CA SiteMinder® Adapter for CA WebFort and CA RiskFort

Installation and Configuration Guide for Microsoft Windows

r2.1.6

Contents

Preface

This guide describes the process to install and configure Arcot Adapter 2.1.6 with CA SiteMinder®. This guide includes the following information:

- The high-level architecture of the integration process
- Installation requirements
- Installing Arcot Adapter
- Post-installation tasks
- Configuration files
- Uninstalling Arcot Adapter
- Arcot Forms Credential Collector (FCC) pages
- Support for Backing Authentication

Intended Audience

This guide is intended for system integrators who are responsible for integrating CA SiteMinder® with Arcot WebFort and Arcot RiskFort. This guide requires that the reader must be familiar with WebFort, RiskFort, and CA SiteMinder® authentication configuration, including custom authentication schemes and FCC pages.

Note: This guide assumes that CA SiteMinder®, Arcot WebFort, and Arcot RiskFort are installed and are independently operational.

Information Included in this Guide

This guide is organized as follows:

- Chapter 1, “Arcot Adapter for CA SiteMinder® Overview”, describes the high-level integration architecture of Arcot Adapter with CA SiteMinder and describes the other Arcot products that Adapter interacts with.
- Chapter 2, “Preparing for Installation”, lists the prerequisite software and configurations required to install Arcot Adapter.
- Chapter 3, “Installing Arcot Adapter”, describes the steps to install Arcot Adapter.
- Chapter 4, “Deploying and Configuring Token Server”, describes the steps to deploy and configure Token Server.
- Chapter 5, “Deploying and Configuring Arcot Customization Engine”, describes the steps to deploy and configure the Customization Engine.
- Chapter 6, “Configuring Arcot Shim and FCC Pages”, describes the steps to configure the FCC pages and Shim.
- Chapter 7, “Configuring the CA SiteMinder Policy Server”, describes the steps to configure CA SiteMinder® to use Arcot Adapter.
- Chapter 8, “Uninstalling Arcot Adapter”, lists the steps to uninstall the Arcot Adapter components and the database that is used by Arcot Adapter.
- Appendix A, “Configuring Backing Authentication Scheme”, describes the steps to configure support for external or third-party authentication schemes or mechanisms.
- Appendix B, “Third-Party Software Licenses”, lists the third-party software that are used with Arcot Adapter.
- Appendix C, “Glossary”, describes the terms that are used in this guide.

Related Publications

Related Arcot publications include:

Arcot WebFort 6.0 Installation and Deployment Guide	This guide provides the procedures for installing and deploying WebFort.
Arcot WebFort 6.0 Administration Guide	This guide provides the procedures for administering and configuring WebFort.
ArcotID Client Reference Guide	This guide provides information about ArcotID Clients.
Arcot RiskFort 2.0 Installation and Deployment Guide	This guide provides the procedures for installing and deploying RiskFort.
Arcot RiskFort 2.0 Administration Guide	This guide provides the procedures for administering and configuring RiskFort.

Conventions Used in This Book

The conventions and formats used in this guide are described in the following paragraphs:

Typographical Conventions

This guide uses the following typographical conventions:

Italic	Emphasis, Guide names
Bold	User input, GUI screen text
`Fixed`	File and directory names, extensions, command prompt, CLI text, code in running text
`Fixed Bold`	Target file or directory name in the path
`Fixed`	Command prompt, CLI text, code
`Fixed-Italic`	File or directory name that might be different from user to user
Link	Links within the guide, URL links

Formats

This guide uses the following formats to highlight special messages:

Note: Highlights helpful suggestions and references to materials not included in this guide.

Tip: Highlights a procedure that will save time or resources.

Warning: Ignoring this type of note may result in a malfunction or damage to the equipment.

Important: Information to know before performing an operation.

Caution: Makes the user attentive of the possible danger.

Book: Provides reference to other guides.

Contact CA

For your convenience, CA provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA products. At *http://ca.com/support*, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums

- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA product documentation, you can send a message to *techpubs@ca.com*.

If you would like to provide feedback about CA product documentation, complete our short customer survey, which is available on the CA Support website at *http://ca.com/docs*.

Chapter 1
Arcot Adapter for CA SiteMinder® Overview

CA SiteMinder provides centralized security management capability that enables customers, partners, and end users to securely access the web to deliver applications and data. Integrating SiteMinder with Arcot WebFort and Arcot RiskFort lets you protect your resources with the multifactor and risk-based adaptive authentication.

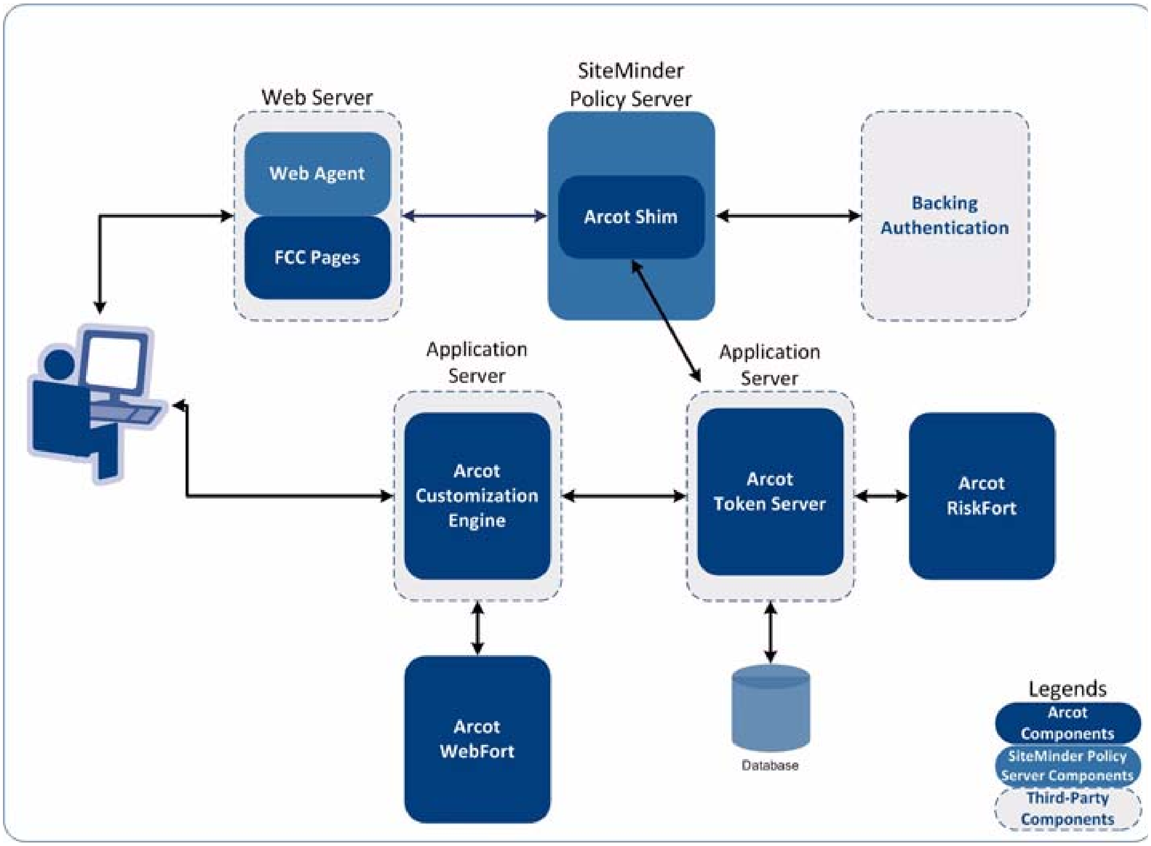
This chapter introduces you to the basic concepts of Arcot Adapter and covers the following sections:

- Arcot Adapter Architecture
- Arcot Adapter Features

Arcot Adapter Architecture

The following figure illustrates how Arcot Adapter and its components integrate with the SiteMinder Policy Server components.

Figure 1-1 Arcot Adapter Architecture Diagram



Note: If you plan to use a single database for Arcot products and other applications, Arcot recommends that you use a separate schema for Arcot products.

As illustrated in Figure 1-1, Arcot Adapter includes the following components:

- Arcot Token Server
- Arcot Customization Engine (ACE)
- Arcot Shim
- Arcot Form Credential Collector (FCC) Pages

- Arcot WebFort
- Arcot RiskFort

Arcot Token Server

Token Server is responsible for creating, maintaining, and tracking the tokens that are used to associate the authentication and risk status of a logon session across multiple Arcot and SiteMinder components. The tokens, which contain the information of the user and the session state, enable the other Arcot components to remain stateless.

Token Server also acts as a proxy to RiskFort by providing risk evaluation services to other Arcot components. Token Server receives the risk evaluation input parameters from the calling application and passes them to RiskFort. After the risk evaluation is complete, Token Server inserts the risk evaluation result into the token for further examination or for processing by other components.

Note: Based on the implemented workflow, risk evaluation can be performed *before* or *after* the user authentication. If the risk evaluation takes place after user authentication, the result of the authentication is persisted in the token and then the risk evaluation is performed.

Token Server also provides a token and validation mechanism to securely communicate the authentication result, risk result, and the subsequent action to be performed by Arcot Shim.

Arcot Customization Engine (ACE)

Arcot Customization Engine is the only component (along with FCC pages) of Arcot Adapter that a user interacts with directly. ACE is a state machine that guides the end user through the authentication process, risk evaluation process, or both by presenting JavaServer Pages (JSPs) to collect the user information required for authentication.

ACE also maintains the state data of the user workflow, conducts WebFort authentication, and reads or writes RiskFort Device ID information required by RiskFort.

Note: Not all user activities require user input. For example, risk assessment can be done without any user input.

ACE is shipped with the following *preconfigured authentication flows*:

- **ArcotID Authentication**: This includes ArcotID authentication using Arcot WebFort.

- **ArcotID and Risk Evaluation**: This authentication workflow is a combination of ArcotID and Risk evaluation (performed by Arcot RiskFort) workflows.
- **Basic Authentication and Risk Evaluation**: this authentication workflow combines the basic authentication scheme, which is configured in SiteMinder, and the risk evaluation workflow. The risk evaluation can be configured to either precede or succeed the basic authentication, thus, offering two different authentication workflows.

In addition to the four sample authentication workflows, ACE also provides you the capability to customize an authentication workflow. Also, you can configure multiple workflows at any time.

Arcot Shim

Arcot Shim which integrates with SiteMinder, acts as an interface between SiteMinder and the Arcot Adapter components (Token Server and ACE), and other Arcot products (WebFort and RiskFort).

Arcot Shim is an instance of a shared library and resides in the SiteMinder Policy Server instance. The Arcot Shim implements the SiteMinder Authentication API.

Arcot Form Credential Collector (FCC) Pages

Arcot FCC (referred to as FCC later in the guide) pages are static HTML pages used by Arcot Shim to collect user inputs during basic authentication and to display error messages, if any. These pages are deployed on the same web server where the SiteMinder Web Agent resides.

Arcot WebFort

Arcot WebFort protects users by providing strong, two-factor authentication, without changing their familiar username/password-based sign-on experience. As a result, it significantly enhances the varied authentication management capabilities (including step-up authentication) of a SiteMinder deployment by adding a transparent layer of strong multi-factor authentication.

Arcot RiskFort

Arcot RiskFort provides real-time protection against fraud during online transactions. Arcot RiskFort gathers data during the login process to track suspicious activities and formulates a Risk Score and Advice based on the organization's business rules and security protocols. The Risk

Advice then determines if the transaction is to be allowed or denied, whether a greater degree of authentication is required, or if the customer service or a network security personnel need to be notified.

> **Note:** Arcot WebFort and Arcot RiskFort are packaged separately and should be installed separately. Refer to the Arcot WebFort and Arcot RiskFort documentation for more information on these products.

Arcot Adapter Workflow

The following steps explain the procedure for user authentication and risk assessment of a transaction (refer to Figure 1-1):

1. The user accesses a resource that is protected by SiteMinder.
2. SiteMinder disambiguates the user.
3. If the authentication has to be performed by Arcot components, then the Arcot Shim redirects the user to Arcot WebFort.
4. The ACE guides the user through the authentication and risk evaluation processes.
5. Depending on the authentication and the risk evaluation results, the Arcot Token Server saves the user’s state in a token and securely communicates the authentication and risk result to Arcot Shim.
6. Finally, the Arcot Shim evaluates and forwards the authentication result to SiteMinder.

   If the user is authenticated successfully, the risk result is positive, and the user is authorized to access the protected resource, then the user is granted access to the protected resource.

Arcot Adapter Features

This section lists the key features of Arcot Adapter for SiteMinder:

- **Support for Arcot WebFort 6.0**

  Arcot Adapter 2.1.6 supports the latest Arcot WebFort 6.0 release. This release of WebFort includes many new features, which includes support for an LDAP directory server, support for two-way SSL communication, and the flexibility to add new authentication and issuance features.

  > **Book:** For more information about Arcot WebFort 6.0, see the WebFort documentation.

- **Support for Arcot RiskFort 2.0**

  Arcot Adapter 2.1.6 also supports Arcot RiskFort 2.0 release to evaluate risk of each incoming transaction.

  > **Book:** For more information about Arcot RiskFort 2.0, see the RiskFort documentation.

- **Improved Architecture**

  Arcot Adapter can be integrated with SiteMinder to provide strong authentication and evaluate transaction risk for SiteMinder users. It also provides the ability to customize the authentication workflows.

- **Integration with Backing Authentication Schemes**

  Arcot Adapter supports backing authentication schemes that are supported by SiteMinder. For this, the Shim acts as an interface between SiteMinder and the backing authentication scheme. The Shim forwards the authentication requests to the backing scheme. After performing the authentication, the backing authentication scheme sends the result back to Arcot Shim, which in turn is posted to SiteMinder by the Shim. In this case, Arcot Adapter can just be used for risk evaluation.

  Note: For more information about Backing Authentication schemes, see Appendix A, “Configuring Backing Authentication Scheme”.

- **Ability to Customize Workflows**

  Arcot Adapter provides the ability to customize the authentication workflows. This feature provides the SiteMinder customers the ability to use the strong, software-based authentication, and advanced risk management in conjunction with other authentication technologies supported by SiteMinder.

- **Support for Multiple Instances of Shim**

  You can deploy and configure multiple instances of Arcot Shim to support multiple authentication schemes. Each instance can be used to secure different resources.

- **Support for All ArcotID Client Types**

  Arcot Adapter supports all types of the ArcotID Client, which are used for strong authentication using WebFort.

- **Support for SiteMinder’s Basic Authentication**

  Arcot Adapter also supports the SiteMinder’s Basic (username/password) authentication. This authentication can be used in conjunction with the risk evaluation feature provided by RiskFort.

Chapter 2
Preparing for Installation

This chapter lists the software requirements for installing Arcot Adapter and discusses other prerequisites for a SiteMinder integration. The following sections are covered in this chapter:

- Software Requirements for Shim
- Software Requirements for FCC Pages
- Software Requirements for Token Server
- Software Requirements for Arcot Customization Engine
- Prerequisites for Integration

Software Requirements for Shim

Before proceeding with the Shim installation, ensure that a supported version of the software listed in the following table is installed and configured.

Table 2-1. Software Requirements for Shim

Software	Supported Version	Supported Operating System
CA SiteMinder Policy Server	6.00.05.00 and higher for Windows 32 (6.0 SP5)	Windows Server 2003

Software Requirements for FCC Pages

Before configuring the FCC pages, ensure that a supported version of the software listed in the following table is installed and configured.

Table 2-2. Software Requirements for FCC Pages

Software	Supported Version	Supported Operating System
CA SiteMinder Web Agent	6.0 and higher	For more information about the supported operating systems, see the SiteMinder Platform Support Matrix.

Software Requirements for Token Server

This section discusses the following topics:

- Software Requirements
- Configuring a Database Server
- Configuring the Application Server

Software Requirements

This sub-section provides the list of minimal software required to install Token Server.

Software Requirements

Before you deploy and configure Token Server, ensure that a supported version of the software listed in the following table is installed and configured.

📘	**Book:** For more information about the RiskFort installation requirements, see the *Arcot RiskFort 2.0 Installation and Deployment Guide*.

Table 2-3. Software Requirements for Token Server

Software	Supported Version	Supported Operating System
Arcot RiskFort	2.0	Windows 2003

Database Requirements

The following table lists the database requirements for Token Server.

Table 2-4. Minimum Database Version

Database Server
MS SQL Server 2005, Enterprise Edition (SP2)
Oracle Database 10*g*

JDK and Application Server Requirements

The following table lists the JDK and the application server requirements for Token Server.

Table 2-5. Minimum JDK and Application Server Version

JDK	Application Server
Sun JDK 5.0 Update 10	Apache Tomcat 5.5.23
Sun JDK 1.4.2 Update 13	Apache Tomcat 5.0.28
IBM 32-bit v5.0, SR2	IBM WebSphere 6.1
Compatible JDK version	BEA WebLogic Server 10.0 MP1

Configuring a Database Server

Before installing Arcot Adapter and integrating it with SiteMinder, you need to set up a database that is used by Token Server.

Use the following information when setting up the database server yourself, or provide this information to your database administrator (DBA) when you request a database account. This section includes the following topics:

- Configuring Microsoft SQL Server
- Configuring an Oracle Database

Configuring Microsoft SQL Server

Complete the following procedures to configure MS SQL Server for use with Token Server:

Book: For more information about completing the tasks in the following sections, see the MS SQL Server documentation.

1. Verifying the Authentication Mode
2. Creating a Database
3. Creating a Database User

Verifying the Authentication Mode

Verify that MS SQL Server is configured to use the "**SQL Server"** authentication method. If SQL Server is configured to "**Windows Only**" authentication, then Token Server cannot connect to the database.

Creating a Database

Use the following criteria to create a database

1. The recommended name is *arcottokenserver.*
2. Configure the database size to grow automatically.

Creating a Database User

To create a database user

Note: MS SQL server refers to user as a *Login.*

1. Go to the MS SQL management Graphic User Interface (GUI).
2. Enter the Login name. The recommended name is *tsadmin*.
3. Set the following parameters:
   a. Authentication to **"*SQL Server Authentication*"**.
   b. Default database to the database you created. For example, *arcottokenserver.*

c. Password for the login.

d. User Mapping for the default database you created. For example, *arcottokenserver.*

Configuring an Oracle Database

Complete the following procedures to configure an Oracle Database for use with Token Server:

> **Book:** For more information about completing the following tasks, see the Oracle documentation.

1. Creating a New Database
2. Creating a Database User

Creating a New Database

Create a new database that is used to store the Token Server information. The recommended name is *arcottokenserver.*

Creating a Database User

Create a user with the following criteria

1. Create a user with a schema in the new Token Server database. The recommended user name is *tsadmin*.
2. Grant the user `CONNECT` and `RESOURCE` privileges.
3. Grant the user the `CREATE TABLE` privilege.
4. Grant the user the `ALTER EXTENT PARAMETERS` privilege.
5. Grant the user privileges to modify the storage extents for the LOB columns.

Configuring the Application Server

Token Server is a web application that requires a servlet container for its deployment. Because Token Server uses JNDI to connect to the database, you must create a JNDI connection.

We recommend that Token Server communicate with other components using SSL mode. To configure Token Server for SSL, enable the application server on which Token Server is deployed for SSL communication.

Based on the application server you are using, refer to Chapter 4, “Deploying and Configuring Token Server” for details on these steps.

Software Requirements for Arcot Customization Engine

Before deploying and configuring the Customization Engine, ensure that a supported version of the software listed in the following table is installed and configured.

Book: For more information about installing Arcot WebFort, see the *Arcot WebFort 6.0 Installation and Deployment Guide*.

Table 2-6. Software Requirements for Arcot Customization Engine

Software	Supported Version	Supported Operating System
Arcot WebFort	6.0	Windows 2003

JDK and Application Server Requirements

The following table lists the JDK and the application server requirements for the Customization Engine.

Table 2-7. Minimum JDK and Application Server Version

JDK	Application Server
Sun JDK 5.0 Update 10	Apache Tomcat 5.5.23
Sun JDK 1.4.2 Update 13	Apache Tomcat 5.0.28
IBM 32-bit v5.0, SR2	IBM WebSphere 6.1
Compatible JDK version	BEA WebLogic Server 10.0 MP1

Prerequisites for Integration

The following requirements must be met before proceeding with the integration:

- At least two instances of application servers are running.
- Required number of database instances are ready with applicable schemas for storing the information required by Arcot Adapter.
- Arcot WebFort is installed on the required operating system.

> **Book:** See the *Arcot WebFort 6.0 Installation and Deployment Guide* for installation details.

- Arcot RiskFort is installed on the required operating system.

> **Book:** See the *Arcot RiskFort 2.0 Installation and Deployment Guide* for installation details.

- A SiteMinder Policy Server and a SiteMinder Web Agent are installed and configured.

> **Book:** See the appropriate SiteMinder documentation for installation details.

- Create at least one object of following types by using the SiteMinder Policy Server User Interface (r6.x) or Administrative User Interface (r12.x), as applicable. Refer to the appropriate SiteMinder documentation for more information on creating these objects:
  - Agents
  - Domains
  - Administrators
  - Realms
  - Users
  - User directories

  - Rules for the realms
- The redistributable package of Microsoft Visual C++ 2005 (x86), **`vcredist_x86.exe`**, is installed on the system(s) where the SiteMinder components are available.

  If not already installed, then you can install this redistributable package from the Arcot Adapter package, or from the following site:

  http://www.microsoft.com/downloads/details.aspx?FamilyID=9b2da534-3e03-4391-8a4d-074b9f2bc1bf&displaylang=en

Chapter 3
Installing Arcot Adapter

This chapter provides instructions for installing Arcot Adapter to provide the ArcotID authentication and risk evaluation services to the SiteMinder users. It discusses the following topics:

- Installing Arcot Adapter
- Installation Directory

	Note: Before you install Arcot Adapter, be sure that you have met the prerequisites detailed in Chapter 2, “Preparing for Installation”.

Installing Arcot Adapter

The Arcot Adapter installer supports the following installation types:

- Complete
- Custom

To install and configure Arcot Adapter on a single computer, use the *Complete* option. The *Complete* option lets you install all the components of Arcot Adapter on a single system. These components include the following:

- Customization Engine
- Token Server
- Shim
- FCC pages
- Scripts required for setting up the Database that you intend to use for Arcot Adapter

To install and configure Arcot Adapter in a distributed environment, use the *Custom* option. The Custom option lets you install selected components on multiple systems.

Note: Before you begin, be sure that all prerequisite software components are installed and the Token Server database is configured.

To install Arcot Adapter

1. Navigate to the directory where the `Arcot-Adapter-2.1.6-Windows-Installer.exe` file is located and double-click the file to run the installation wizard.

   The Welcome screen appears.

2. Click **Next** to continue.

   The License Agreement screen appears.

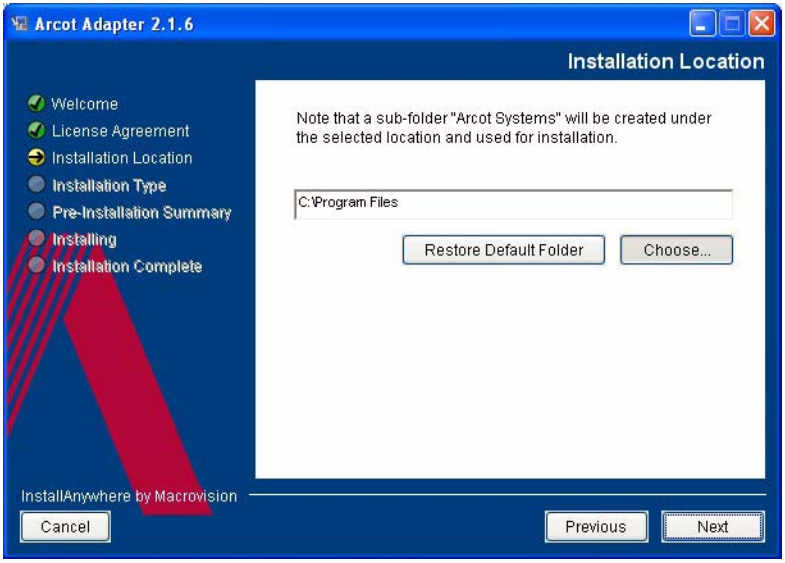
3. Read the license agreement carefully, select the **I accept the terms of the License Agreement** option, and click **Next.**

   The Installation Location screen appears.

4. The installer now verifies if any other Arcot product is installed on the computer.

   If it does not find an existing Arcot product installation, then you will be prompted for an installation directory. In this case, the Installation Location screen, shown in the following figure, appears.

Figure 3-1 The Installation Location Screen (No Arcot Product Found)

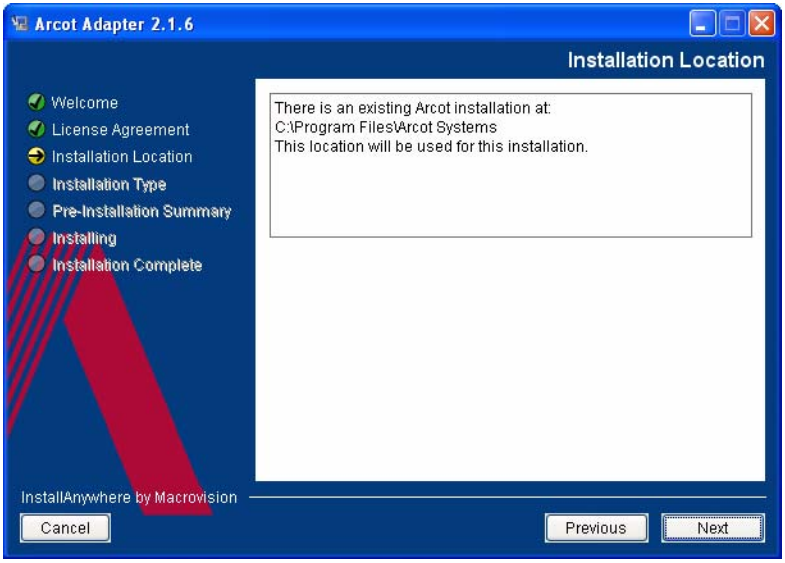


Note that a sub-folder "Arcot Systems" will be created under the selected location and used for installation.

C:\Program Files

If the installer detects an existing Arcot product installation, then you will not be prompted for an installation directory. The following screen appears when an existing `Arcot Systems` directory is found on the computer.

Figure 3-2 The Installation Location Screen (Arcot Product Found)



There is an existing Arcot installation at:
C:\Program Files\Arcot Systems
This location will be used for this installation.

5. You can accept the default directory specified by the installer to install Arcot Adapter. You can also click **Choose** to navigate and to specify a different directory.

   Click **Next** to install in the specified directory.

   The Type of Installation screen appears.

6. Select the type of installation:

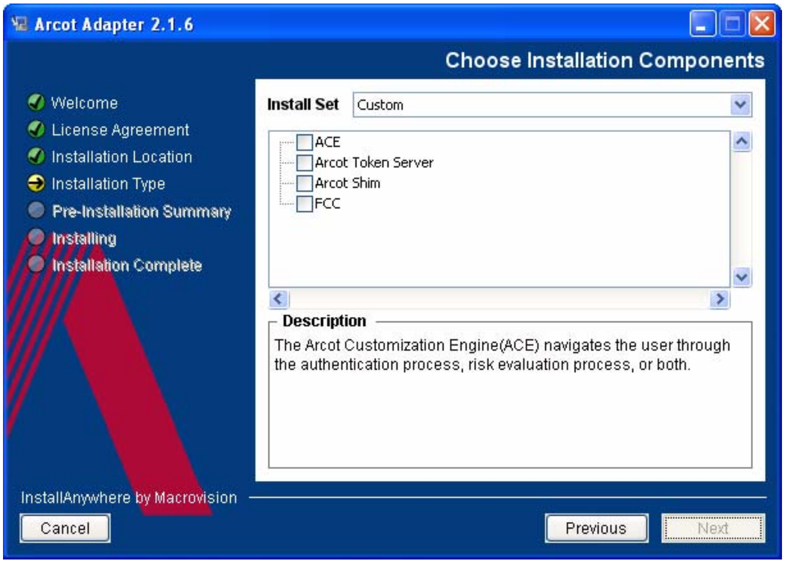
   - **Complete:** Select this option if you want to install *all* components of Arcot Adapter on the current system.
   - **Custom:** Select this option if you want to install only *selected* components of Arcot Adapter on the current system. In this case, you will need to install the remaining components on other system(s).

7. Click **Next** to continue.

   If you selected **Complete**, then proceed to Step 9.

   If you selected **Custom**, then the Choose Installation Components screen, as shown in the following figure, appears.

Figure 3-3 The Choose Installation Components Screen



Install Set: Custom

- □ ACE
- □ Arcot Token Server
- □ Arcot Shim
- □ FCC

Description

The Arcot Customization Engine(ACE) navigates the user through the authentication process, risk evaluation process, or both.

8. (**Custom Installation Only**) Select the components that you want to install on the current system and click **Next** to continue (Figure 3-3).

Table 3-1. Arcot Adapter Components

Components	Description
ACE	ACE navigates the user through the authentication process, risk evaluation process, or both.
Arcot Token Server	Arcot Token Server generates, maintains, and tracks the tokens that are used to associate the authentication and risk status of the user’s session across Arcot Adapter and SiteMinder components.
Arcot Shim	Arcot Shim is the core component that enables interaction between the Arcot components, SiteMinder, and other authentication schemes.
FCC	The FCC pages collect authentication input from the user and send it to the backing authentication scheme for authentication and risk evaluation.

9. Click **Next** to continue.

The Pre-Installation Summary screen appears.

Review the information on this screen, and if you need to change a previous selection, then click **Previous** to do so. After changing the required selection, click **Next** to go to the next screen.

10. Click **Install** to begin the installation process.

    The Installing Arcot SiteMinder Adapter screen appears. This might take several minutes.

    On successful installation, the Installation Complete screen appears.

11. Click **Done** to complete the installation.

Installation Logs

After installation, you can access the installation log file `Arcot_Adapter_2.1.6_InstallLog.log` from the following directory:

`<installation_dir>\logs\`

Note: `<installation_dir>` is the directory where Arcot Adapter is installed. By default, it is `C:\Program Files\Arcot Systems`.

Note: If for some reason, the installation failed, then an error log is available in the same location where you ran the Installer from.

Installation Directory

Arcot Adapter installs the files listed in the following table.

Table 3-2. Directory Structure

Component	Location	Files
Shim	`<installation_dir>\adapterSM\certs`	Contains the default root CA certificate, keystore, and truststore files the Shim requires.
	`<installation_dir>\adapterSM\lib`	Contains the following files: • **`ArcotLog2FileSC.dll`**: Log library file. • **`ArcotSiteMinderAdapter.dll`**: Shim library file. • **`vcredist_x86.exe`**: Microsoft re-distributable package.
	`<installation_dir>\conf`	Contains the `adaptershim.ini` file that specifies the Arcot Adapter configuration parameters.
FCC	`<installation_dir>\adapterSM\fcc`	Contains the FCC pages and **`js`** directory, which contains the JavaScript files. The **`fcc`** directory contains the following files: • **`shim2.fcc`** This page accepts the username, which is used for further processing. This FCC page is used in a Two-Page login scenario. In this scenario, the LDAP password is collected by the `shimfinal2.fcc` page. • **`shim.fcc`** This page accepts the username and the LDAP password as input for authenticating the user. This FCC page is used in a One-Page login scenario. • **`shimerror.fcc`** This page is displayed if an error occurs during authentication. • **`shimfinal2.fcc`** This page collects the LDAP password of the user. It is used in the Two-Page login scenario.

Table 3-2. Directory Structure

Component	Location	Files
		• `shimfinal.fcc` This page is used by the Customization Engine to redirect the user back to the Policy Server after risk evaluation. • `shimunknownuser.fcc` This page is displayed if you access the FCC pages directly and not as a result of redirection. • `shimerror.unauth.html` This page is displayed if the user enters incorrect credentials and exceeds the maximum number of login attempts SiteMinder allows.
Token Server	`<installation_dir>\adapterTS`	Contains a WAR file, and the JDBC drivers, properties files, and keystores that Token Server requires. Contains the following subdirectory: • `certs` Contains the keystore and truststore files that Token Server requires.
	`<installation_dir>\dbscripts`	Contains the SQL scripts required to create the Token Server schema in the supported database.
Customization Engine	`<installation_dir>\adapterACE`	Contains the WAR, keystore, and properties files the Customization Engine requires. Contains the following subdirectory: • `certs` Contains the keystore and truststore files the Customization Engine requires.
	`<installation_dir>\docs`	Contains the Java documents for the Customization Engine API.

Table 3-2. Directory Structure

Component	Location	Files
	`<`*`installation_dir`*`>\logs`	Contains the installation and uninstallation log files.
	`<`*`installation_dir`*`>\ext-license`	Contains the licenses of the third-party software that are used by Arcot Adapter.
	`<`*`installation_dir`*`>\Uninstall Arcot Adapter 2.1.6`	Contains the files required for uninstalling Arcot Adapter.

Chapter 4
Deploying and Configuring Token Server

This chapter provides the details that are required to deploy and configure Token Server successfully. It covers the following topics:

- Copying the JDBC Drivers
- Running Database Scripts
- Creating a JNDI Connection
- Deploying Token Server
- Enabling SSL
- Editing the Token Server Properties File
- Editing the Log Properties File
- Testing the Configuration

Copying the JDBC Drivers

Token Server connects to the database using a JDBC connection. The Arcot Adapter installation package provides the JDBC drivers that are used by Token Server. If you are using Apache Tomcat to deploy Token Server, then you can use these drivers. If you are using IBM WebSphere or Oracle WebLogic to deploy Token Server, then you can either use the default drivers that are shipped with the application server or the drivers that are provided with Arcot Adapter.

Copy the following driver:

For Oracle Database:

`ojdbc14.jar`

For MS SQL Database:

`sqljdbc.jar`

from

<*installation_dir*>\adapterTS\

to

<*TOMCAT_ROOT*>\common\lib\

Running Database Scripts

Note: Before you run the database scripts, be sure that you are logged in as the same database user you created when configuring the database server.

The Arcot Adapter installation includes the scripts that are required to create necessary tables in the database.

To create the required database tables

1. Navigate to the following location:

   For MS SQL:

   <*installation_dir*>\dbscripts\mssql\

   For Oracle:

   <*installation_dir*>\dbscripts\oracle\

2. Run the **arcot-db-config-for-adapter-tokenserver-2.1.sql** script to create the database tables.

   The arcot-db-config-for-adapter-tokenserver-2.1.sql creates the **ARTSTOKENS** table in the database. This table contains information about the token such as the token ID, time when the token was issued and last used, and the timestamp of communication with RiskFort.

Creating a JNDI Connection

This section describes the steps to create the JNDI connection on the following application servers that are supported by Token Server:

- Apache Tomcat
- IBM WebSphere
- Oracle WebLogic

Apache Tomcat

To create a JNDI connection in Apache Tomcat

1. Be sure that the Apache Tomcat application server is installed and functional.
2. Create a new file named **`arcottoksvr.xml`**.
3. Copy `arcottoksvr.xml` file to the following directory:

   <*TOMCAT_ROOT*>/conf/Catalina/localhost

   For Apache Tomcat 5.5.x and MS SQL database:

4. Open the `arcottoksvr.xml` file and add the following code:

```
<Context path="/arcottoksvr" docBase="arcottoksvr"
debug="5" reloadable="true" crossContext="true">
<Resource name="jdbc/ArcotTokenServerDataSource" auth="Container"
type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="30000"
driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
username="tsadmin" password="123456"
url="jdbc:sqlserver://<host>:<port>;databaseName=arcottokenserver"/>
</Context>
```

For Apache Tomcat 5.5.x and Oracle Database:

5. Open the `arcottoksvr.xml` file and add the following code:

```
<Context path="/arcottoksvr" docBase="arcottoksvr"
debug="5" reloadable="true" crossContext="true">
<Resource name="jdbc/ArcotTokenServerDataSource" auth="Container"
type="javax.sql.DataSource"
maxActive="100" maxIdle="30" maxWait="30000"
driverClassName="oracle.jdbc.driver.OracleDriver"
username="tsadmin" password="123456"
url="jdbc:oracle:thin:@<host>:<port>:<sid>"/>
</Context>
```

6. Replace the following parameters in the **Context**, **Resource** section.

Table 4-1. Configuration Parameters

Parameter	Replace With
`username`	Logon identifier of the database user.
`password`	Logon password of the database user.
`url`	Change the `host` and `post` information in the `url` parameter to that of the database server.

7. Save and close the `arcottoksvr.xml` file.

IBM WebSphere

In the IBM WebSphere Management Console, use the **Resources**, **JDBC Node** option to create a data source with the following JNDI name:

`jdbc/ArcotTokenServerDataSource`.

Oracle WebLogic

To create a JNDI connection in Oracle WebLogic

1. In the Oracle WebLogic Administration Console, use the **Services**, **JDBC**, **Data Sources** option to create a data source with the following JNDI name:

   `ArcotTokenServerDataSource`

2. Expand the `arcottoksvr.war` file by using the following command:

   ```
   jar -xvf arcottoksvr.war
   ```

3. Edit the following section in the **`web.xml`** file available in the expanded directory:

   ```
   <resource-ref>
       <description>Your Description</description>
       <res-ref-name>ArcotTokenServerDataSource</res-ref-name>
       <res-type>javax.sql.DataSource</res-type>
       <res-auth>Container</res-auth>
     </resource-ref>
   ```

4. Delete the `jdbc/` prefix from the data source name in the `TsSqlMapConfig.xml` file that is available in the following directory:

   ```
   arcottoksvr\WEBINF\classes\com\arcot\integrations\toksvr\server\
   tsimpl\
   ```

5. Re-war the `arcottoksvr.war` file and deploy it in the application server.

Deploying Token Server

You use the **`arcottoksvr.war`** file to deploy Token Server. This file is available at the following location:

`<`*`installation_dir`*`>\adapterTS\`

To deploy Token Server

1. Install `arcottoksvr.war` on the application server.

For example, on Apache Tomcat the location to install the WAR file is as follows:

```
<APP_SERVER_HOME>\webapps
```

Note: The deployment procedure depends on the application server that you are using. See your application server vendor documentation for detailed instructions.

2. Restart the application server.

   The application server should now contain a directory named **arcottoksvr**.

Enabling SSL

We recommend that you enable Token Server to communicate with the other components over SSL. Enable the application server on which Token Server is deployed for SSL communication. For more information, see your vendor-specific documentation.

Editing the Token Server Properties File

You can choose one of the following methods to edit the Token Server properties file:

- Method 1: Use this method if you have *already deployed* Token Server.
- Method 2: Use this method if you have *not deployed* Token Server.

Method 1

To edit the Token Server properties file on the application server

1. Navigate to the following directory:

Note: The location mentioned here is specific to Apache Tomcat. If you are using any other application server, refer to the application server vendor documentation for the corresponding path.

```
<APP-SERVER-HOME>\webapps\arcottoksvr\WEB-INF\classes\
```

2. Make a copy of the **`arcottokenserver.properties.src`** file and rename it to the following:

   `arcottokenserver.properties`

3. Edit the `arcottokenserver.properties` file to set the following parameters:

Table 4-2. RiskFort Parameters

Parameter	Description
`RiskFortHOST.1`	The IP address or the Fully Qualified Distinguished Name (FQDN) of the RiskFort host system.
`RiskFortPORT.1`	The port where RiskFort is listening to incoming requests. Default: **`7680`**
`RiskFortTRANSPORT_TYPE`	The default protocol for RiskFort to start up. Default: **`TCP`** **Book:** We recommend that the Token Server communicate with RiskFort over SSL. For more information about configuring RiskFort to communicate over SSL, see the *Arcot RiskFort 2.0 Installation and Deployment Guide*.
`RiskFortCA_CERT_FILE`	The path for the CA certificate file of the server. The file *must* be in `.PEM` format. Provide the *complete path* for the file.
`RiskFortCONNECTION_TIMEOUT`	The time (in milliseconds) before RiskFort is considered unreachable.
`RiskFortREAD_TIMEOUT`	The maximum time allowed for a response from RiskFort.
`RiskFortCONNECTION_RETRIES`	The maximum number of retries allowed to connect to RiskFort.
`RiskFortUSE_CONNECTION_POOLING`	Specifies If a connection pooling with the RiskFort server is enabled or disabled. Default: **`1`** *(enabled)*
`RiskFortMAX_ACTIVE`	The maximum number of connections that can exist between Token Server and RiskFort. Default: **`32`**
`RiskFortTIME_BETWEEN_CONNECTION_EVICTION`	The time (in milliseconds) after which the connection eviction thread is executed to check and delete any idle RiskFort server connection.

Table 4-2. RiskFort Parameters (Continued)

Parameter	Description
`RiskFortIDLE_TIME_OF_CONNECTION`	The interval (in milliseconds) after which an idle RiskFort Server connection is dropped. Default: **`1800000`**
`RiskFortWHEN_EXHAUSTED_ACTION`	The behavior when the maximum number of supported connections are exhausted. Default: **`BLOCK`**

4. Configure the token-related parameters, which are described in the following table.

Table 4-3. Token Parameters

Parameter	Description
`TokenMaxInactivitySeconds`	The time for which the token can be idle after an operation is performed on it. If there is no action on the token within this period, then the token becomes unusable. Default: **`180`**
`TokenMaxLifetimeSeconds`	The maximum amount of time the token is accessible after it is generated. Default: **`900`**
`TokenCleanupIntervalSeconds`	The frequency at which the expired tokens are checked and deleted from the database. Default: **`30`**
`TSMClass`	The type of storage mechanism to be used for Token Server, which is a JDBC database. Default: **`com.arcot.integrations.toksvr.server.tsmimpl.iBatisTSMImpl`**

5. To enforce a secure communication between Token Server and other components, ensure that the parameter `RequireSecureConnection` is set to `true`, which is the also the default value.

6. Proceed with log properties file configuration using the "Method 1" as described in the section, "Editing the Log Properties File".

Method 2

Arcot Adapter installs the `arcottokenserver.properties` file on the file system. Edit this file as follows:

1. Open the `arcottokenserver.properties` file from the following directory:

   `<installation_dir>\adapterTS\`

2. Edit the parameters as described in Step 3 to Step 5 of Method 1.
3. Proceed with log properties file configuration using the "Method 2" as described in the section, "Editing the Log Properties File".

Editing the Log Properties File

You can choose one of the following methods to edit the Token Server log properties file:

- Method 1: Use this method if you have *already deployed* Token Server.
- Method 2: Use this method if you have *not deployed* Token Server.

Method 1

To edit the Token Server log properties file on the application server

1. Navigate to the following directory:

> **Note:** The location mentioned here is specific to Apache Tomcat. If you are using any other application server, then refer to the application server vendor documentation for the corresponding path.

   `<APP-SERVER-HOME>\webapps\arcottoksvr\WEB-INF\classes\`

2. Make a copy of the **`log4j.properties.src`** file and rename it to the following:

   `log4j.properties`

3. Edit the `log4j.properties` file to set the following log information:

   Table 4-4. Log Parameters

Parameter	Description
`log4j.appender.toksvrlog.File`	The log file name and the location where Token Server logs must be written to. By default, the Token Server log file name is `arcottokenserver.log` and is created in the `<APP_SERVER_HOME>\logs` directory.

Method 2

Arcot Adapter installs the `log4j.properties` file on the file system. Edit this file as follows:

1. Open the `log4j.properties` file from the following directory:

   ```
   <installation_dir>\adapterTS\
   ```

2. Edit the parameters as described in Table 4-4.
3. Create the `arcottoksvr.war` file with the edited `arcottokenserver.properties` and `log4j.properties` files.
4. Deploy the `arcottoksvr.war` file in the application server.

Testing the Configuration

To test Token Server configuration

1. Restart the application server.
2. Access Token Server using the following URL:

   https://<Host>:<Port>/arcottoksvr/index.jsp

3. Open the Token Server log file from the location you have configured it in the `log4j.properties` file. By default, the log file is available in the following directory:

   For Apache Tomcat 5.5

   ```
   <APP_SERVER_HOME>\logs
   ```

4. Check the following entry in the log file, which indicates that Token Server is configured successfully.

```
Servlet com.arcot.integrations.toksvr.server.TokenCreator starting
up

Servlet com.arcot.integrations.toksvr.server.TokenRemover starting
up

Servlet com.arcot.integrations.toksvr.server.TokenReader starting up
```

Chapter 5
Deploying and Configuring Arcot Customization Engine

This chapter lists the tasks that you must perform to deploy and configure the *Arcot Customization Engine* (ACE) successfully. It covers the following topics:

- Deploying ACE
- Enabling SSL
- Editing the ACE Properties File
- Editing the Log Properties File
- Testing the Configuration

Note: Before deploying and configuring the ACE application, be sure that the WebFort Server is started and running.

Deploying ACE

You use the **arcotauthui.war** file to deploy ACE. This file is available at the following location:

`<installation_dir>\adapterACE\`

To deploy the ACE application

1. Install `arcotauthui.war` on the application server.

   For example, on Apache Tomcat the location to install the `WAR` file is as follows:

   `<APP_SERVER_HOME>\webapps`

   Note: The deployment procedure depends on the application server that you are using. See your application server vendor documentation for detailed instructions.

2. Restart the application server.

   The application server should now contain a directory named **`arcotauthui`**.

Enabling SSL

We recommend that you enable ACE for SSL communication. To enable ACE to communicate with Token Server over SSL, set the following configuration parameters in the `arcotauthui.properties` file.

- TokenServerTrustStore
- TokenServerTSPassword
- TokenServerKeyStore
- TokenServerKSPassword

Editing the ACE Properties File

You can choose one of the following methods to edit the ACE properties file:

- Method 1: Use this method if you have *already deployed* ACE.
- Method 2: Use this method if you have *not deployed* ACE.

Method 1

To edit the ACE properties file on the application server

1. Navigate to the following directory:

> **Note:** The location mentioned here is specific to Apache Tomcat. If you are using any other application server, refer to the application server vendor documentation for the corresponding path.

```
<APP-SERVER-HOME>\webapps\arcotauthui\WEB-INF\classes\
```

2. Make a copy of the **`arcotauthui.properties.src`** file and rename it to the following:

   `arcotauthui.properties`

3. Edit the `arcotauthui.properties` file to set the following parameters:

Table 5-1. The Customization Engine Configuration Parameters

Parameter	Description
`ArcotAuthUILandingURL`	The URL of the Customization Engine's controller JSP. The controller JSP depends on the authentication workflow. This is an optional overriding parameter for the Landing URL.
`TokenServerHostname`	The IP address of Token Server host system.
`TokenServerPort`	The port of the application server on which Token Server is deployed.
`TokenServerURLBase`	The path where Token Server is available. Default: **`arcottoksvr/servlet`**
`TokenServerSecureConnection`	Specifies whether ACE is configured to communicate with Token Server over SSL. Limits: • `true` • `false` Default: **`true`**
`TokenServerTrustStore`	The path where the root SSL certificate of the server is present. This parameter is valid if `TokenServerSecureConnection` is set to `true.`
`TokenServerTSPassword`	The password of the truststore. This parameter is valid if the `TokenServerTrustStore` path is provided.
`TokenServerKeyStore`	The path of the client SSL certificate.
`TokenServerKSPassword`	The password of the keystore.
`TokenServerConnTimeoutMS`	The interval (in milliseconds) after which an idle Token Server connection is dropped. Default: **`15000`** (15 seconds)
`TokenServerReadTimeoutMS`	The period (in milliseconds) for which the ACE must wait for a response from Token Server. Default: **`30000`** (30 seconds)

Table 5-1. The Customization Engine Configuration Parameters (Continued)

Parameter	Description
TokenServerTestConnAtStartp	Whether a test token must be created when the web application is started. **Note:** If you are using JRE 1.4.2.x and the ACE starts before Token Server, then ACE cannot time-out the connection, and it does not start up. Limits: • true • false
WebFortHostName	The IP address or the Fully Qualified Distinguished Name (FQDN) of the WebFort host system.
WebFortPort	The port where WebFort is listening to incoming requests. Default: **9742**
WebFortTransport	The default protocol for WebFort to start up. Default: **TCP** **Note:** We recommend that the ACE communicate with WebFort over SSL. For more information about configuring WebFort to communicate over SSL, see the *Arcot WebFort 6.0 Installation and Deployment Guide*.
WebFortCA_CERT_FILE	The path for the CA certificate file of the server. The file *must* be in .PEM format. Provide the *complete path* for the file.
WebFortMaxConnPoolSize	The maximum number of connections that can exist between the Customization Engine and WebFort. Default: **32**
DeviceIDType	The type of cookie that must be stored on the end-user’s system. RiskFort uses Device ID to register and identify the device that is used by a user during transactions. The Device ID needs to be set as a cookie on the user’s computer. This cookie can either be a HTTP cookie or a Flash cookie. Limits: • httpcookie • flashcookie

Table 5-1. The Customization Engine Configuration Parameters (Continued)

Parameter	Description
`ClientType1`	The ArcotID Client type that must be used when an ArcotID authentication is performed by WebFort. Limits: • `Flash` • `ActiveX` • `Applet` • `UnsignedApplet`
`ClientType2`	The client type to be used for authentication. If the ArcotID Client type specified in the `ClientType1` parameter is not available, then the Arcot Adapter checks this parameter for which client to use. Limits: • `Flash` • `ActiveX` • `Applet` • `UnsignedApplet`
`ClientType3`	The client type to be used for authentication. If the ArcotID Client type specified in the `ClientType1` and `ClientType2` parameters are not available, then the Arcot Adapter checks this parameter for which client to use. Limits: • `Flash` • `ActiveX` • `Applet` • `UnsignedApplet`
`StopActionMode`	This option lets you stop the automatic posting or redirecting of the ACE pages. The pages include a button which you must click to continue to the next pages. Limits: • `true` • `false` Default: **`false`**

4. Proceed with log properties file configuration using the “Method 1” as described in the section, “Editing the Log Properties File”.

Method 2

Arcot Adapter installs the `arcotauthui.properties` file on the file system. Edit this file as follows:

1. Open the `arcotauthui.properties` file from the following directory:

   `<installation_location>\adapterACE\`

2. Edit the parameters as described in Table 5-1.

3. Proceed with log properties file configuration using the "Method 2" as described in the section, "Editing the Log Properties File".

Editing the Log Properties File

You can choose one of the following methods to edit the ACE log properties file:

- Method 1: Use this method if you have *already deployed* ACE.
- Method 2: Use this method if you have *not deployed* ACE.

Method 1

To edit the ACE log properties file on the application server

1. Navigate to the following directory:

> **Note:** The location mentioned here is specific to Apache Tomcat. If you are using any other application server, then refer to the application server vendor documentation for the corresponding path.

   `<`*`APP-SERVER-HOME`*`>\webapps\arcotauthui\WEB-INF\classes\`

2. Make a copy of the **`log4j.properties.src`** file and rename it to the following:

   `log4j.properties`

3. Edit the log4j.properties file to set the following log information:

Table 5-2. Log Parameters

Parameter	Description
log4j.appender.authuiout.File	The log file name and the location where the Customization Engine logs must be written to. Default: the Customization Engine log file name is arcotauthui.log and is created in <APP-SERVER-HOME>\logs directory.

Method 2

Arcot Adapter installs the log4j.properties file on the file system. Edit this file as follows:

1. Open the log4j.properties file from the following directory:
   <installation_location>\adapterACE\
2. Edit the parameters as described in Table 5-2.
3. Create arcotauthui.war with the edited arcotauthui.properties and log4j.properties files.
4. Deploy the arcotauthui.war file in the application server.

Testing the Configuration

To test the Customization Engine configuration

1. Restart the application server.
2. Open the Customization Engine log file from the location you have configured it in the log4j.properties file. By default, the log file is available in the following directory:

   For Apache Tomcat 5.5

   <*APP_SERVER_HOME*>\logs
3. Check the following entry in the log file, which indicates the Customization Engine is configured successfully:

```
InitializeTokenSvrClientServlet for Adapter ACE Frontend  version
2.1
WebFort connection test successful
```

Chapter 6
Configuring Arcot Shim and FCC Pages

This chapter provides the details that are required to configure Arcot Shim and Form Credential Collector (FCC) pages successfully. It covers the following topics:

- Deploying the FCC Pages
- Deploying the Shim
- Enabling SSL
- Configuring the Shim
- Testing the Configuration

Deploying the FCC Pages

To deploy the FCC pages, copy the FCC pages and the `js` directory from the following location:

`<installation_dir>\adapterSM\fcc\`

Copy these files to the root directory of your web server where the SiteMinder Web Agent is installed.

In addition to copying the files, you must also create a virtual directory in the web server, and name this directory as `arcotlogin`. By default, Arcot looks for a virtual directory named `arcotlogin`. If you use a different name, then you must edit the `<path>` of the following parameters found in the **`adaptershim.ini`** file:

- ErrorPageURL
- InitialFCCURL
- FinalFCCURL

The virtual directory must point to the directory where the `js` directory and FCC pages are copied.

Deploying the Shim

The files required to deploy Arcot Shim are available at the following location:

`<installation_dir>\adapterSM\lib\`

To deploy the Shim

1. The SiteMinder Policy Server requires the Shim library and the log library files to be present in the System Path variable. You can do this by performing one of the following operations:

   Note: If you deploy the Shim while the SiteMinder Policy Server is running, restart the SiteMinder Policy Server.

   - Copying the `ArcotSiteMinderAdapter.dll` and `ArcotLog2FileSC.dll` files, available at:

     `<installation_dir>\adapterSM\lib`

     to the `<policy_server_home>\bin` directory of SiteMinder Policy Server.
   - Including the `<installation_dir>\adapterSM\lib` directory in the **`Path`** variable.

2. Be sure that the Microsoft VC++ 2005 SP1 Redistributable package (`vcredist_x86.exe`) is installed on the system hosting the Policy Server. If not, install it is from the following location:

   `<installation_dir>\adapterSM\lib\`

Enabling SSL

We recommend that you enable the Shim for SSL communication. To enable the Shim to communicate with Token Server over SSL, set the following configuration parameters in the `adaptershim.ini` file:

- TokenServerTrustedRootPEM
- TokenServerClientSSLCert
- TokenServerClientPrivateKey

Configuring the Shim

The Shim configuration is defined in the `adaptershim.ini` file, which specifies the configuration parameters that are required to let Arcot Adapter and SiteMinder to communicate with each other. The `adaptershim.ini` file is installed at the following location:

`<installation_dir>\conf`

The section `[arcot/integrations/smadapter/Default]` contains the parameters that you need to set according to the authentication flow that you want to use. Table 6-1 explains the parameters of this section.

Table 6-1. Configuration Parameters

Parameter	Mandatory	Description
`DisambigSchemeLib`	No	The DLL library name of an authentication scheme to use for user disambiguation. **Note:** This parameter does not support the refresh option. If you use the Arcot Adapter authentication, then you must restart the SiteMinder Policy Server.
`DisambigSchemeParam`	No	The parameter string to pass to the authentication scheme performing the user disambiguation. The string must be structured the same way as the SiteMinder Policy Server would build the string from the configuration parameters for the scheme.
`AuthSchemeLib`	No	The library name of an authentication scheme to use as a backing scheme for primary authentication. **Note:** This parameter does not support the refresh option. If you use the Arcot Adapter authentication, then you must restart the SiteMinder Policy Server. **Note:** This parameter is not used in a delegated authentication scenario.

Table 6-1. Configuration Parameters (Continued)

Parameter	Mandatory	Description
`AuthSchemeParam`	No	If the backing authentication scheme is configured, then this parameter is passed to it as its configuration string. It must be set to have the same content as the SiteMinder Policy Server would set from the scheme configuration dialog. You can determine this by examining the scheme setup dialogs in the SiteMinder Policy Server administration interface. As you change parameters, the dialog shows the **Parameter** that the SiteMinder Policy Server would send. **Note:** This parameter is not used in a delegated authentication scenario.
`TokenServerBaseURL`	Yes	The URL where Token Server is available. The syntax to specify the Token Server URL is: `https://<Host>:<Port>/arcottoksvr/servlet/`
`TokenServerRetries`	Yes	The maximum number of retries allowed to connect to Token Server. If this value is **`0`**, it signifies only one connection attempt is allowed.
`TokenServerResponseWait`	Yes	The time period (in seconds) for which the Shim will wait for Token Server to respond before logging an error.
`TokenServerTrustedRootPEM`	If HTTPS is enabled	The location of the certificate of the trusted root certificate authority, if Token Server is enabled for HTTPS. The file *must* be in `.PEM` format.
`TokenServerClientSSLCert`	If HTTPS is enabled	The location of the client-side SSL certificate, if Token Server is enabled for HTTPS. The file *must* be in `.PEM` format.
`TokenServerClientPrivateKey`	If HTTPS is enabled	The private key of the client in `.PEM` format, if Token Server is enabled for HTTPS. The file *must* be in `.PEM` format.
`ArcotAuthUILandingURL`	Yes	The controller JSP URL of the Customization Engine. **Note:** Although you can use multiple sample workflows, only one `ArcotAuthUILandingURL` can be configured.
`UseCustomizationEngineAuth`	No	Specifies whether ACE should perform authentication. • **OnePage** - `false` • **TwoPage** - `false` • **DelegatedAuth** - `true` • **UseHTMLAuth** - `false`

Table 6-1. Configuration Parameters (Continued)

Parameter	Mandatory	Description
`InitialPhasePrimaryAuth`	No	Specifies whether primary authentication should be performed before risk evaluation. This parameter is applicable if `UseCustomizationEngineAuth` is set to `false`.
`ErrorPageURL`	Yes	The URL of the error FCC page. This page is displayed to the user if an error occurs.
`InitialFCCURL`	Yes	The URL of the initial FCC page served to the user. The Shim reports this URL to SiteMinder during initialization. When the user attempts to access a protected resource and authentication is required, SiteMinder directs the user to this page. Depending on the authentication workflow, the page can collect information such as the username or username and password.
`FinalFCCURL`	Yes	The URL that ACE uses to forward control back to the Shim. ACE retrieves this URL from the workflow state record.

Configuring Global Information

The global Shim configuration parameters are available in the `GLOBAL SETUP` section of the `adaptershim.ini` file. The following table describes the parameters of the `[arcot/integrations/smadapter]` section.

Table 6-2. Global Configuration Parameters

Parameter	Mandatory	Description
`WatchInterval`	Yes	The polling interval, in seconds, that the Shim uses to monitor the `adaptershim.ini` file. The Shim monitors the `adaptershim.ini` file and allows configuration changes without restarting the SiteMinder Policy Server. It monitors the configuration file at this interval and, if the file has changed, then it reloads the file. Default: **`300`** (seconds)
`ShimIdentifierString`	No	A unique identifier of the Shim instance. The value that you specify is appended with the section name to create identifier.
`LogSupported`	Yes	The number of log files to send log messages to. If you set this value to **`0`**, then no logging is performed.

Configuring the Log Information

The Shim generates log messages as part of its operation to support error reporting, auditing, and debugging. The level of details logged by the Shim can be configured.

All Shim log messages, except trace messages, are written to the SiteMinder Policy Server log file (`smps.log`). All trace messages are logged in the files that are configured in the SiteMinder Policy Server.

All entries that are logged in the `smps.log` file are also logged in the Arcot Adapter log file (`arcotadaptershim.log`). However, the level of message details in the Arcot Adapter log file is determined by the `HandleLevel` parameter.

The log-related parameters are defined in the `LOGGING SETUP` section of the `adaptershim.ini` file.

Setting up the Log Parameters

The following table describes the log parameters defined in the `[arcot/integrations/smadapter/LogLibrary<n>]` section.

Table 6-3. Log Configuration Parameters

Parameter	Description
`DLLName`	The name of the library file that performs logging. **Note:** Do *not* specify the suffix of the file name; it is automatically added at run-time.
`HandleLevel`	The level of detail that must be included in the log messages. Messages with a value equal to or greater than the specified value are included in the log message. For example, if the value is set to 2, then messages of severity level 2 to 7 are logged. Limits: • `1=low` • `2=info` • `3=notice` • `4=warning` • `5=error` • `6=alert` • `7=fatal` **Note:** We recommend that you set this value to `3` or lower.

Table 6-3. Log Configuration Parameters (Continued)

Parameter	Description
`EntryPoint`	The function within the library that must be called to get a handle to the logging object. **Note:** This is fixed for a given log handler DLL.
`ParamSupported`	The count of parameters to pass to the logging object.
`Param1=LOG_FILE_NAME`	The name and location of the log file. Default: **`%ARCOT_HOME%\logs\arcotadaptershim.log`**
`Param2=LOG_FILE_ROLLOVER_INTERVAL`	Specifies how often to rollover the log file to a backup file. Limits: • `HOURLY` • `DAILY` • `WEEKLY` • `MONTHLY`
`Param3=MAX_LOG_FILE_SIZE`	The maximum size of the log file. This is an alternative way to indicate rollover, if the rollover interval is not set. The file size is expressed in bytes. For example: `Param3=MAX_LOG_FILE_SIZE=10000000` The above value indicates the size of the log file is approximately **`10`** MB. **Note:** If this parameter is set to **`0`**, then the log file will continue to grow indefinitely.
`Param4=BACKUP_LOG_FILE_LOCATION`	The complete path where the backup log file is stored. The path provided must be valid. Default: **`%ARCOT_HOME%\logs\backup`**
`Param5=LOG_LINE_FORMAT`	The format of the logging string. This indicates the attributes that are logged on each line of the file. **Note:** If this parameter is not set, then the legacy format is used. Limits: • `LTZ=System Timezone, Date, and Time` • `SEV=Severity` • `PID=ProcessID` • `TID=ThreadID` • `MID=MessageIDNumber` • `MSG=Log Message Text` • `LID=LoggingID`

Testing the Configuration

To test the Shim configuration

1. Open the `arcotadaptershim.log` log file available in the following directory:

> **Note:** By default, the installer does not create this file. It is generated when the Shim receives the first authentication request.

```
<installation_dir>\logs
```

2. Check the following entry in the log file, which indicates that the Shim is configured successfully.

```
Logger initialized
STARTING [Arcot Adapter 2.1]
Starting watchdog thread...
```

Chapter 7
Configuring the CA SiteMinder Policy Server

This chapter describes the SiteMinder Policy Server configuration steps that you must perform after installing Arcot Adapter.

To configure the SiteMinder Policy Server to use Arcot Adapter

Book: Refer to the appropriate SiteMinder documentation for more information on the tasks listed in this procedure.

1. Create a custom authentication scheme in SiteMinder. In the SiteMinder Policy Server User Interface or Administrative UI (as appropriate for your version of SiteMinder), create a new Authentication Scheme by configuring the following properties:
   - **Name**: A name for the authentication scheme.
   - *(Optional)* **Description**: A description for the authentication scheme.
   - **Authentication Scheme Type**: Set this to **Custom Template**.
   - **Library**: The Arcot Adapter library file name (`ArcotSiteMinderAdapter`)
   - **Parameter**: The name of the configured workflow. This value must correspond to section name in the `adaptershim.ini` file. The following are the default values that are provided in the `adaptershim.ini` file:
     - `OnePage`
     - `TwoPage`
     - `DelegatedAuth`
     - `UseHTMLAuth`
2. Assign the new authentication scheme to each realm that contains resources that should be protected by Arcot Adapter.
3. For the SiteMinder Policy Server to work with Arcot Adapter, set the following Agent Configuration Object (ACO) parameters:
   - **CssChecking -** Set this to YES.

- **FCCCompatMode -** Set this to YES.
- **AgentName -** Set this to the name of the agent.
- **LogFileName -** Enter the log file name of the Web Agent. (This is not a mandatory setting, but can be used for debugging.)
- **DefaultAgentName -** Enter the name of the default Web Agent.
- **DefaultPassword -** Enter the Web Agent password.
- **LogFileSize -** Enter the size of the Web Agent log file.
- **LogFile -** Set this to YES.
- **RequireCookies -** Set this to YES.
- **TraceConfigFile -** Enter the name of the trace configuration file. (This is not a mandatory setting, but can be used for debugging.)
- **TraceFile -** Set this to YES.
- **TraceFileName -** Enter the name of the trace file.
- **TraceFileSize -** Enter the size of the trace file.

Chapter 8
Uninstalling Arcot Adapter

The steps to uninstall Arcot Adapter include:

- Dropping Arcot Adapter Schema
- Uninstalling Arcot Adapter
- Post-Uninstallation Steps

Dropping Arcot Adapter Schema

Note: If for some reason, you need to retain the database, then *do not* proceed with the instructions in this section. Refer to section "Uninstalling Arcot Adapter" to proceed with the uninstallation.

To drop the Arcot Adapter database schema

1. Based on the database you are using, navigate to one of the following subdirectories:

   For MSSQL:

   `<installation_dir>\dbscripts\mssql\`

   For Oracle:

   `<installation_dir>\dbscripts\oracle\`

2. Run the following script to delete the database tables:

   - `drop-adapter-tokenserver-2.1.sql`

   This drops all database tables created by Arcot Adapter.

Uninstalling Arcot Adapter

To uninstall Arcot Adapter

1. Navigate to the following directory:

   ```
   <installation_dir>\Uninstall Arcot Adapter 2.1.6
   ```

2. Double-click `Uninstall Arcot Adapter 2.1.6.exe`.

   The Uninstall Options screen appears.

3. Select **Complete Uninstall** to uninstall all components of Arcot Adapter and go to Step 5.

   To uninstall the selected components, select **Uninstall Specific Features** and click the **Next** button.

   The Choose Product Features screen appears.

4. (**For Uninstalling Specific Components Only**) This screen displays the Arcot Adapter components that are installed on the current system. Deselect the components you wish to uninstall and click the **Next** button.

   The Backup Location screen appears.

5. If you want to take a backup of important files, such as configuration or log files, then select a location where you want to store these files and click **Uninstall** to uninstall Arcot Adapter.

   The Uninstallation *Complete* screen appears at the end of successful uninstallation.

6. Click **Done** to exit the wizard.

Post-Uninstallation Steps

Perform the following post-uninstallation steps:

1. Delete the installation directory (*`<installation_dir>`*).

   > **Note:** If multiple Arcot products are installed on this system, then delete this directory only if Arcot Adapter is the last product to be uninstalled.

2. If you have deployed the Token Server and Customization Engine, then undeploy the following `WAR` files from the appropriate location on your application server.

   - `arcotauthui.war -` Customization Engine
   - `arcottoksvr.war -` Token Server

   For example, on Apache Tomcat the location is `<APP-SERVER-HOME>\webapps`. Here, *`APP-SERVER-HOME`* represents the directory path where Apache Tomcat is installed.

> **Note:** If you have a distributed-system deployment, then locate these `WAR` files on the system where you have deployed the particular component.

Appendix A
Configuring Backing Authentication Scheme

Arcot Adapter supports external or third-party authentication schemes or mechanisms. These mechanisms are referred to as *backing authentication* in the Arcot Adapter terminology.

If a backing authentication scheme is configured, the Arcot Shim acts as an interface between SiteMinder and the backing authentication mechanism. It forwards the authentication requests to the backing method, and when it receives the authentication result back from the backing authentication method, it posts the same to the SiteMinder Policy Server. In this case, Arcot Adapter is used for risk evaluation.

When a backing authentication scheme is configured for the Arcot Shim, it dynamically loads the external authentication scheme. The Shim can also delegate the SiteMinder authentication calls to the backing authentication scheme.

Typically, the Shim is transparent to the backing authentication scheme. However, if the Customization Engine directs that the transaction should be terminated immediately (for example, risk evaluation indicates `DENY`), then the Shim can override successful authentication result from the backing authentication scheme.

This appendix details the process of configuring a backing authentication scheme with Arcot Adapter:

1. Configuring the Shim for a Backing Authentication Scheme
2. Configuring the Policy Server for the Backing Authentication Scheme
3. Configuring FCC Pages

Configuring the Shim for a Backing Authentication Scheme

You configure the backing authentication scheme in the SiteMinder Policy Server User Interface or Administrative UI (as appropriate for your version of SiteMinder). However, the Arcot Shim loads the backing authentication schemes. As a result you configure a majority of the authentication scheme in the Shim configuration file.

To configure the Shim to use a backing authentication scheme, the following three parameters must be configured in `adaptershim.ini` file:

1. Scheme DLL Name (`AuthSchemeLib`)

   The shared library name for the backing authentication scheme is configured in the Shim configuration file, as `AuthSchemeLib` parameter for authentication.

2. Scheme Parameter String (`AuthSchemeParam`)

   Most scheme configuration data is stored in the parameter string. This string is configured by using the `AuthSchemeParam` in the Shim configuration file. The content of this string is specific to the backing authentication scheme you are using.

3. FCC URLs (`ErrorPageURL` and `FinalFCCURL`)

   Arcot Adapter serves these FCC pages to the user for handling user interactions and for handling errors. Be sure that these are configured to point to the following:

   - `ErrorPageURL: shimerror.fcc`
   - `FinalFCCURL: shimfinal.fcc`

Configuring the Policy Server for the Backing Authentication Scheme

If the authentication scheme requires a shared secret, then it must be configured in the Policy Server User Interface or Administrative UI (as appropriate for your version of SiteMinder) as the **Shared Secret** for the Arcot Shim scheme. The Shim, which itself does not use a shared secret, passes this parameter to the backing authentication scheme.

Note: If the backing authentication schemes are used for disambiguation and authentication, then they must use the same shared secret, because *only one* shared secret can be configured.

The Policy Server User Interface or Administrative UI (as appropriate for your version of SiteMinder) can help you determine the library name and the proper configuration string for a given scheme. Using it, you can create a sample scheme configuration. The Administrative UI lets you set the various scheme parameters and shows you the resulting string.

In SiteMinder Policy Server r12.x, create an authentication scheme with the appropriate template. The **Advanced** section of the creation page shows the **Library** and **Parameter** fields.

Book: For more information, see the CA SiteMinder documentation.

Configuring FCC Pages

To configure the FCC pages to accommodate the backing authentication scheme

1. Include **`ArcotAdapterIntegration.js`** in your code:

```
<script type="text/javascript"
src="js/ArcotAdapterIntegration.js"></script>
```

2. Include the following in your HTML code *before* processing anything related to `smusermsg`:

```
<div id="formDiv" style="display:none">
            <form name=authUsrMsgForm>
             <textarea name=authUsrMsgTxtArea COLS=0
ROWS=0>$$smusrmsg$$</textarea>
            </form>
</div>
```

3. Extract the value of the **`smUserMsg`** variable by using the **`ArcotExtractUserMsg()`** function:

```
smUserMsg =
ArcotExtractUserMsg(document.authUsrMsgForm.authUsrMsgTxtArea.value)
;
```

4. Before submitting the form, call the **`ArcotPrepareSubmit()`** function:

```
ArcotPrepareSubmit(document.Login,
document.authUsrMsgForm.authUsrMsgTxtArea.value);

document.Login.submit();
```

Sample FCC Code

The following is an FCC code sample that illustrates the FCC modifications required for implementing your backing authentication scheme.

```
---------------------------------------------------------------
@username=%USER%
@smretries=0

<!-- SiteMinder Encoding=ISO-8859-1; -->
<html>
<head>
<title>Any Authentication Scheme for SiteMinder</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">

<script type="text/javascript"
src="js/ArcotAdapterIntegration.js"></script>


<script language="javascript" type="text/javascript">

var smUserMsg;
function login() {
         // Process form for submission.
         // ....
         // ....
         // ....
         // Previously
         // document.Login.submit();

         // Change to
         ArcotPrepareSubmit(document.Login, document.authUsrMsgForm.authUsrMsgTxtArea.value);
         document.Login.submit();
}


function ProcessSMUserMsg() {
         // previously
         // smUserMsg = $$smusrmsg$$;

         // change to
         smUserMsg = ArcotExtractUserMsg(document.authUsrMsgForm.authUsrMsgTxtArea.value);



         // Use the variable smUserMsg like before
         // .....
         // .....
         // .....

}


</script>
</head>
```

```
<body>
<h3>Any Authentication Scheme for SiteMinder</h3>

<!--
Arcot Form to get siteminder user msg.
have this always before processing anything related to smusermsg.
-->
<div id="formDiv" style="display:none">
            <form name=authUsrMsgForm>
             <textarea name=authUsrMsgTxtArea COLS=0
ROWS=0>$$smusrmsg$$</textarea>
            </form>
</div>


<script>
ProcessSMUserMsg();
</script>

<form NAME="Login" METHOD="POST">
      <INPUT TYPE="HIDDEN" NAME="SMENC" VALUE="ISO-8859-1">
      <INPUT type="HIDDEN" name="SMLOCALE" value="US-EN">
     <input type="password" name="PIN" size="11" style="margin-left:
1px">
      <input type="hidden" name="target" value="$$target$$">
      <input type="hidden" name="smauthreason"
value="$$smauthreason$$">
      <input type="hidden" name="smagentname"
value="$$smagentname$$">
      <input type="hidden" name="type" value="$$type$$">
      <input type="hidden" name="realmoid" value="$$realmoid$$">
      <input type="hidden" name="USER" value="">
      <input type="hidden" name="PASSWORD" value="">
      <input type="button" value="Login" onClick="login();">
</form>

</body>
</html>
--------------------------------------------------------------
```

Appendix B
Third-Party Software Licenses

This appendix lists the third-party software packages that are used by Arcot Adapter. These include:

Apache

- The Apache Software License, Version 1.1. Copyright (c) 2000 The Apache Software Foundation. All rights reserved.
    - `log4j-1.2.9.jar`
- Copyright © 2009 The Apache Software Foundation, Licensed under the Apache License, Version 2.0.
    - `commons-codec-1.3.jar`
    - `commons-collections-3.1.jar`
    - `commons-httpclient-3.1.jar`
    - `commons-lang-2.4.jar`
    - `commons-pool-1.4.jar`
    - `ibatis-2.3.0.677.jar`

json-lib-0.7.1.jar

Copyright (c) 2002 JSON.org

Json2.js

Copyright (c) 2002 JSON.org

Microsoft SQL Server 2005 JDBC Driver (sqljdbc.jar)

Copyright © 1993-2008 Microsoft Corporation. All rights reserved.

msvcp80.dll

© 2009 Microsoft Corporation. All rights reserved.

msvcr80.dll

© 2009 Microsoft Corporation. All rights reserved.

Oracle Database 10*g* JDBC Driver

Copyright © 1995-2007, Oracle. All rights reserved.

SWFObject

Copyright (c) 2007 Geoff Stearns, Michael Williams, and Bobby van der Sluis. This software is released under the MIT License.

Other Trademarks

- Microsoft®, Windows®, Windows NT®, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Java™ and all Java-based trademarks are trademarks of Oracle® in the United States, other countries, or both. Other company, product, and service names may be trademarks or service marks of others.
- WebSphere is a trademark of IBM in the United States and other countries.DB2 and WebSphere are trademarks of IBM in the United States and other countries.
- BEA WebLogic Server® is a trademarks of Oracle® in the United States and other countries. BEA WebLogic Server® and Solaris SPARC are trademarks of Oracle® in the United States and other countries.

Appendix C
Glossary

ArcotID	Is a secure software credential that supports two-factor authentication. To authenticate to WebFort using ArcotID, the user needs the ArcotID file and the associated password.
Arcot Adapter	Arcot product that increases the security of web resources that are protected by CA SiteMinder.
Delegated Authentication	In this method the authentication and risk processing is done by Arcot WebFort and Arcot RiskFort respectively. The Arcot Shim redirects the user to Customization Engine, which does the authentication and risk processing and then returns the control back to CA SiteMinder. There are two scenarios in this type: 1. Only ArcotID authentication 2. ArcotID authentication and risk evaluation
Customization Engine	Component of Arcot Adapter that interacts with WebFort to authenticate the user. Customization Engine is a set of JavaServer Pages (JSPs).
One-Page Login	It is an authentication workflow, in which the user enters the username and password in `shim.fcc` page. After successful authentication, the user is provided access to protected page.
Personal Authentication Message	A secret message set up by the user when the user is enrolled or when the account is created. It is presented to the user (usually after risk evaluation) to assure the user that the user is interacting with the correct and legitimate server. This is also referred to as "server authentication", because it authenticates the server to the user.
Primary authentication	The authentication mechanism used for the primary or main authentication of users. If only one authentication mechanism is used, then it is the primary authentication mechanism.
Question and Answer	Type of authentication method supported by WebFort. In this method, the user sets questions and answers during enrollment. The user has to answer these security questions during authentication.
RiskFort	RiskFort provides a mechanism to evaluate the risk of a given transaction.

Risk Advice	An action (`ALLOW`, `ALERT`, `DENY`, `INCREASEAUTH`) suggested by RiskFort to the calling application, after evaluating the risk of a transaction.
Risk Score	RiskFort generates a score depending on the evaluation result. The score can be a number from `0` through `100`. The greater the number, the higher the risk.
Roaming Download	The process of downloading ArcotID on multiple systems from the WebFort Server.
Secondary Authentication	This is a step-up authentication, which the user has to perform in any of the following cases: • If the risk advice is `INCREASEAUTH` • If the user is downloading ArcotID from WebFort **Note:** QnA method is used as a secondary authentication method. You can use any customized authentication methods for this purpose.
Shim	Component of Arcot Adapter that redirects the user to other components for authentication and risk evaluation.
Token Server	Component of Arcot Adapter that generates the token for the user to keep track of the user information.
Two-Page Login	It is an authentication workflow, in which the user enters the username first and after the secondary authentication the user enters the password. If authenticated successfully, the user is granted access to the protected resource The `shim2.fcc` page is used in this authentication workflow.
WebFort	WebFort provides two-factor software-based strong authentication.