CGIs

The HTML pages and accompanying CGIs are installed in the same directory as the ClearTrust SecureControl Web Server Plug-in installation files, in two peer sub-directories:

```
/Securant/SecCtrl/web-server-name Plugin/securecontrol-cgis
/Securant/SecCtrl/web-server-name Plugin/securecontrol-docs
```

To use the default HTML forms as shipped, you have nothing to do except make sure the `cgi.conf` has the appropriate parameter settings for your environment and then enter the appropriate URL to access the forms on the Web server.

In addition, the `cgi.conf` file must exist in the location set by the `CT_CGI_PROPFILE` or the CGIs won't work. In Windows NT and in Unix environments, the `CT_CGI_PROPFILE` is set automatically during Web Server Plug-in installation; this environment variable is set to the path of the `cgi.conf` file, located by default in the same sub-directory as the Web Server Plug-in's `default.conf` file:

```
CT_CGI_PROPFILE:  C:/Securant/SecCtrl/Netscape Server
Plugin/conf/cgi.conf
```

During Windows NT Web Server Plug-in installation, you may be prompted for the CGI Configuration details, in which case the `cgi.conf` file will already contain any values you entered during installation. The Solaris Web Server Plug-in installation process requires you to manually edit the `default.conf` file for appropriate parameter settings (see Table 3-9).

NOTE: Implementing the administration Forms is completely optional.

TABLE 3-6: CGI Default.conf Parameters

Parameter	Description	Usage Note
newuser.server.name	Name of the API server to which the CGI will connect	
newuser.server.port	Port on the API server to which the CGI will connect	

Parameter	Description	Usage Note
newuser.server.use_ssl	Specifies whether or not the API Server will communicate with the CGIs via an SSL connection	Must map to the rest of your ClearTrust and Web Server environment; that is, if you configure one component for SSL, you must configure all components for SSL.. Check the default.conf files of the other components for their SSL settings.
newuser.admin.user	Administrator username that can access t	
newuser.admin.password	Password for Administrator	
newuser.admin.role	Administrative role under which you are connecting	
newuser.admin.group	Administrative group under which you are connecting	
reset.server.name	Name of the API server to which the CGI will connect	
reset.server.port	Port on the API server to which the CGI will connect	
reset.server.use_ssl	Specifies whether or not the API Server will communicate with the CGIs via an SSL connection.	
reset.admin.user	Administrative username you are using to connect	
reset.admin.password	Password associated with the administrative username you are using to connect	
reset.admin.role	Administrative role under which you are connecting.	
reset.admin.group	Administrative group that can execute the reset CGI	
reset.password.length	Password length (maximum number of characters allowed)	
change.server.name	Name of the API server to which the CGI will connect	

Parameter	Description	Usage Note
change.server.port	port on the API server to which the CGI will connect	
change.server.use_ssl	Specifies whether or not the API Server will communicate with the CGIs via an SSL connection	
change.password.length	password length limit (maximum number of characters allowed)	

To Modify the Administrative CGIs

You can modify or customize the HTML files included with the product for your own purposes. However, keep in mind that certain fields are required in order to work with the CGIs (which are compiled C code). Table 3-7 provides a reference of the form fields required by the ClearTrust SecureControl administrative CGIs. If you create your own forms for doing any of these tasks, be sure to include all the required fields in your HTML so that they will work with the corresponding CGIs.

TABLE 3-7: Web-based Administration CGIs and Required HTML Form Fields

		new-user.cgi	reset-password.cgi	change-password.cgi
HTML Field	**Description**	**ct_add.html**	**ct_reset.html**	**ct_change.html**
username	User name	required	required	required
first_name		required	na	na
last_name		required	na	na
email		required	na	na
old_password		na	required	na
new_password		na	required	na
confirm_new_password		na	required	na
password	user's password	required	na	na
confirm_password		required	na	na

HTML Field	Description	new-user.cgi ct_add.html	reset-password.cgi ct_reset.html	change-password.cgi ct_change.html
start_date	date on which the user account will be valid. This field is then expanded into three fields: start_date_year, start_date_month, and start_date_day	required	na	na
end_date	date on which the user account will be invalid. This field is then expanded into three fields: end_date_year, end_date_month, and end_date_day	required	na	na
is_private	set to y or n; administrators must belong to the same Role to view	optional	na	na

To modify or create from scratch new versions of any of the forms included with ClearTrust SecureControl:

1. Use an HTML or text editor to open the file and edit the HTML code, or create the your own HTML page. Refer to the table of required fields to make sure you include all the necessary ones.
2. Post the form to the appropriate CGI. For example, to implement the new user CGI in your HTML form, you'd include:

```
<form action="/securecontrol-cgis/new-user.cgi"
method="post">
```

Customizing the Environment

When Users attempt to access a SecureControl protected Resource, they'll see the default browser-based Basic authentication prompt. You can customize the Plug-in configuration many ways so that your users will see your own HTML prompts,

This section describes how to customize a ClearTrust SecureControl-enabled Web Server to return customized HTML pages based on end-user identity using the ClearTrust SecureControl API and the ClearTrust SecureControl Web Server Plug-in API. It contains the following sections:

- Creating Personalized Content
- Providing a Dynamic Menu of Applications

Creating Personalized Content

Every virtual enterprise network has different needs. With ClearTrust SecureControl, you can easily customize ClearTrust SecureControl's Java modules to create dynamically-generated Web pages that contain content or a menu of applications that a User is entitled to access, based on the user's identity.

Users names are held in the REMOTE_USER environment variable, in the HTTP header.

You can pass this variable to a CGI application

When a User successfully accesses the secured Web site they are challenged to authenticate.

HTTP header keeps track of the

Their ClearTrust SecureControl Username is set in the REMOTE_USER environment variable on the HTTP header.

A "personalization" application is an application, such as a CGI program or a Servlet, that takes the value from the HTTP header

then uses the Username to access the ClearTrust SecureControl API.

Through the ClearTrust SecureControl API, the personalization Application reads the Web User's preferences, such as preferred color scheme or language, and passes those preferences onto another Application that dynamically generates Web pages.

See the ClearTrust SecureControl Developer's Guide for more information about developing applications using the ClearTrust SecureControl API.

Providing a Dynamic Menu of Applications

When accessing the secured Web site, the User is challenged to authenticate. The ClearTrust SecureControl Username is set in the REMOTE_USER environmental variable on the HTTP header. A Personalized Menu CGI Application then checks the Username for access privileges against the Application List Configuration File and returns a list of the Applications to which the User is entitled. The list is then presented as a Web page with links to the appropriate Applications. This page can be customized to contain a welcome greeting to the User.

Since the entry URL is part of a defined ClearTrust SecureControl Application, privileges can be defined using both Basic Entitlements or Smart Rules.

Configuring Personalization with CGIs and Java Servlets

Personalization Modules provide sample code of how a user can use the SecureControl APIs to provide personalized content for Web application users. Personalization Modules come in two forms:

- CGIs written in C using SecureControl's C API
- Servlets written in Java using SecureControl's Java API

Both of these forms offer the same functionality but are implemented differently. You need to configure the Web Server and modify the `Default.conf` file. Instructions are found on the following pages.

Configuring the Web Server for Personalization

To use the Personalization Modules, you must first configure the Web Server. Before configuring the Web Server, note that the following directories contain the Personalization Modules:

/personalization-docs: directory where the Personalization HTML files and the SHTML files are located.

/personalization-cgi: directory where the Personalization CGIs are located.

/personalization-img: directory where the Personalization Images are located.

/personalization-servlets: directory where the Personalization Servlets are located.

1 Add the following to the CGI directory:

Prefix:/personalization/cgi-bin

CGI Directory: D:/Securant/SecCtrl/Netscape Server Plugin/personalization-cgi

2 Add to additional document directories:

Prefix:/personalization

Directory: D:/Securant/SecCtrl/Netscape Server Plugin/personalization-docs

Prefix:/img

Directory: D:/Securant/SecCtrl/Netscape Server Plugin/personalization-img

Personalization Module Subscription Applications

Personalization Modules have special Applications called Subscription Applications. Subscription Applications allow users to selectively subscribe and unsubscribe from Applications. Applications qualify as Subscription Applications only if:

- The Version field of a Subscription Application is prefixed with the string, “sub”
- The Description field of a Subscription Application contains the main URL link of the Application.
- A Boolean User Property has been defined using the same name as the Subscription Application’s name.
- A Smart Rule maps the Subscription User Property to the corresponding Subscription. This Smart Rule looks like this: **ALLOW (SubscriptionProperty) IS TRUE**.
- Within the PersonalizationServelet.java and the PersonalizationPropertyConstants.h, the Subscription UserPropertyNames array must include the name of the Subscription Application.

The default Subscription Applications are: Internet 50 News, Real Estate Today, Community Lending, Interest Rate Watch, and Small Business News.

Using the SecureControl CGI-Forms

The SecureControl Plug-in installation includes several CGIs that can be customized to meet your organization's individual needs. To configure the SecureControl CGI-Forms, add these URIs to the SecureControl access control system using the ClearTrust SecureControl Admin. Make sure to set access permissions appropriately. Only those with appropriate administrative access should be able to access these CGI-forms. The specific details of access control for CGI-forms will depend on your particular security setup.

Configuring the SecureControl Plug-in for Personalization

To use the Personalization Module with CGIs, you must modify the Web Server Plug-in's `Default.conf` file:

TABLE 3-8: Web Server Plug-in Default.conf Parameters Relevant for Personalization

Parameter	Description	Usage Note
securecontrol.plugin.login_home_location	/personalization/cgi-bin/PersonalizationAccess.cgi	For Java Servlets, set to: /personalization/personalized_access.shtml
securecontrol.plugin.login_form_location_basic	/personalization/login.html	
securecontrol.plugin.error_user_location_basic	/personalization/login_bad_user.html	
securecontrol.plugin.login_error_pw_location_basic	/personalization/login_bad_pw.html	
securecontrol.plugin.login_error_unknown_location_basic	/personalization/login_unknown.html	

Configuring Personalization with CGIs

The personalization CGI is grouped with the administrative CGIs in the `cgi.conf` file. The personalization CGI can help you customize the environment for users. To use the Personalization Module, you must modify the Web Server Plug-in's `default.conf` file to include settings for all these parameters (none are optional).

TABLE 3-9: CGI Default.conf Parameters

Parameter	Description	Usage Note
personalization.server.name	name of the API server to which the CGI will connect	
personalization.server.port	port on the API server to which the CGI will connect	
personalization.server.use_ssl	Specifies whether or not the API Server will communicate with the CGIs via an SSL connection	
personalization.admin.user	administrative username you are using to connect	
personalization.admin.password	password associated with the administrative username you are using to connect	
personalization.admin.role	administrative role under which you are connecting	

There are three basic Personalization Modules for CGIs. These include:

PERSONALIZATIONACCESS.C
The Personalization Access is a dynamically generated HTML page that lists and links to all the Applications and Subscriptions the User has access to. This is intended to be the main screen for the User accessing his Personalization Portal.

PERSONALIZATIONMODIFYPROFILEFORM.C
The Personalization Modify Profile Form is a dynamically generated HTML form that displays the User's current settings for the Personalization Portal. Through the HTML Form, the User can manually set attributes in their Profile. It works in conjunction with the Personalization Modify Profile program which takes the Form's settings and Modifies the User's settings.

PERSONALIZATIONMODIFYPROFILE.C
The Personalization Modify Profile CGI takes values set by the user in the Personalization Modify Profile Form, and then modifies the User based on those values. It then generates HTML to notify success or failure.

`ct.personalization_html.c` and `ct_personalization_html.h` contain common HTML output required by different components of the Personalization CGI.

PERSONALIZATIONCONSTANTS.H

The PersonalizationConstants.h file contains shared constants for the following CGI files:

MODIFY_FORM_ACTION: Location for the Form to specify the fields to Modify a User's Profile.

Default Location: `/Personalization/cgi-bin/PersonalizationModifyProfileForm.cgi`

MODIFY_ACTION: This points to the location of PersonalizationModifyProfile CGI.

Default Location: `/Personalization/cgi-bin/PersonalizationModifyProfile.cgi`

ACCESS_ACTION: This points to the location of PersonalizationAccessCGI.

Default Location: `/Personalization/cgi-bin/PersonalizationAccess.cgi`

IMAGE_DIR: The directory where your images are stored.

Default Location: `'/img'`

NOTE: This directory must be unprotected so the images can be displayed in the login html.

PERSONALIZATIONPROPERTYCONSTANTS.H

Contains the Names User Properties that the user has the ability to modify through the PersonalizationModifyProfileForm.

LOGIN: The key for the User's User Id

FIRSTNAME: The key for the User's First Name

LASTNAME: The key for the User's Last Name

EMAIL: The key for the User's Email

The following are constants corresponding to SecureControl UserProperty names that hold Personalization information about the user.

LANGUAGE: The key for the User's Language setting

USER_TITLE: The key for the User's Title setting

COLOR: The key for the User's Color setting

USER_ADDRESS: The key for the User's Address setting

PersonalizationAccess.c

PersonalizationModifyProfile.c

PersonalizationModifyProfileForm.c

To use the Personalization Servlets with Servlets, you must:

1. Identify the location of the personalization servlets in the ClearTrust SecureControl Web Server Plug-in's default.conf file by modifying these parameters:
2. Specify *include* in your Web Server's Servlet Engine classpath of the scapi.jar.file.
3. Add as a Java System property the location of the cgi.conf file using the java command-D option:

   DCT_CGI_PROPFILE="D:\Securant\SecCtrl\Netscape Server Plugin\conf\cgi.conf
4. Point your Servlet engine to retrieve the Servlets from the `\personalization-servlets` directory.

Files Associated with the Servlets

The following files are related to the Servlets:

- personalized_access.shtml
- personalized_modify_profile.shtml
- personalized_modify_profile_form.shtml

These three files provide the basic template of the static html, which serves as the backdrop for the Personalization Modules. Embedded within each shtml file is the <Servlet> tag which is processed by the Servlet Engine to the corresponding Servlet to generate the Dynamic HTML.

PersonalizationServlet.java is the base class of all Personalization Servlets. It also contains the constants, the ACTION constants, and UserProperty constants, which are similar to the Personalization CGI's constants specified in the PersonalizationPropertyContants.h and PersonalizationConstants.h.

PersonalizationAccess.java is the PersonalizationAccess Servlet responsible for generating the HTML to display all the Applications the User has access to.

PersonalizationModifyProfileForm.java the PersonalizationModifyProfileForm Servlet generates the HTML form to allow User's to input the parameters to modify their profile. The Servlet works in conjunction with the PersonalizationModifyProfile Servlet.

PersonalizationModifyProfile.java the PersonalizationModifyProfile Servlet is responsible for taking the values set by the User in the PersonalizationModifyProfileForm and Modifying the User based on those values. It then generates the HTML to notify success or failure.

PersonalizationSideBar.java the PersonalizationSideBar Servlet generates the HTML to display the image for a SideBar based on the User's current session.

ServerProxyPool.java the ServerProxyPool provides an Object Pool of pre-connected APIServerProxies. Making a connection to the Entitlements Server has associated overhead, so the ServerProxyPool recycles Connections after Servlets complete, thereby saving time and resources.

Chapter 5
Integrating LDAP Directories and ClearTrust

ClearTrust SecureControl solution enables enterprises to leverage the information stored in existing LDAP (lightweight directory access protocol) directories in several ways. For example, you can customize your Web Server Plug-in environment so that users authenticate to LDAP directories (see "Support for Multiple Authentication Types" on page 53).

In addition, you can populate ClearTrust Entitlements database with user account information from LDAP directories by using the SecureControl LDAP Replicator Tool. Essentially, the tool comprises two software components:

- SecureControl Replication Agent, a background software process that handles interaction with LDAP directory services
- SecureControl Replication Manager, the graphical user interface that administrators use to configure, schedule, and manage replication

You can configure regularly scheduled, on-going replication with a wide range of LDAP directory products. This chapter tells you how; it includes the following sections:

- Installing SecureControl LDAP Replicator Tool
- Configuring LDAP Directories for Replication
- Troubleshooting the LDAP Replicator Tool Installation

Installing SecureControl LDAP Replicator Tool

The components that make up the ClearTrust LDAP Replicator tool are Java applications (Java 1.2.2 runtime environment) that runs on Windows NT or Solaris. The installation process is handled by the Java-based

InstallAnywhere program, so the process is essentially the same, regardless of platform. Minimum requirements for both Solaris and Windows NT are listed in the table.

TABLE 3-10: LDAP Replicator Tool Requirements Summary

	Solaris	Windows NT
Operating system	Sun Solaris 2.5.1	Windows NT Server 4.0
Patches	Sun Solaris 2.5 Recommended Cluster	Service Pack 6
Hardware platform	Sun SPARC Ultra	Pentium II, 300-Mhz (or faster)
Storage (minimum)	1.2-Gbyte	10-Mbyte free hard-disk drive space
Peripherals	CD-ROM drive	CD-ROM drive
Memory	32-Mbyte RAM	32-Mbyte RAM
Supporting software	JRE 1.2.2	JRE 1.2.2

After ensuring that your platform meets the minimum requirements, you can install ClearTrust LDAP Replicator. The installation program (`LDAPInstall.bin` for Solaris; `LDAPInstall.exe` for Windows NT) is located on the CD in the `\installs` sub-directory, and it's also located in the `\installs\ldap` sub-directory of the machine on which you installed ClearTrust SecureControl servers. You can launch the installation program directly by invoking the filename.

To install ClearTrust SecureControl Replication Manager on Solaris:

Installing Directly from the Installation CD

If you have access to the ClearTrust SecureControl Installation CD-ROM:

- Make sure you have logged on as a user who is going to be using the **LDAP Tool**.
- From the installation directory, type:
- cd /cdrom/cdrom0
- cd LDAP
- ./LDAPinstall.bin

- This launches the InstallAnywhere installation program for the client which guides you through the installation process. Once the installation is complete, there will be client startup scripts in this directory.

Installing by Copying Files

To install the LDAP Replicator Tool on Solaris:

1. Copy the file `<CT_HOME>/SecCtrl/installs/ldap/solaris/ldapinstall.bin` from the ClearTrust SecureControl distribution into a temporary directory on the target machine.
2. Go to the temporary directory and type:

   `./LDAPInstall.bin`

   This launches the InstallAnywhere program which guides you through the installation process. The only parameter that you need to set during this process is the directory in which to install the **LDAP Tool**. The installation process sets up client startup scripts called **`LDAPMgr`** and **`LDAPAgent`** in this directory.

To start the LDAP Replicator Tool on Solaris:

Run the following commands in the directory where you installed the LDAP Replicator Tool. First, start the Agent, then start the Manager:

./LDAPAgent

./LDAPMgr

1. Copy the file `Securant/SecCtrl/installs/ldap/LDAPInstall.exe` from the ClearTrust distribution into a temporary directory on the target machine.
2. Go to the temporary directory and type

   `./Install_SecureControl_Mgr`

   The InstallAnywhere wizard loads and guides you through the installation process. A licensing message displays, and then an informational message displays.

   On Windows NT, the installation program is lauched automatically when you select “ClearTrust LDAP Replicator” from the ClearTrust SecureControl startup intallation menu.

To install the Replication Manager on Windows NT:

1. Insert the CD into the target machine's CD-ROM drive. This launches a dialog box that gives you a list of ClearTrust SecureControl components from which to choose.
2. Choose the ClearTrust SecureControl LDAP Replicator Tool and click Install. The InstallAnywhere program launches. The only information you need to provide during installation is the directory in which to install the LDAP Replicator Tool.
3. When the installation process completes, the LDAP sub-directory contains scripts in this directory. The LDAP Replication Agent and LDAP Replication Manager will be stored in a directory called LDAP.

   Before running the LDAP Replication Manager, you must start the Replication Agent.

4. From the Windows NT Startup menu, navigate to the Securant programs folder, to the LDAP Replicator menu.
5. Select SecureControl LDAP Replication Agent to start the agent process.
6. Select SecureControl LDAP Replication Manager to start the administration application.

NOTE: If the Replication Agent didn't start successfully, an Error message with that information will display when you attempt to start the Replication Manager program.

You can also start the agent from the Windows NT command prompt by entering its name—LDAPAgent.exe. Better still, you can install the Replication Agent as a Windows NT service, so it will startup automatically any time you re-boot your machine. (See the readme.txt file in the `LDAP\ntservice` sub-directory for details.)

Configuring the LDAP Replication Tool

The LDAP Replication Tool needs to be configured with four parameters. The LDAP Replication Agent needs a host name and port number, as well as a name for Voyager name. The fourth parameter is needed to register the LDAP Manager's Voyager service with SecureControl's internal security system, which utilizes JSSE. You can edit all of these files in the LDAP Replication Tool's `Default.conf` file.

TABLE 3-11: LDAP Replication Agent and Replication Manager Parameters

Parameter	Description	Usage note
securecontrol.ldap.agent.agent_host	Replication Agent	localhost
securecontrol.ldap.agent.agent_port	Replication Agent	5645
securecontrol.ldap.agent.agent_name	Replication Agent	Default is LDAPAgent
securecontrol.ldap.manager.voyager_port	Registers the LDAP Manager's Voyager service with SecureControl's internal security system	Default is 5646

Troubleshooting the LDAP Replicator Tool Installation

The ClearTrust SecureControl LDAP Replicator Tool writes information to a log file that is often useful in diagnosing problems. This log is located in the directory called `logs` in the directory where you installed the ClearTrust SecureControl LDAP Replicator Tool.

Configuring LDAP Directories for Replication

This section tell you how to configure specific directory services for Replication.

Configuring the Netscape Directory

To replicate from a Netscape Directory Server, the Server must be configured as a replication supplier. If the Netscape Directory Server from which you are going to replicate has not already been configured to be a replication supplier, you must perform the configuration:

1. Use the Netscape Admin Server to go to the Web Server's administration console.
2. Select the Directory Server that you want to use as the supplier.
3. From the Navigation bar, click Replication.
4. Click the Configure this Server link on the left side of the page.
5. In the *Supplier Server Replication Settings* section:
   - Select the directory in which you want the change log to be stored.
   - Enter `cn=changelog` for the changelog suffix.

- set the `max changelog records` and the `max changelog age`.

6 Click OK.

7 Click Apply in the header to apply your changes.

Configuring the PeerLogic-ICL i500 Directory

To replicate from a PeerLogic-ICL i500 Directory Server, you must make the changelogs format (in the `attributes.cfg` file) consistent with the LDAP Replicator Tool format. To modify the changelogs:

1 Shut down the PeerLogic-ICL i500 Directory Server.

2 Insert the following modifications to `attributes.cfg` in the correct numerical order in the i500ldap directory:

```
# Mods for changelogs
#

2.16.840.1.113730.3.1.5
NAME 'changeNumber'
DESC 'a number which uniquely identifies a change made to a directory
entry'
SYNTAX 'INTEGER'
EQUALITY integerMatch
ORDERING integerOrderingMatch
)

(2.16.840.1.113730.3.1.6
   NAME 'targetDN'
   DESC 'the DN of the entry which was modified'
   EQUALITY distinguishedNameMatch
   SYNTAX 'DN'
)

(2.16.840.1.113730.3.1.7
   NAME 'changeType'
   DESC 'the type of change made to an entry'
   EQUALITY caseIgnoreMatch
   SYNTAX 'DirectoryString'
)

( 2.16.840.1.113730.3.1.8
```

```
   NAME 'changes'
   DESC 'a set of changes to apply to an entry'
   SYNTAX 'OctetString'
)

(2.16.840.1.113730.3.1.9
  NAME 'newRDN'
  DESC 'the new RDN of an entry which is the target of a modrdn
operation'
  EQUALITY distinguishedNameMatch
  SYNTAX 'DN'
)

(2.16.840.1.113730.3.1.10
  NAME 'deleteOldRDN'
  DESC 'a flag which indicates if the old RDN should be retained as
an attribute of the entry'
  EQUALITY booleanMatch
  SYNTAX 'BOOLEAN'
)

(2.16.840.1.113730.3.1.11
  NAME 'newSuperior'
  DESC 'the new parent of an entry which is the target of a moddn
operation'
  EQUALITY distinguishedNameMatch
  SYNTAX 'DN'
)

# end changelog mods
```

3 Restart the PeerLogic-ICL i500 Directory Server. The changelogs are now readable by the LDAP Replicator Tool.

Using the ClearTrust SecureControl Replication Manager

This section tells you how to use ClearTrust SecureControl Replication Manager to configure replication, making replication with LDAP directory services an in integrated, automated feature of your ClearTrust

SecureControl protected environment. You'll use the Replication Manager to map LDAP user accounts to ClearTrust user accounts; to schedule replication tasks; or to import an entire LDAP directory.

Whenever you use the Replication Manager

NOTE: Be sure to start the LDAP Replication Agent before attempting to start the LDAP Replication Manager.

Creating New Replication Tasks

To replicate User data from an LDAP database into ClearTrust SecureControl, you must first define a replication task.

To define a replication task:

1. Make sure the ClearTrust SecureControl Replication Agent is running.
2. Launch the ClearTrust SecureControl Replication Manager by starting the application from the Windows NT Start menu or by executing the LDAPManager.exe file (or LDAPManager.bin for Solaris).

   The LDAP Replication Manager is where you perform all configuration and management of LDAP replication tasks. All of the buttons on this window are described below:

NOTE: All existing LDAP entries should be imported before any replication occurs to ensure correct behavior.

- **Schedule** - This button lets you schedule tasks for replication according to selected intervals specified in each task.
- **Import** - This button lets you import all of the existing users and groups in the LDAP database into the ClearTrust SecureControl database.
- **Unimport** - This button lets you delete all the users and groups from the ClearTrust SecureControl database that you previously imported.
- **Replicate Now** - This button lets you override scheduled replication time and will perform replication when you click it.
- **Reset Last Change Log #** - This button resets the change log to zero.
- **Task Name** - This field includes the name of the tasks you have created.
- **Next Run** - This field displays the scheduled time for the next replication.
- **Current State** - This field displays the current state of the selected task.

- **Last Change Log #** - This field displays the number of the last change log entry replicated.
- **View Log** - This button lets you view the change log. You can click on the up and down arrows to select the number of lines you would like the system to display.

- **New** - This button lets you add a new task. You add a task by clicking New.
- **Copy** - This button lets you copy a task. You copy a task by selecting it from the list in the Task Name field. After you select the task, click Copy and specify the nes task name.
- **Remove** - This button lets you remove a task. You remove a task by selecting it from the list in the Task Name field. After you select the task, click Remove.
- **Properties** - This button lets you view the properties of a task.

3 Click **New** to open the New Task Configuration window. Use the Replication Manager Task Configuration window to create new tasks and modify existing ones. This window contains five tab panels that allow you to design replication tasks with a high degree of flexibility.

4 Enter a name for the new task in the **Task Name** field.

5 Choose one of the following tabs to configure this task: **Connection Settings**, **Schedule**, **User Mapping**, **User Properties Mapping**, or **Group Mapping**.

When you click **Apply** on this panel, the changes you make to a task are saved. When you click **OK**, all changes are saved and the Task Configuration panel closes.

Connection Settings

To configure connection settings:

1 Click Connection Settings from the SecureControl Replication Agent: Task Configuration window to open the Connecting Settings tab.

2 Enter the following information in the *LDAP Supplier* panel.

- **Supplier Type** - The type of directory.
- **Host** - The hostname of the machine where the LDAP server is running.
- **Port** - The port the LDAP server is listening on.

- **Bind DN** - The Distinguished Name (DN) of the LDAP entry that has permission to access the change log in the LDAP Server. This can be the unrestricted User *cn=Directory Manager* in the default Directory Server installation or an entry with the appropriate ACL assigned to it. See your Web server's administrator guide for details.
- **Password**: The password for the Bind DN.

3 Click the **Test** button in the *LDAP Supplier* panel to check that the information you supplied is valid. If you have any problems, double-check your LDAP Supplier Server configuration to make sure the settings you entered are correct. This must be configured properly for the replication to take place.

4 Supply the following information in the *ClearTrust SecureControl API Server* panel.

- **Host**: the host where the ClearTrust SecureControl API server resides.
- **Port**: the port the ClearTrust SecureControl API server listens on.
- **Admin User**: the username of the ClearTrust SecureControl User with the appropriate administrative permissions to replicate.

NOTE: Either this admin User must be a Super User, or all users to be replicated must be in the same Administrative Group as the admin User.

- **Password**: the admin User's password.
- **Admin Rol**e: the role for the admin User.
- **Enable SSL**: specify whether or not the API server has been configured to require the use of SSL. All software components in the environment must be configured for SSL or none of them should be. Check the SSL setting of the default.conf files for all components (ClearTrust SecureControl servers, Web Server Plug-in, and the Web Server itself) if you're not sure.

5 Click the **Test** button in the *ClearTrust API Server Information* panel to test the information. A window appears displaying the results of the connection. If errors occur, double-check your ClearTrust SecureControl API Server configuration to make sure the settings you entered are correct. The Connection Settings tab is now complete.

Schedule

To configure schedule settings:

1 Click **Schedule** from the SecureControl Replication Agent: Task Configuration window to open the Schedule tab.

2 To configure how you want the LDAP replication scheduled, click on either **By Interval** or **By Time of Day**.

- **By Interval** - You can choose to replicate at minute intervals throughout the day by entering the number of minutes in the **Replicate every** field.
- **By Time of Day**- You can choose to replicate at a particular time each day by entering the time in 24-hour format.

User Mapping

To configure rules that map LDAP entries to SecureControl Users:

1 Click **User Mapping** in the SecureControl Replication Agent: Task Configuration window to open the User Mapping tab. The User Mapping tab displays the User Filter Rules that you need to specify types of filter rules to map LDAP entries to SecureControl Users. These rules are in a pull-down list and are defined as DN Filter Rule and Objectclass Filter Rule. You can specify either or both for user mapping.

2 If you specify a DN filter, only the entries under that DN will be replicated. When you specify an Objectclass filter, only entries of that Objectclass will be replicated. When multiple filters are created, they are linked together logically as Boolean *ORs*. That is, entries that meet any one of the requirements will be replicated. In addition, you must map the appropriate fields in LDAP that will be used to populate the required fields for each entry in ClearTrust SecureControl.

NOTE: You must be very specific when entering DN Filter Rules for Users and Groups (as discussed later) to avoid conflicts. The DN Filter Rules for Users and Groups must be distinct or entries will be mapped incorrectly.

3 Click **Add** to create a new rule. You can also highlight a rule and click on **Modify** to alter that rule. Or you can highlight a rule and click **Remove** to delete that rule.

4 Check either **All**, **Missing**, or **None** in the Generate Expiration field of the User Accounts Expiration portion of the panel.

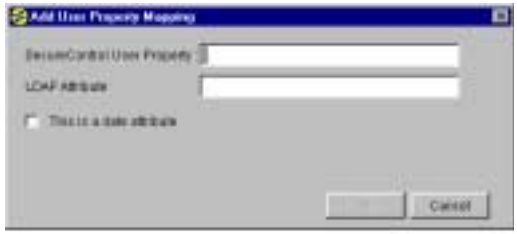
- Select **All** to generate expiration for all LDAP Users.
- Select **Missing** to generate expiration only for missing LDAP users.
- Select **None** to generate no expiration for user accounts.

5 Check either **At this Date** and **Set Date** to specify expiration date. Or check **Expires in** and choose a number from the pull-down menu to designate the number of months before user accounts will expire.

To map additional LDAP attributes to ClearTrust SecureControl User Properties:

1 Click **User Properties Mapping** in the SecureControl Replication Agent: Task Configuration window to open the Properties Mapping tab. This displays the Standard Properties Mapping Field that contains Standard Property information and LDAP Attribute information. Type in the LDAP Attribute to be mapped to standard SecureControl User Properties.

2 You can use the **Add** and **Remove** buttons to map or remove additional attributes stored in LDAP to User Properties in your ClearTrust SecureControl database. For example, if you keep track of a User's contract number in your LDAP database and want this data to be available in ClearTrust SecureControl, you would add this property. Clicking **Add** displays the **Add User Proper Mapping** window as shown below.



3 Enter the name of the **SecureControl User Property** to which you would like to map the LDAP attribute.

4 Enter the name of the **LDAP Attribute** you would like to map. If no LDAP attribute is specified, the system uses the default. Default User Property mappings are shown below.

TABLE 3-12: Default User Property and User Flag Mappings

	Entitlements Database	LDAP Attribute
User Properties	User Name	uid
	First Name	givenname
	Last Name	sn
	Email	mail
	Password	userpassword
User Flags	public	ispublic
	superuser	issuper
	helpdesk	ishelpdesk

5 If the property is a date, click the **This is a date attribute** checkbox on the Add User Property Mapping to display the date format field. You can replace the default date format using the formatting conventions in

The default date format uses two-digits or characters for all values except for the year, which is four digits, and the time-zone, which is three characters or digits.

You can use the table below to create your own date-time string patterns.

Two-characters or any number of the ‘y’ character result in digits. For example, for the month of July, the character sequence

MM= 07

while MMM = Jul

Only the y character can be used for digits.

Symbol	Meaning	Datatype	Example
G	era designator	text	AD
y	year	digit	1996
M	month	text or digit	July and 07
d	day	digit	10
h	hour in am/pm (1-12)	digit	12
H	hour in 24-hour day (0-23)	digit	0
m	minute in hour	digit	30
s	second in minute	digit	55
S	millisecond	digit	978

Symbol	Meaning	Datatype	Example
E	day in week	text	Tuesday
D	day in year	digit	189
F	day of week in month	digit	2 (2nd Wed in July)
w	week in year	digit	27
W	week in month	digit	2
a	am/pm market	text	PM
k	hour in day (1-24)	digit	24
K	hour in am/pm (0-11)	digit	0
z	time zone	text	Pacific Standard Time
‘	escape for text	delimiter	
‘’	single quote	literal	

Group Mapping

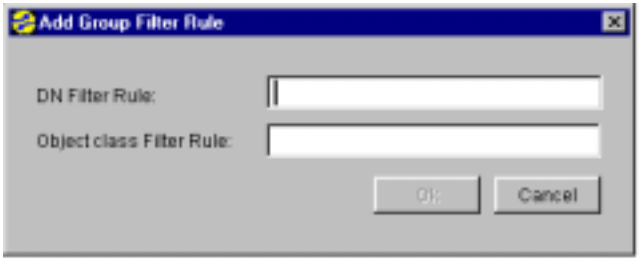
To specify rules to map LDAP entries to SecureControl Groups:

1 Click **Group Mapping** in the Replication Manager Task Configuration window to open the Group Mapping tab. The Group Mapping tab displays the Group Filter Rules that you need to specify to map LDAP entries to SecureControl Groups. These rules are defined as DN Filter Rules or Objectclass Filter Rule. You can specify one or both filter rules for group mapping.

If you specify a DN filter, only the entries under that DN will be replicated. When you specify an Objectclass filter, only entries of that Objectclass will be replicated. When multiple filters are created, they are linked together logically as Boolean *ORs*. That is, entries that meet any one of the requirements will be replicated.

NOTE: You must be very specific when entering DN Filter Rules for Users and Groups to avoid conflicts. The DN Filter Rules for Users and Groups must be distinct or entries will be mapped incorrectly.

2 Click **Add** to create a new rule. You can also highlight a rule and click on **Modify** to alter that rule. Or you can highlight a rule and click **Remove** to delete that rule.



3 In the *Group's Unique Members LDAP Attribute* panel, enter the group's unique member attribute in LDAP that will be used to identify members (users) of each group in SecureControl.

Chapter 6
Redundant Authorization Servers

Reliability of high volume components is a key requirement for distributed software. Yet because they're 'high volume,' such components are also typically the most likely to suffer from performance and reliability problems. For ClearTrust SecureControl, the high volume component is the Authorization Server, which must be available every time a User wants to access a Web Application.

The ClearTrust SecureControl system's architecture provides for failover among multiple Authorization servers to provide redundancy, ensuring that Authorization Servers are always available to their job. This chapter describes how the ClearTrust SecureControl architecture supports failover of Authorization Servers.

Authorization Server and Dispatcher

While the Authorization Server is the highest volume component, the Dispatcher is a low volume component.

The ClearTrust SecureControl Authorization Server handles authorization queries from ClearTrust SecureControl Web Server Plug-Ins.

Each Plug-In transmits its queries and receives responses from an Authorization Server over a single, dedicated TCP connection.

The Dispatcher helps establish and re-establish the connection between Authorization Server and Web Server Plug-in as needed. It's a low volume component with only this function, to assist Plug-ins and Authorization Servers in finding each other.

The Server Dispatcher supports Authorization Server failover as follows.

- When a ClearTrust SecureControl Authorization Server goes down or encounters problems in fulfilling authorization requests, any Plug-in currently connected to will detect the source of the problem, whether it is a dropped TCP connection or significant delay.
- The Plug-in makes a request to the ClearTrust SecureControl Server Dispatcher to locate a working instance of an Authorization Server. The Server Dispatcher maintains an internal list of all Authorization Servers that are registered with it and are responding to periodic test authorizations. It immediately returns the contact information for one or more Authorization Servers to the Plug-In.
- The Plug-in establishes a connection with the indicated Authorization Server(s) and continues servicing requests. The Web server requests that were pending at the time of the Authorization Server failure, do not fail.

Standard Mode vs. Distributed Mode Authorization Servers

ClearTrust SecureControl Plug-ins operate in one of two modes: standard and distributed. Each mode has failover capabilities.

- Standard mode means that ClearTrust SecureControl is running its Authorization servers with a primary Authorization Server and one or more stand-by Authorization Servers. The primary Authorization Server handles all the requests from the Plug-In. The stand-by Authorization Servers are used when the primary Authorization Server is unavailable.
- Distributed mode means that ClearTrust SecureControl is running multiple Authorization Servers across multiple servers. Distributed Authorization Servers can run on NT and UNIX servers simultaneously. The distributed mode load balances Web Server requests in a round-robin fashion across all the Authorization Servers.
- If a single Authorization Server becomes unavailable, the remaining Authorization Servers continue to fulfill Web Server requests.

Web Server Plug-in	Dispatcher	Primary Authorization Server	Standby Authorization Server
Queries Dispatcher for an available Authorization Server.	Sends Authorization server list to Web Server Plug-ins.	Upon startup, notifies Dispatcher of availability.	

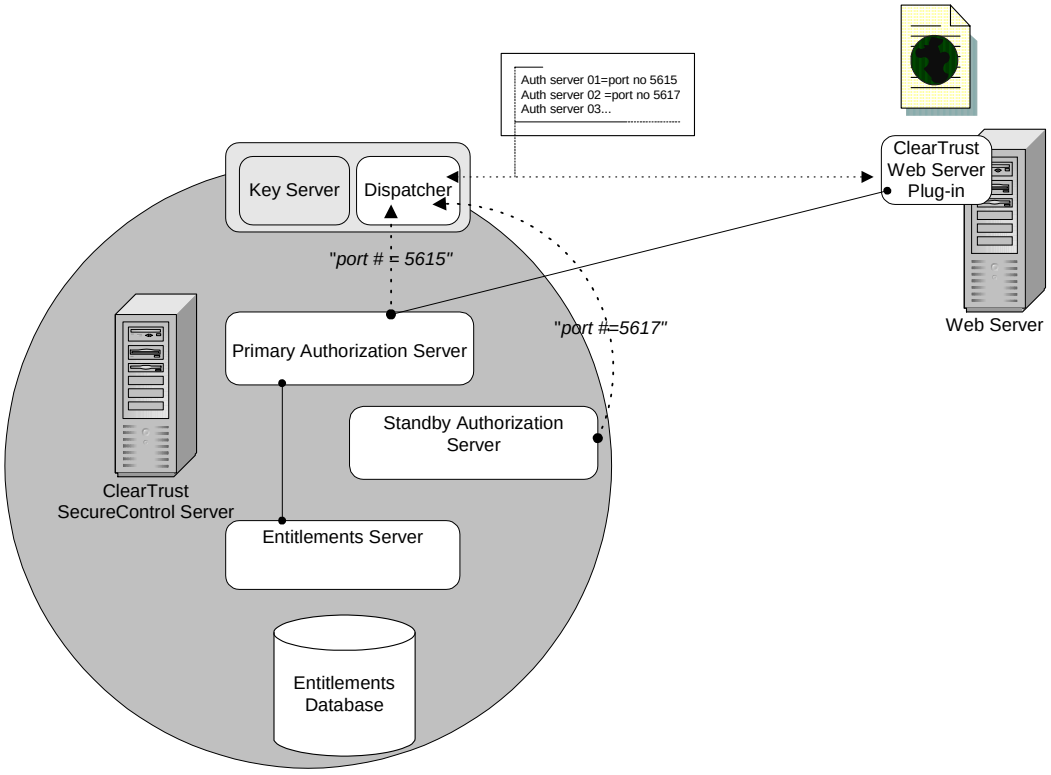
Web Server Plug-in	Dispatcher	Primary Authorization Server	Standby Authorization Server
	Pings Authorization Servers each minute. If no reply for 5 pings in a row, sends email message to Administrator. Queries all Authorization Servers to verify availability.		

Standard Mode

Here's how the standard mode start-up process and failover works:

1. At start-up, the Authorization Servers notify the Server Dispatcher of their availability.
2. When the Web Server Plug-ins are started, they query the Server Dispatcher for an available Authorization Server. The Server Dispatcher queries all the Authorization Servers and selects one of the working servers.
3. The Server Dispatcher sends the contact information (hostname or IP address plus TCP port number) for all Authorization Servers to the Plug-ins.
4. The Plug-ins query the primary Authorization Server for authorization requests. The primary Authorization Server queries the database for entitlements and responds to the Plug-ins' requests.

FIGURE 6-1: Standard Mode Authorization Process



5 If the primary Authorization Server is unavailable, the Plug-in's requests time out after a configurable time period. The Plug-in then needs to access another Authorization Server.

6 The Plug-in contacts the Server Dispatcher to request contact information for a working Authorization Server.

7 The Server Dispatcher has been querying all the registered Authorization Servers to verify availability about once a minute. When it finds an Authorization Server that fails the test query five times in a row, it assumes that server is down and notifies an Administrator via email. An error log is written to record the failure.

8 The Server Dispatcher chooses one of the working Authorization Servers (those that passed the last test query) and sends the contact information of that Authorization Server to the Plug-in. It also immediately schedules another round of test querying. The choice of which Authorization Server to use is deterministic, so that if multiple Web servers all need to failover, they will do so to the same Authorization Server so that the benefits of caching are maximized.

9 The Plug-in begins querying the stand-by Authorization Server for authorization requests.

Distributed Mode

Here's how the distributed mode start-up and failover process work:

1 When the Authorization Servers are started they notify the Server Dispatcher of their availability.

2 When the Web Server Plug-ins are started, they query the Server Dispatcher for available Authorization Servers. The Server Dispatcher queries all the Authorization Servers to verify that they are available.

3 The Server Dispatcher sends a list of all of the available Authorization Servers (host names or IP addresses, and their port numbers) to the Plug-ins.

4 The Plug-ins then start querying the Authorization Servers in a round robin fashion for authorization requests.

5 If any Authorization Server becomes unavailable, the request of the Plug-In times out after a configurable time period. The Plug-in then needs to refresh its list of working Authorization Servers.

6 The Plug-in contacts the Server Dispatcher to request a new list.

7 The Server Dispatcher, meanwhile, has been periodically querying all the registered Authorization Servers to verify availability, roughly once per minute. When it finds that one of the Authorization Servers has failed the test query five times in a row, it assumes that server is down and notifies an Administrator via email. An error log is written to record the failure.

8 The Server Dispatcher then sends the contact information of all the available Authorization Servers to the Plug-in. It also immediately schedules another round of test querying.

9 The Plug-in continues its round robin querying of the new set of Authorization Servers for authorization requests.

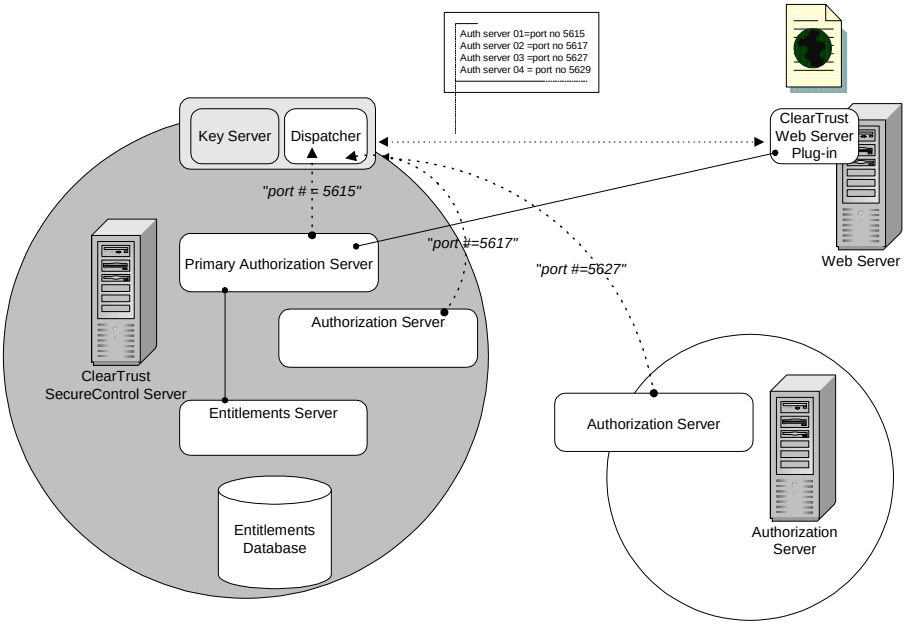


TABLE 6-2: Distributed Mode Authorization Process

Both standard and distributed modes provide failover capability. The essential difference between the two is that Web Servers configured in standard mode direct all of their authorization requests to just one of the available Authorization Servers, while in distributed mode, requests are spread out over all Authorization Servers.

Because all requests go to the same Authorization Server, the server's caches in standard mode are used to full advantage, resulting in fewer database hits and, therefore, greater performance. Under very high load, however, distributed mode achieves better performance because of increased parallelism.

Organizations that have a widely distributed user community or a large numbers of resources to secure may wish to install stand-alone Authorization servers at strategic locations throughout the intranet or extranet for load-balancing purposes.

You can install a stand-alone Authorization server on Solaris or on Windows NT Server as detailed in this chapter. The Windows NT installation takes less than an hour; the Solaris installation is a bit more complex and time consuming, but ultimately depends upon your expertise with Oracle database server installations on the Solaris platform.

Installing and Configuring Stand-Alone Authorization Servers

To install a stand-alone Authorization server on Solaris:

1. Follow the instructions for "Setting up the Operating System as User ROOT on Solaris" as described in the SecureControl Installation and Configuration Guide.
2. Login as the *oracle* user.
3. To run the Oracle installer software, change to the `/orainst` sub-directory of the Oracle CD-ROM and type:

   ```
   ./orainst
   ```
4. Make sure to install the SQL*Net client and TCP/IP adaptor software.
5. When the installation is complete, go to the `Network/Admin` sub-directory of the Oracle install and add the following entry to the tnsnames.ora file. If the tnsnames.ora file does not exist, create a file with that name and add this entry:

   ```
   <Database Instance Name>.world=
   (DESCRIPTION=
   (ADDRESS_LIST=
   (ADDRESS=
   (PROTOCOL=TCP)
   (Host=<Database Host Name>)
   (Port=<Database Port>)
   )
   )
   (CONNECT_DATA=(SID=<Database Instance Name>)
   )
   )
   ```

   where *Database Instance Name* is the name of the database instance that contains the SecureControl data; *Database Host Name* is the name of the machine on which the SecureControl database is running; and *Database Port* is the port on which Oracle is listening (usually 1521).

6 Start the SQL*Net client. You can test the SQL*Net connection using the TNSPing utility.

7 After you validate the SQL*Net connection, insert the SecureControl CD-ROM and run the install.script program.

8 Choose to install the server but **do not** choose to install the API.

9 Create the `Default.conf` file and enter the same variables as you did when you installed the Entitlements Server.

10 Do not run the Database Installer portion of the install.script program.

11 The Authorization Server installation should be complete and you should be able to run the stand-alone authorization server.

To run the SecureControl installer:

1 Log on to Windows NT as an administrator.

2 Exit all currently running programs.

3 Insert the SecureControl CD-ROM to automatically launch the installer program. From the installer program, you can choose which components you want to install. The SecureControl Server is selected by default.

4 Click Install to begin the installation. A verification dialog appears, asking if you want to install the SecureControl server.

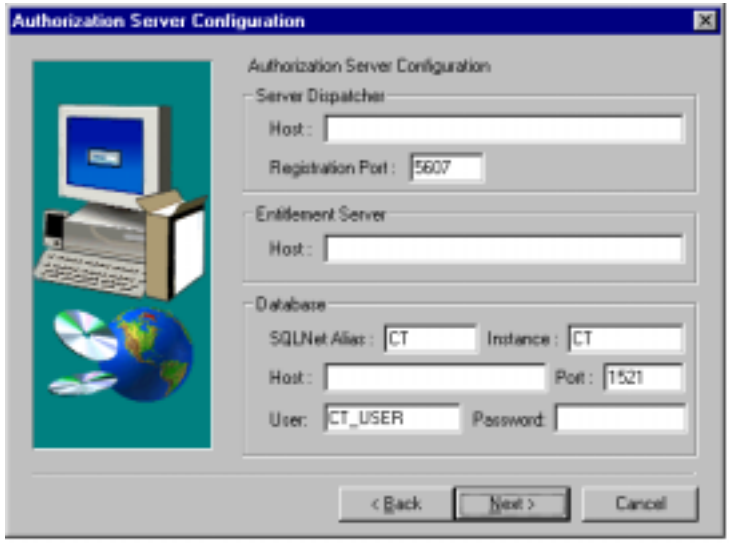
5 Click OK to start the InstallShield installer program.

6 Read the software license terms and if you agree to the terms, click Yes to continue with the installation.

7 You are prompted to select a directory for the SecureControl server installation. Select a directory for your SecureControl server installation.

8 Click Next to proceed with the installation. The system displays the Authorization Server Configuration window.

9



10 Enter the Host name for the Server Dispatcher and then tab to the Entitlements Server area and enter the Host name for the Entitlements Server. Tab to the Database area and enter the Host name and Password for the Database.Choose a database password. When selecting a password, a good rule of thumb is to use a mixture of upper- and lower-case letters. Include at least one digit or punctuation mark and try to avoid words that might be found in a dictionary.

11 Click Next to proceed with the installation.

12 Enter the License Info and License Key you obtained from Securant. Be careful to enter the information exactly as provided. If either the License Info or the License Key is entered incorrectly, the SecureControl server will not run.

13 Click Yes in response to restart your computer. After your computer restarts, SecureControl performs some initial configuration. This process may take several minutes.

14 This installation places the installation programs for distributed Authorization Servers in a directory named `Installs`.

Chapter 7
Installing the ClearTrust Management Tools

Managing the security environment for ClearTrust SecureControl protected resources is handled by two primary client applications:

- ClearTrust SecureControl Entitlements Manager (referred to simply as "ClearTrust SecureControl Manager"), which lets you manage users, groups, realms as well as all the resources you want to secure;
- ClearTrust SecureDetector, which lets you monitor threats to your resources.

This chapter tells you how to install both of these management applications on Windows NT or Solaris operating systems. It includes the following sections:

- Installing ClearTrust SecureControl Manager
- Installing ClearTrust SecureDetector Manager

See the *Policy Administrator's Guide* for information about using these adminstration tools.

Installing ClearTrust SecureControl Manager

The ClearTrust SecureControl Manager is a Java application (Java 1.2.2 runtime environment) that runs on Windows NT or Solaris. The Manager installs using Java-based InstallAnywhere installation shield, so the process is essentially the same, regardless of platform.

TABLE 6-3: ClearTrust Manager Requirements

	Solaris	Windows NT
Operating system	Sun Solaris 2.5.1 (or newer)	Windows NT 4
Patches	Patch cluster [anything?]	Service Pack 4
Other software	JRE 1.2.2	JRE 1.2.2
Processor	Sun Ultra	Pentium II, 300-Mhz (or faster)
RAM	32 MB	32-MB minimum
Hard-drive space	15-MB	15-MB

After ensuring that your platform meets the minimum requirements, you can install ClearTrust SecureControl Manager.

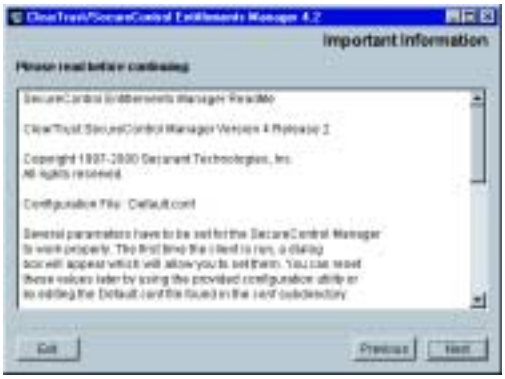
The installation program (`MgrInst.exe`) is located on the CD in the `\installs` sub-directory, and it's also located in the `\installs\client` sub-directory of the machine where you installed the ClearTrust SecureControl servers. You can launch this directly from either Windows NT or Solaris by invoking the filename. (On Windows NT, the installation program is lauched automatically when you select "ClearTrust SecureControl Manager" from the ClearTrust SecureControl startup intallation menu.)

To install ClearTrust SecureControl Manager on Solaris:

1. Copy the file `Securant/SecCtrl/installs/client/MgrInst.exe` from the ClearTrust distribution into a temporary directory on the target machine.
2. Go to the temporary directory and type

   `./Install_SecureControl_Mgr`

The InstallAnywhere wizard loads and guides you through the installation process. A licensing message displays, followed by an informational message, as shown in the screenshot. This is the same information discussed in “Starting the ClearTrust Manager” and in the “Configuration Parameters for First-time Run” table (Table 6-4).



The only information you need to provide during installation is the directory path for the install. When the installation is complete, a client startup script called `SCManager` appears in the directory you identified. See “Starting the ClearTrust Manager” for additional preliminary setup details.

To install ClearTrust SecureControl Manager on Windows NT:

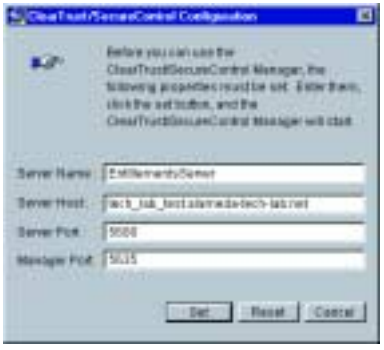
1. Insert the CD into the target machine’s CD-ROM drive. (Alternatively, you can copy `MgrInst.exe` from the `..Securant/SecCtrl/installs/Client/` sub-directory where the ClearTrust SecureControl servers are located to the target machine and launch it.) The InstallShield wizard starts automatically, displaying a menu of ClearTrust SecureControl components.
2. Select the radio button next to ClearTrust SecureControl Manager and click the Install buttton to continue with the installation.

   Shortly, InstallAnywhere loads and guides you through the installation process. A licensing message displays, and then an informational message displays, as shown in the screenshot. This is the same information discussed in “Starting the ClearTrust Manager” and in the “Configuration Parameters for First-time Run” table (Table 6-4). Click the Next button to proceed through these messages. The only information you need provide during installation is the directory path for the install.

When installation completes, *ClearTrust SecureControl Entitlements Manager* is a menu item on the Windows NT Start menu (Programs\Securants\ClearTrust-SecureControl\Manager). See “Starting the ClearTrust Manager” for additional setup details.

Starting the ClearTrust Manager

The first time you run a newly installed copy of the ClearTrust SecureControl client, you’ll need to confirm some parameter settings to enable the client software to identify and connect to the ClearTrust SecureControl Entitlements Server.



You can typically accept the defaults and only need enter the host name of the server on which SecureControl Entitlements server is running. (If you changed any of the default port numbers when you installed ClearTrust SecureControl, you must change the entries here to match.) See Table 6-4 for details about all four parameters and information about the values to enter.

TABLE 6-4: Configuration Parameters for First-time Run

Parameter	Description	Usage note
Server name	Name of the ClearTrust SecureControl Entitlements Server	Must match the ClearTrust SecureControl’s default.conf file `securecontrol.eserver.name` parameter. Default is “EntitlementsServer.”
Server host	DNS name of the machine running the ClearTrust SecureControl Entitlements server	Enter the host name of the machine on which you installed the ClearTrust SecureControl servers.

Parameter	Description	Usage note
Server port	Port number of the Voyager ORB port used for communication between the Manager software and the Entitlements server.	Must match the value for the `securecontrol.eserver.voyager_port` in the ClearTrust SecureControl default.conf file. Default is 5600.
Manager port	Port number of the client workstation on which the ClearTrust Manager is installed.	Leave unchanged unless you have reason to change. Default is 5635.

3. Enter the host name of the machine on which ClearTrust SecureControl servers are running. Change any of the other settings as required.
4. Click the Set button to save the Manager's configuration and continue. Shortly, the ClearTrust SecureControl Manager login dialog displays.
5. Enter the default Administrator's name and password. These are "admin" and "admin," all lower-case letters.

> **WARNING** In a production environment, be sure to change the `admin` account password immediately (or disable it entirely after you've created bona fide accounts for Administrators using the Manager), before you begin to administer Users and Resources. See the *Policy Administration Guide* for information.

The Manager client application connects to the Entitlements server and the management GUI displays.

Setting the port numbers, host name, and server name for the Manager is a one-time event: Once you've done this, the next time you launch the Manager, you'll simply see the login name and password prompt. See the *Policy Administration Guide* for more information about using ClearTrust SecureControl Entitlements Manager.

If the Manager doesn't display:

1. Make sure the ClearTrust SecureControl servers have all started.
2. Check the values for the parameters in the default.conf file and make sure they match the values in the `/manager/conf/default.conf` as required (see Table 6-4).
3. Look for error messages in the `/logs` directory where ClearTrust SecureControl is installed. See Appendix D, "Troubleshooting" on page 167 for more information about log messages.

Installing ClearTrust SecureDetector Manager

SecureDetector is a fully-integrated module of SecureControl that monitors access attempts at the application layer. SecureDetector comprises a management tool and a background process, the SecureDetector Daemon.

Detecting events is a time-sensitive taks, so Securant recommends that you install SecureDetector on the same machine as ClearTrust SecureControl. If you want to install on another machine, be sure to synchronize the time on both machines, so that events will be time-stamped and logged correctly.

See the Policy Administration Guide for information about configuring and using SecureDetector.

The installer for SecureDetector is located on the ClearTrust SecureControl CD, in the `/SecureDetector` directory.

To install SecureDetector on Solaris:

1. Log in as `root` on the target machine.
2. Mount the CD drive.
3. Go to the /SecureDetector directory in and type:

   `./SDInstall.bin`

   The InstallAnywhere application launches and guides you through the installation process. The only information you to provide for the installation is the name of the directory in which to install.

To install ClearTrust SecureDetector Manager on Windows NT:

1. Insert the CD into the target machine's CD-ROM drive and double-click on the filename to launch the installer.
2. InstallAnywhere loads and guides you through the installation process; the only information you need provide during installation is the directory path for the install.

   When installation completes, *ClearTrust SecureDetector* is a menu item in your Securant applications group, on the Windows NT Start menu.

   See the *Policy Administrator's Guide* for details about using this tool to setup monitoring and alerts, and to report on access events in your SecureControl protected resources.

Parameter	Description	Usage Note
securecontrol.plugin.securid_sdconf_file_loc	Specifies the directory path (Solaris only) to the sdconf.rec file; it does not specify the complete path of that file.	For example: if sdconf.rec is located in /ace/data: securecontrol.plugin.securid_sdconf_file_loc=/ace/data

Chapter 8
ClearTrust Components and Firewalls

Firewalls are a foundation element of the virtual enterprise network, typically providing protection against threats at the network layer. The three broad general categories of this network infrastructure device are:

- Packet-level (or filter-based)
- Application-level (or proxy-based)
- Circuit-level

For any two ClearTrust SecureControl components to communicate across a firewall, you must configure the firewall to allow connections on a specific port.

To run a particular component across the firewall from your ClearTrust SecureControl server, you must configure your firewall to allow particular TCP/IP connections from the Web Server Plug-in, GUI, or API to the ClearTrust SecureControl server.

ClearTrust SecureControl components must make direct connections to the server, and so cannot run across purely application-level firewalls. To run ClearTrust SecureControl components across an application-level firewall, contact Securant.

For extranet applications especially you'll likely want to run ClearTrust SecureControl components in various configurations with firewall (see Figure 8-1).

You can configure the ClearTrust SecureControl Web Server Plug-in, the Manager, and the API client on the opposite side of firewall from the ClearTrust SecureControl server.

Each of these three components is configured separately and may be placed inside or outside the firewall regardless of how the other two are configured.

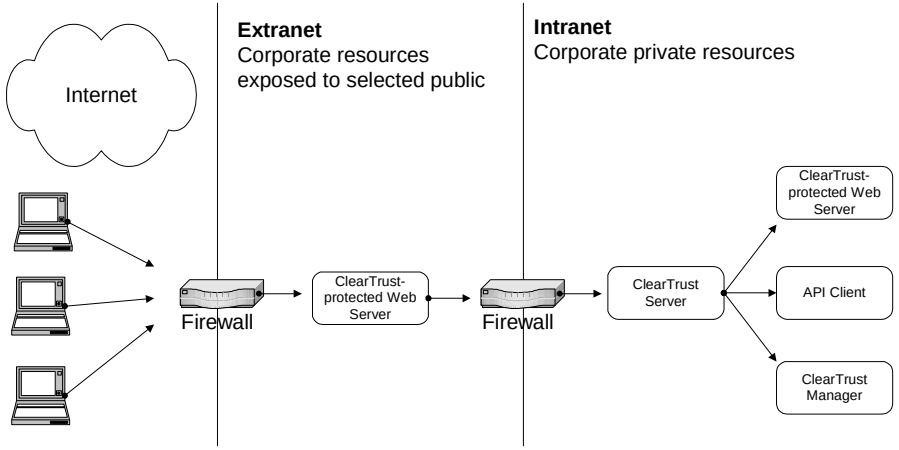


FIGURE 8-1: ClearTrust SecureControl Components Inside and Outside Firewalls

Running the Web Server Plug-in Across a Firewall

The ClearTrust SecureControl Web Server Plug-in makes three types of connections to the ClearTrust SecureControl server:

- Server Dispatcher (default port 5608)
- One or more Authorization Servers (default ports 5615 & 5617)
- Key Server (default port 5606)

The Server Dispatcher and the Key Server run on the same host. In most cases, the Authorization Servers runs on the same host as well, but you can run Authorization servers on a different host. You must select the hosts and ports for each of the connections.

A typical ClearTrust SecureControl installation consists of a single Server Dispatcher and Key Server running on the same host as a pair of Authorization Servers. To run the Plug-in across a firewall from your ClearTrust SecureControl server, configure each of the three components the firewall, the Plug-in, and the ClearTrust SecureControl server to use the appropriate ports.

Firewall Configuration

Configure your firewall to allow traffic through to port 5606, 5608, 5615, and 5617.

Select the port numbers and configure your firewall to allow connections to those ports. Note that these ports are the server ports; the client ports are assigned dynamically by the operating system at runtime.

Plug-in Configuration

You only need to configure the ports for the Server Dispatcher and the Key Server on the Plug-in. The Web Server Plug-in receives the Authorization Server host names and ports from the Server Dispatcher at runtime.

ClearTrust SecureControl Server Configuration

Next, configure the ClearTrust SecureControl server to use the appropriate ports.

Using a text editor, open the default.conf file (located in /Securant/ct_root/conf directory) and make sure the dispatcher's listener port number and key server port numbers match the port numbers you've configured at the firewall.

Locate the directory where the ClearTrust SecureControl server is installed

Within our example, CT_ROOT, is a directory called conf, and within conf is a file called Default.conf. Edit Default.conf and update the following lines to reflect the ports you wish to use:

```
securecontrol.aserver.dispatch.list_port
securecontrol.aserver.dispatch.key_port
```

Now the Server Dispatcher and Key Server will be listening on the correct ports.

Open the startup scripts for the Authorization servers (located in /Securant/ct_root/bin/Auth01.bat and Auth02.bat)

Next, make the corresponding change to the Authorization Servers.

The Authorization Servers are launched via scripts that set the appropriate parameters. To select the ports used by the Authorization Servers, edit the appropriate scripts. Within CT_ROOT there is a directory called bin which contains the scripts.

Edit auth00.bat and auth01.bat and look for one line in each file which contains -DLISTEN_PORT= and the port number. Edit that line so that the listen port is set to the port you would like to use.

SecureControl is now running JSSE so every process using Voyager needs to have its own port. The following configuration parameters have also been added to the Authorization server command line of auth00.bat and auth01.bat, respectively:

```
-DVOYAGER_PORT=5616 for auth00.bat
```

```
-DVOYAGER_PORT=5618 for auth01.bat
```

If you want to change these default values, you can edit these lines so that the port is set to the port you would like to use.

On Windows NT systems, you can also make these changes to the Registry by using the registry editor and changing the values in these keys (for authorization servers 00 and 01, respectively):

```
hkey_local_machine\system\currentcontrolset\services\javaser
viceauth00\parameters\options=”-noclassgc -ms32M -mx64M -
DLISTEN_PORT=5615-DVOYAGER_PORT=5616-
DCT_ROOT=d:\Securant\SecCtrl\ct_root”
```

```
hkey_local_machine\system\currentcontrolset\services\javaser
viceauth01\parameters\options=”-noclassgc -ms32M -mx64M -
DLISTEN_PORT=5616-DVOYAGER_PORT=5618 -
DCT_ROOT=d:\Securant\SecCtrl\ct_root”
```

ClearTrust SecureControl Manager and Firewalls

ClearTrust SecureControl Entitlements Manager communicates with the ClearTrust server using CORBA over TCP/IP. The ClearTrust Server includes a third party product called Objectspace Voyager that handles CORBA communications between components. Voyager uses one fixed port on the server for listening for inbound connections from the client to the server. You will need to select what port number to use for the fixed port. It also opens several connections from the server to the client out through the firewall.

Firewall Configuration

Configure your firewall to allow inbound connections from the ClearTrust SecureControl Entitlements Manager to the ClearTrust SecureControl server machine on a single fixed port of your choice. You must then configure the ClearTrust SecureControl server and ClearTrust SecureControl Entitlements Manager to use this port.

To configure the server, find the `securecontrol.eserver.voyager_port` parameter in your server's Default.conf configuration file. Set this parameter to the port number you have chosen. The server should now be able to receive inbound connections from the ClearTrust SecureControl Entitlements Manager

SecureControl runs with the JSSE and therefore every process that uses Voyager needs to have its own port. The default port for the Entitlements Manager is 5635. To change this default port number, configure the following parameter in the Default.conf file of the SecureControl Manager:

```
securecontrol.gui.voyager_port=5635
```

Using the ClearTrust SecureControl API Across a Firewall

All API activity is conducted over a single TCP/IP connection. By default, the API server listens on port 5601—you can select any other port you like.

To access the API across a firewall from your ClearTrust SecureControl server, configure the firewall, your API code, and the ClearTrust SecureControl server to use the appropriate port.

Firewall Configuration

Set up your firewall to allow incoming TCP connections to the ClearTrust SecureControl server on the API port. The port used by the client is assigned by the client operating system at runtime.

API Coding

C API

When establishing your connection to the API server using the ct_connect() call, pass the correct port in the *port* parameter.

Java API

When establishing your connection to the API server via the APIServer proxy constructor, pass the correct port in the second parameter.

ClearTrust SecureControl Server Configuration

Configure the ClearTrust SecureControl server to use the appropriate port for the API Server. Locate the directory where the ClearTrust SecureControl server is installed. For our example, we call that directory CT_ROOT.

Within CT_ROOT is a directory called *conf*, and within conf is a file called Default.conf. Edit Default.conf and update the following line to reflect the ports you wish to use:

```
securecontrol.eserver.api.port
```

Port Mapping Reference

Pre-4.2 Release	Release 4.2	Description
1570	5600	Entitlements Server 0 Voyager Port
5091	5601	Entitlements Server 0 API Port
5097	5606	Key Dispatcher 0 key_port
5098	5607	Key Dispatcher 0 reg_port
5099	5608	Key Dispatcher 0 list_port
5020	5615	Authorization Server 00 Listener Port
5120	5616	Authorization Server 00 Voyager Port
5021	5617	Authorization Server 01 Listener Port
5121	5618	Authorization Server 01 Voyager Port
5022	5619	Authorization Server 02 Listener Port
5122	5620	Authorization Server 02 Voyager Port
5023	5621	Authorization Server 03 Listener Port
5123	5622	Authorization Server 03 Voyager Port
5024	5623	Authorization Server 04 Listener Port

Pre-4.2 Release	Release 4.2	Description
5124	5624	Authorization Server 04 Voyager Port
3067	5635	Manager 0 Voyager Port
n/a	5636	Manager 1 Voyager Port
2389	5645	LDAP Replication Agent 0 Port
1389	5646	LDAP Replication Manager 0 Port

Appendix A
ClearTrust Configuration File Reference

The ClearTrust SecureControl system includes several key components that function by sending data through various well-known (in the context of the ClearTrust system, not well-known in the broader sense) ports. A default.conf file contains the names and port addresses of much of the system. ClearTrust clients access the API and Entitlements Server through ORB (object request broker) and port definitions.

The server side configuration information is contained in the `Default.conf` file located in the `ct_root/conf` subdirectory. The `Default.conf` file is built during the ClearTrust SecureControl installation.

Each Authorization Server has a port that it uses to communicate with ClearTrust SecureControl-protected Web Servers. t the port number is supplied to the Authorization Server as a startup parameter and is not part of the `Default.conf` file

ClearTrust SecureControl Default.conf Parameter Settings

Database Parameters The Authorization Server and the Entitlements Server establish connectivity with the database using these parameter settings.

Parameter	Description	Usage Note
securecontrol.db.admin.name	Name of the database user account that the data server uses to connect	
securecontrol.db.admin.password	Password for the User account that the data server uses to connect	

Parameter	Description	Usage Note
securecontrol.db.driver	Class name of the JDBC driver used by the ClearTrust SecureControl servers to connect to the database	For Sybase database, value is com.sybase.jdbc2.jdbc.SybDriver
securecontrol.db.instance	Name of the database instance file to which the ClearTrust SecureControl servers are connecting	
securecontrol.db.owner.name	Name of the database User account that owns the objects in the ClearTrust SecureControl schema	
securecontrol.db.owner.password	Password for the database User account that owns the objects in the ClearTrust SecureControl schema	
securecontrol.db.tablespace.data	Tablespace or segment where the tables of the ClearTrust SecureControl schema are kept	
securecontrol.db.tablespace.index	Tablespace or segment where the indices of the ClearTrust SecureControl schema are kept.	
securecontrol.db.type	Type of database	Oracle or Sybase
securecontrol.db.url	JDBC URL used to connect to the database	
securecontrol.db.user.name	Name of the database User account the authorization server uses to connect	
securecontrol.db.user.password	Password for the above User account	
securecontrol.db.rdbmserver	Name of the database management system server.	Relevant only for Sybase installations
securecontrol.db.table.indexed	Specifies whether or not the Entitlements Server creates an index in memory (cache) of the database tables[a]	For large databases (>100K users, groups, and realms), set to *no* (to leave indexing off). For smaller datasets, you can turn indexing on by setting to *yes*.
securecontrol.db.sd.tablespace.data	Tablespace or segments where the tables of the ClearTrust Secure Detector schema are kept	
securecontrol.db.sd.tablespace.index	Tablespace or segments where the indices of the ClearTrust Secure Detector schema are kept	

a. The internal indexing provides for faster scrolling (mostly used in the Manager) of large data sets. While data sets smaller than 100,000 will benefit from increased response time (most notably in the Manager), it is recommended that you do not use the indexing feature for large data sets because indexing is memory- intensive feature. The memory signature of the Entitlements Server will probably become unwieldy at about 100,000 Users/Groups/Realms.

Authorization Server Parameters These parameters define the behavior of the Authorization Server.

Parameter	Description	Usage Note
securecontrol.net.ssl.use	Instructs the Authorization Server and API to use SSL for communications	Specify *yes* to use SSL or *no.* Value must match securecontrol.plugin.ssl.use parameter in Default.conf files of all Web Server Plug-ins
securecontrol.aserver.authorization_mode	Sets the Authorization Server to either active or passive mode. In active mode, only resources that have an associated application function are protected. In passive mode, all resources are protected whether or not they are in an application. In either mode, explicit permission (either through Basic Entitlements or Smart Rules) is necessary for a user to access any protected resource.	Default value is *active.*
securecontrol.aserver.num_db_connections	Number of database connections to the database that the authorization server can open.	Enter a number 1 thru 5; default setting is 4. Using more than one connection may improve performance when the running an authorization server in distributed mode, but using more connections requires more memory.
securecontrol.aserver.thread_pool_size	internal configuration option.	Default value is 20. Do not change.
securecontrol.aserver.user_activity_log_delimiter	Character used to separate the columns in the User Activity Log	Default value is comma (,); any characters allowed.
securecontrol.aserver.user_activity_log_level	Controls the amount of information sent to the User Activity Log. Higher levels indicate more detailed logging. Higher levels log all events from lower levels plus additional events.	0 – No logging 10 – User validation errors (default; for example, bad password, expired account) 20 – Denials of access 30 – All accesses
securecontrol.aserver.user_activity_log_name	name of the User Activity Log	Default is CT_UserActivity; Any valid filename without extension is okay.
securecontrol.aserver.user_activity_log_size_in_k=1000	max size of the User Activity Log before it will archived with a date stamp.	Default is 1000. Any integer allowed.
securecontrol.aserver.dispatcher.encrypt_type	Toggles between encryption types.	Default is D (DES-EDE); Alternates are C (cleartext) or B (Blowfish).

Parameter	Description	Usage Note
securecontrol.aserver.dispatcher.key_port	Dispatcher's key port. This must match the port of the dispatcher's listener port.	Default value is 5606; Any valid dispatcher port number.
securecontrol.aserver.dispatcher.session_key_life	Sets session key lifetime.	Default is 1 hour. Any unit of time (0 to n seconds; 0 to *n* minutes; 0 to *n* hours).
securecontrol.aserver.max_connections	Total number of incoming connections from the Web Server Plug-ins.	Default is n/a. Any value 0 to *n* (that makes sense in the context of the specific Web server environment).
securecontrol.aserver.secure_detector.log_level	User logging levels (for user activity)	Default is 10. Options are 0, 10, 20, and 30.
securecontrol.aserver.multi_home_addresses={hostname_or_IP_1{:hostname_or_IP_2:...hostname_or_IP_n}}	Only necessary for multi-homed machines. A colon-delimited list of hostnames (or IP addresses). Binds IP addresses to server socket of the Authorization Server's ORB.	Leave blank unless server is multi-homed. Must contain all addresses (or hostnames) to ensure communication between Entitlements Server and Authorization Servers.
securecontrol.aserver.export_address=hostname_or_IP	Controls address (hostname or IP) that Plug-in uses to contact Authorization Servers.	Leave blank unless server is multi-homed, firewall, or multi-DNS configuration.

LDAP Parameters ClearTrust SecureControl provides LDAP authentication via the ClearTrust SecureControl Authorization Server and the ClearTrust SecureControl Web Server Plug-in.

Parameter	Description	Usage Note
securecontrol.aserver.ldapauth.reconnect_timeout	Amount of time the Authorization Server tries to reestablish a connection with an LDAP Server when the connection breaks.	Default is 1 minute; can be *n* of seconds; *n* of minutes; *n* of hours.
securecontrol.aserver.ldapauth.ldap_servers	LDAP server name. Separate multiple LDAP server names with a comma. Include port number if default number (389) not used.	Default port is 389; default for LDAPS is 636. securecontrol.aserver.ldapauth.ldap_servers=roadrunner.securant.com, taz.securant.com:390

Parameter	Description	Usage Note
securecontrol.aserver.ldapauth.base_dns	Base DN of the corresponding LDAP server.	Use commas to separate multiple servers. Use double quotes to distinguish attributes in base DN. For examplesecurecontrol.aserver.ldapauth.base_dns=o=securant.com,"o=xyz,c=us"
securecontrol.aserver.ldapauth.ldap_over_ssl	Connect to LDAP servers using SSL.	Specify 'yes' to use SSL (and make sure LDAP server is also configured for SSL) or 'no.' Separate multiple servers with comma.
securecontrol.aserver.ldapauth.num_concurrent_connection=	Maximum number of concurrent connections for each LDAP server specified in	Default is 5; values 1 - 20 inclusive acceptable.
securecontrol.aserver.ldapauth.ldapattr_map_scuid=	Configure alternative to UID/password authentication.	Map specified LDAP attributed to SecureControl UID. For example, securecontrol.aserver.ldapauth.ldapattr_map_scuid=uid,cn specifies that one LDAP server has UID deined and another LDAP server has Common Name (cn) defined.

Authorization Server Cache Sizing Parameters The ClearTrust SecureControl object model includes the following cache definitions.

Parameter	Description	Usage Note
securecontrol.aserver.cache.url_to_function	Maps a URL to the corresponding ACCESS Application Function object	Default size is 2500; maximum possible size is total number of files and directories on all Web Servers using the Plug-in.
securecontrol.aserver.cache.user	Maps a User name to a User object	Default size is 4000; maximum possible size is total number of Users in the Entitlementes database.

Parameter	Description	Usage Note
securecontrol.aserver.cache.user_property	Maps a User and a User Property Definition to a User Property.	Default size is 0; maximum possible size is total number of users *total number of user property definitions.
securecontrol.aserver.cache.smart_rule	Maps an Application Function to a list of Smart Rules	Default size is 250; maximum possible size is total number of Applications in the Entitlements database.
securecontrol.aserver.cache.master	Maps a User and an Application Function to a final result	Default is 8000; maximum possible size is total number of users * total number of Applications. Key: A User and an Application Function Value: A Boolean determining accessibility
securecontrol.aserver.cache.non_access_function	Maps an Application Name-Application Function Name pair to a given Application Function. The ClearTrust API uses this parameter to determine accessibility to functions (beyond ACCESS).	Default size is 500; maximum possible size is number of Application Functions in the Entitlements database.

Authorization Dispatcher Parameters The Authorization Server and Authorization Dispatcher communicate using the parameters in this table.

Parameter	Description	Usage Note
securecontrol.aserver.dispatch.admin_address	ClearTrust Administrator's email address.	
securecontrol.aserver.dispatch.smtp_host	Hostname of the Dispatcher's email server	
securecontrol.aserver.dispatch.host	Hostname of the Dispatcher process. Authorization Server uses this parameter to establish communication with a Dispatcher.	Must be a fully qualified domain name. For example, server-name.company.com
securecontrol.aserver.dispatch.reg_port	Port number of Dispatcher's registration port. Used during Authorization server initialization process. Authorization Server registers itself with a Dispatcher to advertise availability.	
securecontrol.aserver.dispatch.list_port	Dispatcher's listener port. ClearTrust Web Server Plug-in locates Dispatcher using this value. Dispatcher provides a list of available Authorization Servers in response.	Value must match listener port values in the Web Server configuration file.

Parameter	Description	Usage Note
securecontrol.aserver.dispatch.plugin_auth_format	Identifies the format of the information —IP address or hostname—sent from the Dispatcher to the Web Server Plug-In.	Default value is IP; can set 'hostname' as alternative.

Entitlements Server and API Server Dispatcher Parameters The Entitlements Server uses the following Dispatcher definitions to identify itself to the ORB and allow for client connections. The API Server (a component within the Entitlements Server) also uses Dispatcher definitions to establish a server socket connection.

Parameter	Description	Usage Note
securecontrol.eserver.api.log_size	Maximum size of the API Server's three log files (Error, Transactions, Logon).	
securecontrol.eserver.api.port	API Server's listener port. API clients connect to this port.	
securecontrol.eserver.host	Hostname of Entitlements Server process	Must be fully qualified domain name.
securecontrol.eserver.name	Name used to identify the Entitlements Server in the Voyager namespace.	
securecontrol.eserver.user.defaultLifetime	Default lifetime of a User's account	Default is
securecontrol.eserver.voyager.port	Voyager namespace port. ClearTrust Manager clients look up the Entitlements Server object on this port.	
securecontrol.eserver.admin.log.logToSecureDetector	Turns on login and error logging to SecureDetector log.	Default is 'no.' Can set to 'yes' to log SecureDetector logins and errors.
securecontrol.eserver.API.log.logToSecureDetector	Turns on logging for SecureDetector API transactions	Default is 'no.' Change to 'yes' to turn on logging.
securecontrol.eserver.multi_home_addresses={hostname_or_IP_1{:hostname_or_IP_2:...hostname_or_IP_n}}	Determines the IP addresses (or hostnames) that bind to the server socket of the Entitlements Server's ORB.	Leave blank unless server is multi-homed. Use colon-delimited list for multiple IP addresses (or hostnames) to enable Entitlements Server to accept connections on those addressses from ClearTrust Manager and Authorization Servers.

User Activity Event Log Parameters You configure the user activity event log via the `Default.conf` configuration file. You can change any of the following configuration options:

Parameter	Description	Usage Note
securecontrol.aserver.user_activity_log_delimiter	Identifies the delimiter characters to separate fields in the log files.	Default is comma (,). Any combination allowed (to support a range of report-generation tools).
securecontrol.aserver.user_activity_log_level	Determines the level of detail recorded in the log file	Default is 10; allowed values are 0, 10, 20, and 30.
securecontrol.aserver.user_activity_log_name	Determines the name of the user activity log file.	Default is CT_UserActivity. Do not include a suffix when selecting a name.
securecontrol.aserver.user_activity_log_size_in_k	Size, in kilobytes, at which log file will be rotated. Rotation means the existing log file is closed and hten renamed using a timestamp; a new file is opened.	Default is 1000.

Appendix B
Web Server Plug-ins Reference

Plug-in Configuration File Options

The ClearTrust SecureControl Plug-in configuration file has the following options:

Authentication Type and Resource List Parameters. You must configure each resource for the type of authentication you want the Web server to perform.

Parameter	Description	Usage Note
securecontrol.plugin.auth_resource_list	Specifies authentication modes for ClearTrust SecureControl-protected resources the Web Server	Default is BASIC. Options include CERTIFICATE, EDIRECT, SECURID, LDAP, and NT

Single Sign-on Parameters None of the SSO parameters apply unless the `securecontrol.plugin.sso` parameter is set to *yes*.

Parameter	Description	Usage Note
securecontrol.plugin.sso	Turns Single Sign-On functionality on (yes) or off (no);	Default is 'yes' (SSO is on). Set to 'no' to disable SSO.
securecontrol.plugin.session_lifetime	Limits how long a User can continue to access ClearTrust SecureControl-enabled Web Servers without being re-prompted for ID and password.	You can set the session life to any number of hours, minutes, or seconds, but decimals are not allowed. For example, 90 min is allowed, but not 1.5 hours
securecontrol.plugin.idle_timeout	Specifies how long the User can be idle before ClearTrust SecureControl requires re-authentication. If a user does not access the system for a length of time longer than the specified timeout, the system will request the user's ID and password again.	You can set the idle timeout to any number of hours, minutes, or seconds, but decimals are not allowed. For example, 90 min is allowed, but not 1.5 hours.

Parameter	Description	Usage Note
securecontrol.plugin.auto_challenge	Specifies that when either the session life or the idle timeout expires the browser automatically challenges the User to re-authenticate.	If you set the auto challenge to no, the User receives a 403 forbidden error message and must recycle the browser to re-request the page.
securecontrol.plugin.fudge factor	Specifies the allowable difference in clock time between the Web Servers.	You can set the fudge factor to any number of hours, minutes, or seconds, but decimals are not allowed. For example, 90 min is allowed, but not 1.5 hours.
securecontrol.plugin.secure	Indicates whether the ClearTrust SecureControl Plug-in should set the secure field in the cookie used for single sign-on. If you set the secure field to no, browsers will not send the cookie except over an SSL connection.	If you set the secure field to yes, you must also configure the Web Server for SSL support.
securecontrol.plugin.cookie_domain	Specifies the domain field of the cookie used for single sign-on. ClearTrust SecureControl cookies will be sent to all hosts in the specified domain, and thus single sign-on will be enabled for those hosts.	Normally you should set the cookie domain to yourcompany.com, but you can limit single sign-on to a particular subdomain.
securecontrol.plugin.path	Specifies the path field of the cookie used for single sign-on.	In most cases, you should not change this parameter.
securecontrol.plugin.cookie.expiration	Specifies the length of the cookie lifetime (as opposed to the session lifetime).	If this value is set, the browser saves the cookie to disk, thereby allowing the cookie (and thus the session) to continue between browser invocations. If this value is not set or it is set to a zero value, the browser does not save the cookie to disk and the User must re-authenticate between browser invocations
securecontrol.plugin.cookie_ip_check	Specifies whether the Web server Plug-in should compare the IP of the client machine with the IP stored in the cookie before using the cookie.	The default value is **yes**. There are certain environments where this functionality does not apply, however. For example, in an environment with load balancing across multiple servers, the client IP will constantly be different and so disqualifies the cookie (if there is no cookie, there is no SSO).

Authorization Server Parameters

Parameter	Description	Usage Note
securecontrol.plugin.auth_server_timeout	Number of seconds the Plug-in waits for replies from ClearTrust SecureControl Authorization Servers.	Default is 5 seconds; valid for most sites.
securecontrol.plugin.auth_server_mode	Determines how the Plug-in distributes authorization queries for consecutive HTTP requests across the set of ClearTrust SecureControl Authorization Servers.	*Standard mode* means that queries will continue to go to just one of the Authorization Servers as long as it is available. Any additional servers are used only when the first Authorization Server fails. *Distributed mode*, the Plug-in distributes queries among all known Authorization Servers in round-robin order. In general, Standard mode is preferred because it takes maximal advantage of the Authorization Server's caching. In environments with multiple Authorization Servers running on different machines, however, Distributed mode may achieve better overall performance.

Server Dispatcher Parameters

Parameter	Description	Usage Note
securecontrol.plugin.dispatcher_host	DNS name of the Server Dispatcher's host	
securecontrol.plugin.dispatcher_port	port number of the Server Dispatcher	Value must match the value given in the ClearTrust SecureControl file Default.conf for the parameter securecontrol.aserver.dispatch.list_port
securecontrol.plugin.dispatcher_timeout	Number of seconds the Plug-in will wait for replies from the Server Dispatcher	Default value is 10; should be sufficient for most installations
securecontrol.plugin.key_port	Port number of the Key Server	Must match the value given in the ClearTrust SecureControl file Default.conf for the parameter securecontrol.aserver.dispatch.list_port

Web Server Parameters

Parameter	Description	Usage Note
securecontrol.plugin.web_server_name	Name by which this Web Server is known to the ClearTrust SecureControl system	Must be an exact match to the name of the Web Server defined within the ClearTrust SecureControl Entitlements Database
securecontrol.plugin.realm	Specifies the name of the HTTP authentication realm of resources protected by this Web Server. (Not the same as the ClearTrust SecureControl concept of Realm.)	If configuring multiple Web Servers for ClearTrust SecureControl, use the same HTTP realm name. This can be anything, but it must be the same for all ClearTrust-protected Web Servers.
securecontrol.plugin.cookie_domain	Specifies the scope of ClearTrust SecureControl's Single Sign-On	Set to *yourcompany.com*, *yourorganization.org, yournetwork.net*, or sub-domain, such as *dept.company.com*. Do not use top level domain (.com or .org) alone.

SSL Parameters

Parameter	Description	Usage Note
securecontrol.plugin.use.ssl	Specifies whether the Plug-in should encrypt communications with the Authorization Server or not.	Default is 'no.' Setting must match the use_ssl setting in the ClearTrust Server default.conf. If set to 'yes,' configure the Web Server also.

Form-Based Authentication Parameters This section describes form-based authentication parameters.

Parameter	Description	Usage Note
securecontrol.plugin.form_based_enabled	Turns form-based authentication on or off.	Default is *yes* (on). Can set to *no* to turn off, in which case the browser's BASIC authentication dialog box prompts Users for their login information

Parameter	Description	Usage Note
securecontrol.plugin.login_home_location	Specifies the location of the HTML page ClearTrust SecureControl issues Users after a successful authentication.	
securecontrol.plugin.logout_form_location	Specifies the location of the HTML page ClearTrust SecureControl issues Users after they log out	
securecontrol.plugin.login_form_location_basic	Specifies the location of the HTML page ClearTrust SecureControl issues Users for BASIC authentication	
securecontrol.plugin.login.error_user_location_basic	Specifies the location of the HTML page ClearTrust SecureControl issues Users when they submit an invalid username for BASIC authentication	
securecontrol.plugin.login.error_pw_location_basic	Specifies the location of the HTML page ClearTrust SecureControl issues Users when they submit an invalid password for BASIC authentication	
securecontrol.plugin.login.error_unknown_location_basic	Specifies the location of the HTML page ClearTrust SecureControl issues Users when errors other than an invalid username or password (i.e. server errors) occur during BASIC authentication	
securecontrol.plugin.login_form_location_securid	Specifies the location of the HTML page ClearTrust SecureControl issues Users for SecurID authentication	
securecontrol.plugin.login.error_location_securid	Specifies the location of the HTML page ClearTrust SecureControl issues Users when an error has occurred during SecurID authentication	
securecontrol.plugin.login_form_location_nt	Specifies the location of the HTML page ClearTrust SecureControl issues Users for NT authentication.	
securecontrol.plugin.login_error_location_nt	Specifies the location of the HTML page ClearTrust SecureControl issues Users when an error has occurred during NT authentication	
securecontrol.plugin.password_expired	Specifies the location of the HTML page ClearTrust SecureControl issues when a User's password has expired	
securecontrol.plugin.login_form_location_ldap	Specifies the location of the HTML page ClearTrust SecureControl issues Users for LDAP authentication.	

Parameter	Description	Usage Note
securecontrol.plugin.login_error_user_location_ldap	Specifies the location of the HTML page ClearTrust SecureControl issues when User enters an invalid username during LDAP authentication..	
securecontrol.plugin.login_error_pw_location_ldap	Specifies the location of the HTML page ClearTrust SecureControl issues when User enters an incorrect password during LDAP authentication	
securecontrol.plugin.login_error_unknown_location_ldap	Specifies the location of the HTML page ClearTrust SecureControl issues when an error occurs during LDAP authentication	

Sample Web Server Plug-in Configuration File

The ClearTrust SecureControl Plug-in configuration file (Default.conf) contains all the configuration information for the ClearTrust SecureControl Plug-in. The following is a sample Plug-in configuration file.

```
// SecureControl Plugin Configuration File

// single sign on
securecontrol.plugin.sso=yes
securecontrol.plugin.session_lifetime=8 Hrs
securecontrol.plugin.idle_timeout=15 Mins
securecontrol.plugin.auto_challenge=yes
securecontrol.plugin.fudge_factor=5 Mins
securecontrol.plugin.secure=no
securecontrol.plugin.cookie_domain=securant.com
securecontrol.plugin.path=/
securecontrol.plugin.cookie_expiration=0 mins
securecontrol.plugin.cookie_ip_check=yes

// authorization server
securecontrol.plugin.auth_server_timeout=5 Secs
securecontrol.plugin.auth_server_mode=STANDARD
```

```
// server dispatcher
securecontrol.plugin.dispatcher_host=dino.securant.com
securecontrol.plugin.dispatcher_port=5608
securecontrol.plugin.dispatcher_timeout=10 Secs
securecontrol.plugin.key_port=5606

// web server
securecontrol.plugin.web_server_name=tech_pubs
securecontrol.plugin.realm=SCRealm
securecontrol.plugin.default_auth_mode=BASIC

// ssl
securecontrol.plugin.ssl.use=NO

// multi auth and form based configurations
securecontrol.plugin.form_based_enabled=yes

securecontrol.plugin.auth_resource_list=/*=BASIC

securecontrol.plugin.securid_sdconf_file_loc=/

securecontrol.plugin.login_home_location=/index.html
//securecontrol.plugin.login_home_location=/personalization/
cgi-bin/PersonalizationAccess.cgi
securecontrol.plugin.logout_form_location=/securecontrol/ct_
logout.html
securecontrol.plugin.login_form_location_basic=/securecontro
l/ct_logon.html

securecontrol.plugin.login_error_user_location_basic=/secure
control/ct_logon_bad_user.html
securecontrol.plugin.login_error_pw_location_basic=/secureco
ntrol/ct_logon_bad_pw.html
```

```
securecontrol.plugin.login_error_unknown_location_basic=/sec
urecontrol/ct_logon_unknown.html


securecontrol.plugin.login_form_location_securid=/securecont
rol/ct_logon_securid.html

securecontrol.plugin.login_error_location_securid=/securecon
trol/ct_logon_error_securid.html


securecontrol.plugin.login_form_location_nt=/securecontrol/c
t_logon_nt.html

securecontrol.plugin.login_error_location_nt=/securecontrol/
ct_logon_error_nt.html

securecontrol.plugin.login_error_password_expired=/securecon
trol/ct_change.html


securecontrol.plugin.login_form_location_ldap=/securecontrol
/ct_logon_ldap.html

securecontrol.plugin.login_error_user_location_ldap=/securec
ontrol/ct_logon_bad_user_ldap.html

securecontrol.plugin.login_error_pw_location_ldap=/securecon
trol/ct_logon_bad_pw_ldap.html

securecontrol.plugin.login_error_unknown_location_ldap=/secu
recontrol/ct_logon_unknown_ldap.html


securecontrol.plugin.login_form_location_custom=/securecontr
ol/ct_logon_custom.html

securecontrol.plugin.login_error_user_location_custom=/secur
econtrol/ct_logon_bad_user_custom.html

securecontrol.plugin.login_error_pw_location_custom=/securec
ontrol/ct_logon_bad_pw_custom.html

securecontrol.plugin.login_error_unknown_location_custom=/se
curecontrol/ct_logon_unknown_custom.html
```

Web Server Reference

Apache Web Server Notes

Using Apache as a Proxy Server

If you want to use Apache as a proxy server, you need to enable Apache's proxy module and make sure it is configured before you install the Plug-in.

To enable the proxy module:

1. Uncomment the following statement in `src/Configuration`:
2. AddModule modules/proxy/libproxy.a
3. Re-compile Apache. For example:

   ```
   cd /opt/apache_1.3.9/src
   ./Configure
   make
   ```

4. Copy the new httpd to where you installed your Apache, for example:

   ```
   cd /usr/local/apache/bin
   cp -p httpd httpd-ORIG
   cp /opt/apache_1.3.9/src/httpd
   ```

   Test your proxy server by choosing some hosts and setting them up as Web Servers.
5. Choose some hosts and set them up as Web Servers.
6. Turn proxy on (see the sample httpd.conf file on the ClearTrust CD)

   cd /usr/local/apache/conf

   edit httpd.conf to enable proxy, for example:

   Proxy Server directives. Uncomment the following lines to enable the proxy server:

   ```
   <IfModule mod_proxy.c>
   ProxyRequests On
   Proxy setting examples
   ProxyPass /hr/http://host1.domain.com/
   ProxyPass /finance/http://host2.domain.com/
   ProxyPass /eng/http://host3.domain.com
   <Directory proxy:*>
   Order deny,allow
   Deny from all
   ```

```
Allow from .your_domain.com
<Directory>
Enable/disable the handling of HTTP/1.1 “Via:” headers.
(“Full adds the server version; “Block” removes all
outgoing Via: headers)
Set to one of Off|On|Full|Block
ProxyVia On
```

If you also wish to enable caching on the proxy server, you must uncomment the relevant lines in the Configuration file, and provide the correct pathnames. You need a CacheRoot location, or caching won’t be enabled:

```
CacheRoot “@@ServerRoot@@/proxy”
CacheSize 5
CacheGcInterval 4
CacheMaxExpire 24
CacheLastModifiedFactor 0.1
CacheDefaultExpire 1
NoCache a_domain.com another_domain.edu
joes.garage_sale.com
</IfModule>
End of proxy directives.
```

7 Re-start Apache. For example:

```
cd /usr/local/apache/bin
./apachectl stop
./apachectl start
```

8 Test your proxy server. For example, attempt to access

`http://your_apache_proxy_server_domain.com/hr`

from a browser. You should be able to see the home page at `host1.domain.com`

ClearTrust SecureControl Web Server Plug-in Configurations 4of 4. If you are using Apache as a proxy server, add the following three directives within proxy settings’ “Directory” directive. For example:

```
<Directory proxy:*>
order deny,allow
Deny from all
Allow from .your domain.com
AuthType Basic
Require valid-user
AuthName SCRealm
</Directory>
```

Apache Web Server and SSL

If you want to use Apache as a secure server, you need to apply the Apache-SSL patch and make sure it is configured properly **before** you install the Plug-in.

You need the following software to configure a secure Apache server:

- Apache-SSL patch (http://www.apache-ssl.com)
- OpenSSL (http://www.openssl.org)
- patch (ftp://ftp.gnu.org/gnu)
- perl (http://www.sunfreeware.com)

9 Set `APACHE_SSL_EXPORT_CERTS` to TRUE in `src/include/buff.h` before you re-compile Apache and build httpsd.

For secure Apache server, set “SSLVerifyClient” to 1, and add “SSLExportClientCertificate” directive to <Directory “/usr/local/apache/htdocs”>...</Directory>.

If you want to use the SecureControl Plug-in for a secure Apache server—one that implements OpenSSL—be sure the secure Apache server works before attempting to incorporate the ClearTrust SecureControl module.

Here’s a summary of the configuration changes you must make:

1 Use a text editor to modify Configuration-HTTPS and add the EXTRA_CFLAGS and AddModule directives for the ClearTrust module:

2 Open the source code for the module (mod_ct_auth.c) and remove the references to the ssl and cryptography libraries from the LIBS section of the file.

3 Find the library files `libssl.a and libcrypto.a` in the ct_auth\lib sub-directory and rename them:

`cd/opt/`*`apache_version`*`/src/modules/ct_auth/lib`

4 Edit mod_ct__auth.c to remove “-lssl -lcrypto” from the LIBS

5 Recompile the Apache Web server:

```
cd /opt/apache_1.3.x/src
./Configure
make
```

Protecting Servlets (Apache JServ)

If your Apache server is running JServ and you want to protect the servlets:

6 Stop the Apache server.

7 Include the jserv.conf after the “CTPluginRoot” directive.

8 Add a “Location” block after the “Include” line to jserv.conf for the servlet zones you want to protect with the following directives:

```
AuthType Basic
Require valid-user
AuthName SCRealm
```

For example:

```
<Location /servlets>
AuthType Basic
Require valid-user
AuthName SCRealm
</Location>
```

9 Restart the Apache server.

Environment Variables

You can access two different environment variables in CGI scripts to identify the authenticated user:

On the Apache Server, use **REMOTE_USER**.

On the Apache Proxy Server, use **HTTP_REMOTE_USER**.

Disabling the Plug-in

As you can with any other Apache modules, you can disable the SecureControl Plug-in by commenting out all references to the Plug-in in the Apache configuration file, recreating the makefile and recompiling the server, and then commenting out all SecureControl directives in the httpd.conf.

To disable the ClearTrust SecureControl Web Server Plug-in:

1 Comment out the ClearTrust module name in src/Configuration:

```
# AddModule modules/ct_auth/mod_ct_auth.o
```

2 Re-compile the Apache Server:

```
cd /opt/apache_1.3.9/src
./Configure
make
```

3 Comment-out all the SecureControl directives in httpd.conf.

Microsoft Internet Information Server (IIS)

About Microsoft FrontPage

FrontPage resources cannot be protected using ClearTrust SecureControl Web Server Plug-in.

Modifying Microsoft IIS Basic Authentication Settings

During the installation of the Plug-in, the installer prompts you to allow the installer to make the change to Microsoft IIS configuration, but if you didn't enable this to happen at the time, you can change the settings later by following these instructions.

To modify the IIS settings manually:

1. Right-click on the icon for the Web Server you are protecting and select Properties from the pop-up menu. The Microsoft Management Console displays.
2. Go to the Directory Security pane of the Properties dialog.
3. Click Edit to open the Authentication Methods dialog box. You'll see three checkboxes: Allow Anonymous Access, Basic Authentication, and Windows NT Challenge/Response.
4. De-select and select the checkboxes as follows:
   - Select Allow Anonymous Access and Basic Authentication
   - De-select Windows NT Challenge/Response.

Netscape Enterprise Server

Sample Netscape on Solaris obj.conf

```
#--- start of file obj.conf ---------
# Netscape Communications Corporation - obj.conf
# You can edit this file, but comments and formatting changes
# might be lost when the admin server makes changes.
Init fn="flex-init" access="/opt/ns-home/https-davetest/logs/access" \

format.access="%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%] \"%Req-
>reqpb.clf-request%\" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-
length%"

Init fn="load-types" mime-types="mime.types"

# Here is the first ClearTrust directive.  This tells the server about the
# functions that implement the ClearTrust Plug-in.  The value of the shlib
# parameter should be the full pathname of the shared object file containing
```

```
# the ClearTrust Plug-in code.  The directory containing this file must also
# be in the environment variable LD_LIBRARY_PATH.  The value of the funcs
# parameter should not be changed.

Init fn="load-modules" \

funcs="ct-nsapi-init,ct-pre-process,ct-post-process,ct-nsapi-user, ct-auth,
\
ct-check-redirect,ct-nsapi-check-auth,ct-ssi-auth" \
shlib="<install_root>/lib/ct-nscp201sol-plugin.so"

# This directive is an optional setting if you are using the ClearTrust Web
based administration.
# The HTML administration uses cgis.  The cgis use the ClearTrust API to
perform the administration.
# Consequently, the cgis need ClearTrust API configuration information.  The
configuration information
# is located in the cgi.conf file
# CT_CGI_PROPFILE: The environment variable defining the location of the
cgi.conf file
Init fn="init-cgi" CT_CGI_PROPFILE="<install_root>/conf/cgi.conf"

# ClearTrust directive 2. This directive tells the Web Server where the
Plug-in is installed so that it can read in configuration information.
Init fn="ct-nsapi-init" plugin_root="<install_root>"

<Object name="default">

# ClearTrust directives 3, 4, 5, and 6.  These should both appear verbatim;
no changes
# to the parameters are necessary.  These directives drive the standard HTTP
# basic authentication process, and permit ClearTrust to use the user name and
# password thus obtained.
#The value of the realm parameter specifies the name of the HTTP
authentication realm of resources protected by this Web #server, and can be
set at your descretion. (Note that this is NOT the same as the SecureControl
notion of Realm; it merely #affects what users will see in their Web browser's
authentication dialog box.)

AuthTrans fn="ct-pre-process"
AuthTrans fn="ct-ssi-auth"
AuthTrans fn="ct-auth" realm="CTRealm"


# ClearTrust directive 7.  This should appear verbatim; no changes
# to the parameter is necessary.  The directives determines whether the server
# needs to serve up a page different from the one requested by the browser
NameTrans fn="ct-check-redirect"

# ClearTrust directive 8 & 9.  Directive 8 define the location of the HTML
files
# for form based authentication, as well as the HTML files for the Web based
administration.
# Directive 9 defines the location of the ClearTrust Web based administration
```

```
cgis.
NameTrans fn="pfx2dir" from="/securant"
dir="/export/home/Plugins/securecontrol-docs"
NameTrans fn="pfx2dir" from="/securecontrol-cgis" name="cgi" \
dir="/export/home/Plugins/securecontrol-cgis"

NameTrans fn="pfx2dir" from="/ns-icons" dir="/opt/ns-home/ns-icons"
NameTrans fn="pfx2dir" from="/mc-icons" dir="/opt/ns-home/ns-icons"

PathCheck fn="unix-uri-clean"
PathCheck fn="find-pathinfo"
PathCheck fn="find-index" index-names="index.html,home.html"

# ClearTrust directive 10 & 11.  This is where we tell the server to use
ClearTrust
# for authorization of HTTP requests.
PathCheck fn="ct-nsapi-check-auth"
PathCheck fn="ct-post-process"

ObjectType fn="type-by-extension"
ObjectType fn="force-type" type="text/plain"

Service fn="imagemap" method="(GET|HEAD)" type="magnus-internal/imagemap"
Service fn="index-common" method="(GET|HEAD)" type="magnus-
internal/directory"
Service fn="send-file" method="(GET|HEAD)" type="*~magnus-internal/*"

# ClearTrust directive 12.  This is the last ClearTrust entry.  It calls the
plugin’s error
# handling if an error occurred during the processing.
Error fn="ct-post-process"

AddLog fn="flex-log" name="access"

</Object>

<Object name="cgi">
ObjectType fn="force-type" type="magnus-internal/cgi"
Service fn="send-cgi"
</Object>

#--- end of file obj.conf ---------
```

Appendix C
Database Reference

The ClearTrust SecureControl solution includes a relational database management system (RDBMS) that ensures the integrity of its internal data and increases security. Properly configured, this full-powered RDBMS enables users to:

- Maintain multiple database servers for redundancy. Multiple database servers can be configured to automatically synchronize their data with one another to ensure integrity.
- Restore the database to its state at a certain point in time. This can be used, for example, to confirm a particular user's permissions.
- Store data in an existing database if the organization already maintains several database servers.

Maintaining and securing the database is critical to the successful operation of the ClearTrust SecureControl solution. Database maintenance processes and procedures vary from organization to organization, and most large organizations have dedicated database administration groups to take care of the RDBMS-specific tasks.

In this section you'll find information about:

- Database Maintenance
- Migrating Data on Solaris

Installation and Configuration Settings Reference

These values are recommended by Oracle Corporation for Oracle7 Release 7.3.4 running on Solaris. See the platform-specific Oracle Installation and Configuration Guide, available on Oracle Technology Network (http://technet.oracle.com) for complete information.

TABLE 9-1: Shared Memory Values for Oracle

Parameter	Recommended value	Description
SHMMAX	32769156	Maximum size (in bytes) of a single shared memory segment
SHMMIN	1	Minimum size (in bytes) of a single shared memory segment
SHMMNI	100	Number of shared memory identifiers
SHMSEG	20	Maximum number of
SEMMNS	200	Number of semaphores in the system
SEMMNI	100	Number of semaphore set identifiers in the system.

These values are recommended by Sybase for Sybase Adaptive Server Enterprise 11.2.9 running on Solaris. See *Installing Sybase Adaptive Server Enterprise on Sun Solaris 2.x (SPARC),* available from Sybase at http://www.sybase.com/TK_Complete_url for complete information.

TABLE 10-1: Shared Memory Values for Sybase

Parameter	Recommended value	Description
SHMMAX	32769156	Maximum size (in bytes) of a single shared memory segment
SHMMIN	1	Minimum size (in bytes) of a single shared memory segment
SHMMNI	100	Number of shared memory identifiers
SHMSEG	20	Maximum number of
SEMMNS	200	Number of semaphores in the system
SEMMNI	100	Number of semaphore set identifiers in the system.

Database Maintenance

Maintaining and securing the database is critical to the successful operation of the ClearTrust SecureControl solution. Database maintenance processes and procedures vary from organization to organization. Most large organizations have dedicated database administration groups to take care of maintenance and other database-related tasks.

ClearTrust SecureControl includes an embedded Oracle database (release 7.3.4).

but also supports Sybase if preferred.

The following maintenance discussion regarding backups, however, pertains only to Oracle.

Your organization must implement a comprehensive database maintenance program suited to its specific needs.

File Structure and Backups

On Windows NT systems, all Oracle files are installed in the ORANT directory under the Securant install directory. On Solaris systems, the location for storing files depends on the directory name specified during the Oracle install process. Database files are in the DATABASE directory below the ORANT (or Solaris) directory. The initialization file is named initorcl.ora.

Backups are necessary to protect against machine outages, disasters, and so on. Because backups are so crucial, some organizations keep two sets of backups and store one off-site for added protection.

Database backups are different than regular data backups because the database data on the disk is constantly changing while the database is operating. There are two main techniques for backing up the database: cold and hot.

Cold Backups

Cold backups are relatively simple and don't require monitoring. In a cold backup, the entire database is shut down and inaccessible during the backup procedure. To perform a cold backup, you simply shut down the database, backup the files using whatever tools you choose, and then restart the database.

In a cold backup then, it is also necessary to shut down all SecureControl processes, which disables access control for a period of time. One technique to minimize this downtime is to copy the data files elsewhere on the disk and then copy those to tape after the database has been restarted.

Hot Backups

Hot backups are a little more complicated. The advantage of a hot backup is that the database is available for use during the backup process. The disadvantage is that a hot backup requires more maintenance and configuration.

There are two parts to a hot backup: a full database dump and a set of archive log files that record every change made to the database. The full database dump is the same as in the cold backup yet data may not be as consistent because the database is still operating. This is why log files are necessary; they allow you to bring the database into a consistent state.

To configure the database for a hot backup, you need to:

- Turn on the ARCHIVELOG configuration option in the database initialization file.
- Specify a location for the log files and monitor that directory to make sure it doesn't run out of disk space.

Refer to one of the following guides for more detailed instructions:

- "Oracle DBA Handbook" by Kevin Loney
- "Oracle Backup & Recovery Handbook" by Rama Velpuri
- "Oracle Database Administration" by Kreines and Lasky

Migrating Data on Solaris

These instructions presume you have an existing ClearTrust/SecureControl installation and that you wish to migrate the data from your existing Entitlements database to work with a new release. Securant provides several SQL scripts (based on the release of the ClearTrust from which you're migrating) to take care of the migration process. (Be sure to read the Release Notes for the ClearTrust SecureControl version you're installing before installing or migrating.)

1. Backup your Entitlements database. (Although the scripts don't alter existing data in the Entitlements database, you should always have a recent backup of the database.)
2. Obtain the script you need from Securant Technical Support (415-315-1591 or support@securant.com). You'll receive the script you need by email. Depending upon the version you're migrating from, the script will be one of these three:

```
migrate_3.5.1_to_4.0.sql
migrate_3.5.1_to_4.2.sql
migrate_4.0_to_4.2.sql
```

3. Copy the script in a temporary directory on the server where the Oracle database is located.
4. Telnet or login to the Solaris machine as the owner of the Oracle database. (If you setup your site using the example in the Installation and Configuration Guide, this will be the user oracle with the password you set during installation.)
5. Change to the directory that contains the migration script.
6. Start SQL*Plus and connect to the Entitlements database:

   ```
   sqlplus username/password
   ```

   where username matches the securecontrol.db.owner.name property and password matches the securecontrol.db.owner.password property in the ClearTrust default configuration (/Securant/SecCtrl/ct_root/conf/Default.conf) file.
7. At the SQL*Plus command prompt, enter the name of the individual script required for your particular migration to run the script. (For additional assistance in using one of these scripts, contact Securant Technical Support at 415-315-1591.) For example:

   ```
   sql>@migrate_3.5.1_to_4.0.sql
   ```
8. To exit SQL*Plus, type exit at the SQL*Plus

   ```
   sql>exit
   ```

Appendix D
Troubleshooting

Sample Log Reference

Entitlements Server (SecCtrl_DataServer_Output.log)

If the Entitlements Server successfully starts, message `SVR-00005-I` is issued. The `SecCtrl_DataServer_Output.log` should resemble:

```
SVR-00000-I:

ClearTrust Entitlements Server
Version 4 Release 2
Build: rel_FM_9_1

Copyright 2000. Securant Technologies.
http://www.securant.com

SVR-00004-I:

Install Type                : Standard install
Voyager Registration Name : EntitlementsServer
API Port                    : 5601
Database Name                  : ORACLE
Debug Enabled                  : false

SVR-00200-I: Loading the USERS from the database.
.SVR-00201-I: USERS loaded.
SVR-00200-I: Loading the GROUPS from the database.
.SVR-00201-I: GROUPS loaded.
SVR-00200-I: Loading the APP from the database.
.SVR-00201-I: APP loaded.
SVR-00005-I: Entitlement Server bootstrap completed.
```

Authorization Server (SecCtrl_Authorizer(*x*).log)

When an Authorization Server successfully starts, message `AUTH-00001-I` is issued. The `SecCtrl_Authorizer(x).log` (where *x* is the instance number, for example, 1, 2, 3…) should resemble:

```
AUTH-00000-I:

ClearTrust Authorization Server
Version 4 Release 2
Build: rel_FM_9_1

Copyright 2000. Securant Technologies.
http://www.securant.com

AUTH-00001-I:
TCP Listen Port              : 5615
Dispatcher Location          : blendo:5607
Entitlements Server Location : blendo

Using SSL                    : NO


Database Instance            :
Number Database Connections  : 1
Number Stmt. Per Connection  : 1

Master Cache Size             : 10000
User Cache Size              : 5000
```

Dispatcher (CT_Dispatcher.log)

When the Server Dispatcher successfully starts, message `DISP-00001-I` is issued. The `CT_Dispatcher.log` should resemble:

```
DISP-00000-I:

ClearTrust Authorization Server Dispatcher
Version 4 Release 2
Build: rel_FM_9_1

Copyright 2000. Securant Technologies.
http://www.securant.com
```

```
DISP-00001-I:
Auth Server Registration Port : 5607
Auth Server Listener Port     : 5608
Key Server Port               : 5606

Successful server bootstrap completed.
URL Cache Size                : 5000
Smart Rule Cache Size         : 500
User Property Cache Size      : 0
API Function Cache Size       : 500

Successful server bootstrap completed.
```

Troubleshooting Installation Problems

To verify the installation:

1. Open a new shell. If you haven't updated your .cshrc or .profile, you must source the secctrl_envs.csh (or secctrl_envs.sh) file as appropriate for your shell to ensure that your environment is properly initialized. These files can be found in the cleartrust/scripts sub-directory. For example:

   `source /cleartrust/scripts/secctrl_envs.sh`

2. Start all ClearTrust server processes by issuing the `secctrl start` command.
3. Open the log files located in the <securant>/securecontrol/<db_instance>/logs directory and review the entries in each log. See the table below for the relevant message information in each log.

ClearTrust Services Log Messages

Component	Log	Message
Entitlements Server	SecCtrl_DataServer_Output.log	SVR-00005-I Entitlement Server bootstrap completed.
Authorization Server	SecCtrl_Authorizer(*x*).log	AUTH-00001-I
Server Dispatcher	CT_Dispatcher.log	DISP-00000-I Successful server bootstrap completed

TABLE 10-2: Troubleshooting Startup Error Messages

Component	Symptom	Problem	Solution
secctrl start	“No such file or directory” error message	Path problem or incorrectly set environment variable	
Entitlements Server	CT_DataServer_Output.log contains failure message: SVR-00004-I:.. fatal: lib..open failed	LD_LIBRARY_PATH not properly set.	Set the path: source secctrl_envs.sh
Entitlements Server or Authorization Server	Error message SVR-00022-E: Unknown failure during initialization...java.sql.		

In this section you’ll find information about:

- Troubleshooting the Installation
- Uninstalling ClearTrust Components

Troubleshooting Solaris Installations

Table 1-2 provides solutions to problems you might encounter while attempting to install ClearTrust SecureControl on Solaris OS.

Error Message

Table 8-1. Troubleshooting Solaris Installations

Problem	**Solution**
When you issue the secctrl start command, a message similar to this appears: `Securant ClearTrust 4.2 Services starting /opt/securant/cleartrust/scripts/secctrl: /cleartrust/scripts/startup: No such file or directory`	Set the appropriate environmental variables. From the `<securant_root>/cleartrust/bin` directory, issue: **`source secctrl_envs.sh`**

Table 8-1. Troubleshooting Solaris Installations (Continued)

Problem	Solution

The EntitlementsServer fails to start. The `CT_DataServer_Output.log` contains:
`SVR-00004-I:`
`Install Type: Standard install`
`Voyager Registration Name: EntitlementsServer`
`API Port: 5601`
`Database Name: CT`
`Debug Enabled: false`
`ld.so.1:`
`/opt/securant/jre/bin/sparc/native_threads/jre:`
`fatal: libweblogicoci34.so: open failed: No such file or directory (libweblogicoci34.so`
	The `LD_LIBRARY_PATH` is not set properly. Issue the following command:
	`source secctrl_envs.sh`

When you start the Entitlements Server or the Authorization Server, the following message appears:
`SVR-00022-E: Unknown failure during initialization.`
`Details:`
`java.sql.SQLException: System.loadLibrary threw java.lang.UnsatisfiedLinkError with the message 'no weblogicoci31 in shared library path'.`
`Stack Trace:`
`java.sql.SQLException: System.loadLibrary threw java.lang.UnsatisfiedLinkError with the message 'no weblogicoci34 in shared library path'.`
`at weblogic.jdbc.oci.Driver.connect(Driver.java:119)`
`at java.sql.DriverManager.getConnection(DriverManager.java:91)`
`at sirrus.util.db.ConnectionManager.openConnection (ConnectionManager.java:41)`
`at sirrus.entitlements.db.dbkona.Bootstrap.bootstrap(Bootstrap.java:11`
`at sirrus.entitlements.db.dbkona.Bootstrap.main(Bootstrap.java:257)`
	The `LD_LIBRARY_PATH` is not set properly. Issue the following command:
	`source secctrl_envs.sh`

Table 8-1. Troubleshooting Solaris Installations (Continued)

Problem	Solution

When starting the Entitlements Server or the Authorization Server, the following message appears:
`SVR-00022-E: Unknown failure during initialization.`
`Details:`
`java.sql.SQLException: ORA-12203: TNS:unable to connect to destination - (CT_ADMIN/********@ISA)`
`Stack Trace:`
`java.sql.SQLException: ORA-12203: TNS:unable to connect to destination - (CT_ADMIN/********@ISA)at weblogic.db.oci.OciConnection.getLDAException(OciConnection.java:99)`
`at weblogic.jdbc.oci.Driver.connect(Driver.java:160)`
`at java.sql.DriverManager.getConnection(DriverManager.java:91)`
`at sirrus.util.db.ConnectionManager.openConnection(ConnectionManager.java:41)`
`at sirrus.entitlements.db.dbkona.Bootstrap.bootstrap(Bootstrap.java:110)`
`at sirrus.entitlements.db.dbkona.Bootstrap.main(Bootstrap.java:257)`
	The Oracle instance or the TNS Listener has not been started. Consult with your Oracle Administrator for information about how to start these services.

Entitlements Server or Authorization Server receive “Out of Memory Exception”. Or the API Client or LDAP Replicator receive “Should Not Occur Error Out of Memory Exception”	Jave heap size is too low. On Solaris systems, you can increase the Java heap size by editing the individual server scripts in the root directory of the installation: `/opt/securant/cleartrust/scripts/secctrl: /cleartrust/scripts/startup.` These scripts contain a parameter with the value of 64M (which means that 64MB of RAM is being used by that Java service).

BY DEFAULT, THE JAVA ENVIRONMENT for servers is started with a maximum heap size. This heap size may be too low in some situations or for a site with a large number of users. If there is a lot of memory on your machine(s) and your site has a large user base, you should change this default setting accordingly.

Troubleshooting Windows NT Installations

Table 1-1 provides solutions to problems you might encounter while installing or running ClearTrust SecureControl on Windows NT.

Table 8-2. Troubleshooting Windows NT Installations

Problem	Solution
When you check the Service Panel, not all the ClearTrust SecureControl services start up.	Start the ClearTrust SecureControl servers as processes: Stop all ClearTrust SecureControl services, execute **Start->Programs->ClearTrust->Stop All Services**. Start the Oracle service from the Services Control Panel by selecting **OracleStartORCL** and clicking **Start** Start each ClearTrust process by executing **Start->Programs->ClearTrust->ClearTrust Server Processes->**<**process**>. As each process starts, it will issue messages indicating whether it succeeded or failed. If the process failed to start it will also generate a reason why the component failed to start. This message should allow you (or your ClearTrust SecureControl technical representative) to diagnose the problem.
When you start the ClearTrust SecureControl Server, it fails, but the window disappears too quickly for you to read the error message.	Start the ClearTrust SecureControl servers manually from the command prompt. The batch files that start the various processes are located in the `Securant\SecCtrl\bin` directory. Or, you can add 'Pause' at the end of the .bat file.

Table 8-2. Troubleshooting Windows NT Installations

Problem	Solution
During installation of the ClearTrust SecureControl Server you did not enter the ClearTrust license information. How can you update this information?	Edit the `Default.conf` file in `Securant\SecCtrl\ct_root\conf` directory. Update the two entries, `securecontrol.license.info` and `securecontrol.license.key`, with the license information
Entitlements Server or Authorization Server generates “Out of Memory Exception”. Or the API Client or LDAP Replicator generates “Should Not Occur Error Out of Memory Exception”	Java heap size is too low. On NT systems, you can increase the Java heap size in one of two ways. You can either edit the .bat files that start the server services. The .bat files are located in the `Securant\SecCtrl\bin` directory. You can also increase the Java heap by editing the values in the registry if you are using the NT services to start the product. The registry locations that contain the values for NT services are: hklm\system\currentcontrolset\services\javaserviceauth00\parameters\options= hklm\system\currentcontrolset\services\javaserviceauth01\parameters\options= hklm\system\currentcontrolset\services\javaservicedispatch\parameters\options= hklm\system\currentcontrolset\services\javaserviceeserver\parameters\options= "-noclassgc -ms32M -mx64M -DLISTEN_PORT=5617 -DVOYAGER_PORT=5618 -DCT_ROOT=d:\Securant\SecCtrl\ct_root" Here, you can change the 64M (which means that 64MB of RAM is being used by that service) to something appropriate for your site. note: hklm= hkey_local_machine

By default, the Java environment for servers is started with a maximum heap size which is limited according to default values. This heap size of 64MB may be too low in some situations or for a site with a large number of users. If your server has a lot of memory and your site has a large user base, you should change this default setting accordingly.

Troubleshooting ClearTrust Web Server Plug-in Installations

Table 8-3 provides solutions to problems you might encounter while installing ClearTrust SecureControl Plug-ins.

Table 8-3. Troubleshooting ClearTrust SecureControl Plug-in Installations

Problem	Solution
I receive the message `Single Sign On has successfully initialized`, but my browser does not seem to be receiving a cookie when I authenticate	Verify that the `CookieDomain` parameter in the `Default.cfg` file contains the correct domain. For a URL residing on a server located at `blendo.securant.com`, the CookieDomain should be set to `securant.com`. Also, make sure that your browser is set to accept cookies. If it is not, Single Sign-On will not work.
All my Web Servers have Single Sign-On successfully initialized, yet Single Sign-On does not always seem to be working between the various machines.	If the Web Servers are residing on different machines, make sure that the difference between the machine clocks are within the `FudgeFactor` values specified within all the `Defualt.cfg` files. Additionally, make sure that the CookieDomain is the same for all the Web Servers
I've protected a resource in ClearTrust SecureControl, yet when I request the URL I am not challenged. What is wrong?	Make sure that the name of the Web Server defined in ClearTrust SecureControl matches exactly the web_server_name parameter in the Default.conf file.

Troubleshooting Single Sign-On Initialization

If you are using the Single Sign-On feature of ClearTrust SecureControl, you should also verify that it has been successfully initialized. Successful initialization of Single Sign-On is indicated by the line:

```
Single Sign On has successfully initialized
```

close to the end of the Solaris standard output display or the Windows NT DBWin32 debug tool display.

Table 10-3 provides solutions to problems you might encounter when the Single Sign-On initialization failed.

TABLE 10-3: Troubleshooting Single Sign-On Initialization

Error Message	Problem	Solution
coca: coca_update_keys, try 0: connect() errno 146 error: can't update session key/s Single Sign On has been disabled.	The Server Dispatcher is not running, and the Plug-in is unable to contact the key server to initialize Single Sign-On.	Start the Server Dispatcher, then stop and restart the Web Server.

Error Message	Problem	Solution
coca: coca_newkey_protocol: parse_reply() code 255 coca: coca_newkey_protocol: parse_reply() code 255 error: can't update session key/s Single Sign On has been disabled.	The secret key on the Web server isn't a correct match to that on the SecureControl server.	Re-export the secret key from the SecureControl server and re-install on the Web server. See "Support for Single Sign-on (SSO)" on page 57 for details.
error: config read error Single Sign On has been disabled Running in "standard" auth server mode.	The Plug-in is unable to read the Single Sign-On configuration files.	Check the correct directory to be sure it is configured correctly and that the directory and its files have the appropriate file permissions for the Web Server to access them.
	Web Server failed during startup.	Contact Securant Technologies for technical support.
Can't find nsldapssl32v30.dll or libldapssl30.so.	These files were not correctly installed from the the ClearTrust SecureControl CD, or the path was set incorrectly.	On NT systems: Copy nsldapssl32v30.dll to c:\winnt\system32 On Solaris systems: Copy libldapssl30.so to <securecontrol-plugin-root>/lib; Add the path where libldapssl30.so is to LD_LIBRARY_PATH of the server's startup script.
	Single Sign-On does not seem to work.	Make sure you created a client key on the ClearTrust SecureControl Server and put the key in <securecontrol-plugin-root>/KeyClient/KeyClient.sec file; Make sure all of the machines have the same system time; Turn off warning for cookie.
	LDAP authentication always fails.	Refer to the debug message and check the LDAP server's Access Log.

Error Message	Problem	Solution
"*** socket error while talking to CT server dispatcher at warrior:5608"	A network socket layer communications error when accessing a URI.	Re-start the Dispatcher/Authorizer.Start the Dispatcher/Authorizer; Stop the Web Server (stop); Start the Web Server (start).
Unable to update session key. Make sure the Dispatcher is running and the plugin has been registered with the Dispatcher.	The Dispatcher/Authorizer has not been started.	Start the Dispatcher/Authorizer; Stop the Web Server (stop); Start the Web Server (start).
	The Web Server does not function properly with the sample obj.conf file.	Contact Securant Technologies for technical support.

Web Server Error Messages

If your browser displays an error message from the Web Server, see the Web Server's error log to determine the cause. Problems are usually caused if you started up the ClearTrust SecureControl servers in the wrong order, or if the servers haven't been started at all.

The examples given here are taken from the error log of the Netscape Enterprise server. Table 10-4 shows two error messages you might see in a Netscape Enterprise Server error log; other web servers may produce similar error messages. The short answer is "restart your ClearTrust SecureControl Servers."

TABLE 10-4: Web Server (Netscape) Log Messages

Symptom	Root cause	Solution
[08/Jul/2000:18:06:17] failure: for host sparky.securant.com trying to GET /, ct_handle_request reports: *** socket error while talking to CT server dispatcher at sparky:5608 ***	The Server dispatcher isn't running	Restart all ClearTrust SecureControl Servers.
[08/Jul/2000:18:13:04] info: for host sparky.securant.com trying to GET /, ct_handle_request() reports: server pool #1 (size 0):	Server Dispatcher is running, but unable locate Authorization Server.	Restart all ClearTrust SecureControl Servers.

Uninstalling ClearTrust Components

Uninstalling any or all of the ClearTrust components is a simple operation usually involving just a menu selection.

ClearTrust SecureControl Servers

Uninstalling on Solaris

To uninstall ClearTrust SecureControl on Solaris CD, manually remove the Clear Trust Directories and the administrator's profile from the shell's resource script. You must also remove the instance of the Oracle database.

Uninstalling on Windows

To uninstall ClearTrust SecureControl from Windows NT:

- Select ClearTrust, then SecureControl, then Uninstall from the Windows NT menu to start the uninstaller program. The uninstaller program deletes all server-side components.
- Restart the system after the uninstaller has completed.

ClearTrust Web Server Plug-ins

Uninstalling the Netscape Plug-in on Windows NT

To uninstall the Netscape Plug-in from Windows NT:

- Select the Add/Remove Programs utility from the Control Panel.
- Click on the Install/Uninstall Tab.
- Scroll through the list of Programs and select ClearTrust Plug-ins. And then select Netscape.
- Click on the Add/Remove button.
- Click OK.
- Restart the system after the Plug-ins have been removed.

Uninstalling the Netscape Plug-in on Solaris

To uninstall the Netscape Plug-in from Solaris:

1. Remove any of the folders you have created to hold the Netscape Plug-in directories.

2 Undo any modifications you made to directories or folders when you installed the Plug-in.

After you have taken care of folders and directories, revert back to the backed up copy of the Obj.conf file as the existing copy was modified when you installed the Plug-in.

Uninstalling the Microsoft IIS Plug-in from Windows NT

To uninstall the Microsoft IIS Plug-in from Windows NT:

- Select the Add/Remove Programs utility from the Control Panel.
- Click on the Install/Uninstall Tab.
- Scroll through the list of Programs and select ClearTrust Plug-ins. And then select IIS.
- Click on the Add/Remove button.
- Click OK.
- Restart the system after the Plug-ins have been removed.

Uninstalling SecureControl Manager

Uninstalling SecureControl Manager on Solaris

In the directory where the ClearTrust SecureControl Manager is installed, type the command:

'./Uninstall_SecureControl_Mgr'

This starts an InstallShield program which will remove the ClearTrust SecureControl Entitlements Manager from the machine.

Uninstalling SecureControl Manager on Windows NT

To uninstall ClearTrust SecureControl Entitlements Manager:

- From the NT Control Panel, double-click the Add/Remove Programs option. A dialog will appear with a list of the applications which are installed on the machine.
- Select the item titled ClearTrust Entitlements Manager 4.0.
- Click Add/Remove to launch an InstallShield program that will remove ClearTrust SecureControl Entitlements Manager from your machine.

Uninstalling the LDAP Replicator Tool from Windows NT

To uninstall the LDAP Tool from Windows NT:
From the Start menu, go to ClearTrust SecureControl and then to LDAP Replicator and select Uninstall SecureControl LDAP Replicator.

Click on Uninstall to launch the UninstallAnywhere program. This removes the LDAP Tool from your machine.

Uninstalling the LDAP Replicatior Tool from Solaris

To uninstall the LDAP Replicatior Tool from Solaris:
Type the following command in the Uninstaller Data directory:

./Uninstall_LDAP_Replicator

This launches the UninstallAnywhere program which removes the ClearTrust SecureControl LDAP Replicator Tool from your machine.

Index