Solaris Smartcard Administration Guide

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part No: 806–7010–05
August 2001

011025@2471

Contents

Preface

A Solaris™ Smartcard enables a user to log in securely to the Solaris 9 desktop environment. A smart card is a plastic card that allows you to access a system by inserting a programmable card into a card reader. This guide explains how to configure systems and smart cards for this form of authentication. It also explains how to use a smart card after it has been configured.

Who Should Use This Book

The Solaris Smartcard Administration Guide is intended for the system administrator who sets up and administers the Solaris Smartcard environment. This guide assumes that you have a solid knowledge of authentication and related network security concepts. If you need an introduction to these concepts, see "Managing System Security (Overview)" in *System Administration Guide, Volume 2*.

If you are merely a user of a Solaris Smartcard, you do not need to read this book. Simply insert your smart card in your card reader and enter your personal identification number (PIN) when prompted to do so.

Related Books

Solaris Smartcard can be used in conjunction with any Solaris administration tools or Solaris commands and procedures. Refer to one or more of the following for additional information on Solaris installation or administration procedures:

- *(SPARC Platform Edition) Installation Guide*
- *System Administration Guide, Volume 1*
- *System Administration Guide, Volume 2*
- *System Administration Guide, Volume 3*
- Other software documentation that you received with your system

Ordering Sun Documents

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at `http://www1.fatbrain.com/documentation/sun`.

Accessing Sun Documentation Online

The docs.sun.com[SM] Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Typographic Conventions

The following table describes the typographic conventions used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

TABLE P–1 Typographic Conventions *(Continued)*

Typeface or Symbol	Meaning	Example
AaBbCc123	What you type, contrasted with on-screen computer output	`machine_name% su` `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **rm** *filename*.
AaBbCc123	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

CHAPTER 1

Solaris Smartcard Overview

This chapter provides an overview of Solaris Smartcard features, supported smart cards and card readers, and planning information.

Solaris Smartcard Features

A Solaris Smartcard provides a somewhat more secure method for logging in to the Solaris desktop environment than is provided by the standard UNIX login. Information stored on the smart card verifies the identity of the user during login. A user who cannot provide the login information that is on the smart card is denied access to the desktop. The Solaris Smartcard software:

- Implements the smart card framework, which is based on the OCF1.1 standard
- Supports a variety of card readers
- Supports three widely-used smart cards
- Allows management from the Solaris Smartcard Console or the Solaris command line
- Protects login to the desktop environment through use of the PIN authentication method
- Lets a user store security credentials directly onto the card (Java cards only)

Solaris Smartcard Requirements

To use the Solaris Smartcard software, you need:

- A SPARC system running the Solaris 8 or Solaris 9 release.
- A supported internal or external card reader and smart cards.

Solaris Smartcard supports the following smart cards and card readers.

- Payflex card
- iButton card
- Cyberflex card
- Sun SCRI External Serial Card Terminal Reader
- Sun SCRI Internal Card Terminal Reader

Smartcard Login

Secure desktop environments can be protected by requiring users to log in with a configured Solaris Smartcard. The following sequence explains what happens in the login process:

1. The user inserts the card into the card reader.
2. The `dtsession` daemon prompts the user to enter a personal identification number (PIN), and then compares the typed PIN with the PIN stored on the card.
3. If the typed PIN and the PIN stored on the card match, the password database specified in the system's `/etc/nsswitch.conf` file (NIS, NIS+, or local files) is searched for this password.
4. If the password is found in the system's password database, the user is considered authenticated and is allowed to log in to the desktop.

Package Descriptions

The following table lists the Solaris Smartcard packages added during a Solaris 9 installation.

TABLE 1–1 Solaris Smartcard Packages

Package Name	Description
SUNWjcom	Java Communications API for smart card support - Java code and Native code
SUNWjcomx	Java Communications API for smart card support - Native code (64–bit)
SUNWjib	Dallas Semiconductor serial iButton OCF Card Terminal Driver
SUNWocf	Open Card Framework - core libraries and utilities
SUNWocfr	Open Card Framework - configuration files
SUNWocfh	Open Card Framework - header files
SUNWocfx	Open Card Framework - core libraries (64–bit)
SUNWpamsc	Pluggable Authentication Module for smart card authentication
SUNWpamsx	Pluggable Authentication Module for smart card authentication (64–bit)
SUNWscgui	Solaris Smartcard Console
SUNWscmos	Pluggable Authentication Module for smart-card authentication
SUNWscmsc	Sun SCRI OCF Card Terminal Driver

To remove a package, use the standard `pkgrm` command. Reinstall the package using the `pkgadd` command.

See "Software Administration (Tasks)" in *System Administration Guide, Volume 1* for information on using these commands.

CHAPTER 2

Getting Started With Solaris Smartcard

This chapter shows the administrator how to set up an initial Solaris Smartcard configuration:

Starting or Restarting the Smartcard Console

The Smartcard Console is the graphical user interface (GUI) used to manage the Solaris Smartcard software.

To Start the Smartcard Console from the Command Line

1. **Log in as root**

Note – If you log in as a regular user, you can run Smartcard, but you can only perform two tasks: Load Applets and Configure Applets.

---

2. **Start the Smartcard Console:**

```
# /usr/dt/bin/sdtSmartCardadmin &
```

To Start the Smartcard Console from CDE

1. **Log in as root to the Common Desktop Environment (CDE).**

   If you are currently running CDE under your login name, exit CDE and log in as root.

Note – If you log in as a regular user, you can run Smartcard, but you can only perform two tasks: Load Applets and Configure Applets.

---

2. **On the CDE control panel, click the up arrow for the Applications subpanel.**

   By default, the Text Note icon, a pinned note with a pencil above it, represents the Applications subpanel.

3. **Select Applications to display the Application Manager.**

4. **Double-click the System_Admin icon in Application Manager.**

5. **Double-click the Smart Card icon to start the Smartcard Console.**

   You may have to scroll down to find the Smart Card icon.

Note – You can also start the Smartcard Console from the desktop Workspace menu by selecting Tools and then sdtsmartcardadmin.

Setting Up a Desktop for Smartcard Login

To set up Smartcard login for the desktop of a Sun workstations running the Solaris 9 operating environment, do the tasks described below.

To Enable a card reader

Although your new workstation has an internal card reader, you must enable it before it can be used. To enable the Sun internal card reader, do the following:

1. **Click Card Readers in the Smartcard Console's Navigation pane.**

   The Add Reader icon is displayed in the Console pane. Icons for any enabled card reader types are also displayed.

2. **Double-click Add Reader in the Console pane.**

   The Add Reader dialog box is displayed.

3. **Double-click the type of card reader you want to add or select it and click OK.**

   To enable the Sun internal card reader, select Sun SCRI Internal Card Terminal Reader. The CardReaders dialog box is displayed.

4. **Select the Basic Configuration tab.**

5. **Click the down arrow under Device Port.**

6. **Select the port that the card reader is attached to.**

7. **Click OK.**

8. **Restart `ocfserv`, if prompted to do so.**

To Activate Card Services

1. **Click Card Services in the Navigation pane.**

   Icons for existing card types are displayed in the Console pane. An "X" on top of an icon indicates that the card service is inactive.

2. **Double-click the icon for the card service to be activated.**

   The Card Services dialog box is displayed. If the selected smart card service is inactive, you will see an "X" next to the word "Inactive" below the list of services.

3. **Click the radio button that reads "Activate** *card type* **services."**

4. **Click OK.**

Note – If the selected smart card service is already active and you want it to remain active, be sure the radio button that reads "Keep *card type* services activated" is selected and click OK. If you want to deactivate services for the selected card, select the radio button that reads "Deactivate *card type* services" and click OK.

To Add Support for a New Card Type (New ATR)

To use a new type of smart card, you have to provide its Answer to Reset (ATR) property to `ocfserv`.

1. **Insert the smart card with the new ATR in the card reader.**

2. **In the Navigation pane, select Smart Cards.**

3. **Double-click the icon representing the type of card currently inserted.**

   The Smart Card dialog box displays a list of the known ATRs for this card type.

4. **If this is a new ATR, click Add.**

   The Add ATR dialog box is displayed, with the ATR of the card inserted in the card reader shown in the "Inserted Card's ATR" listbox.

5. **Select the ATR of the inserted card or type the new ATR in the New ATR field.**

   You can find the new ATR value in the smart card product literature.

6. **Click OK in the Add ATR dialog box.**

   The new ATR is added to the list in the Smart Card dialog box.

7. **Click OK in the Smart Card dialog box to activate the change.**

To Load the Smartcard Applet to a Smart Card

Do the following to load the Solaris Smartcard applet (`SolarisAuthApplet`) to a smart card.

Note – This is a task that can be performed by an end user.

---

1. **Insert the smart card into the reader.**
2. **Select Load Applets icon in the Navigation pane.**
3. **Double-click the SolarisAuthApplet icon in the Console pane**

   The Load Applets dialog box is displayed. Available applets for various card types are displayed in the left listbox.
4. **Select the card type you want to initialize.**

   Choices include CyberFlex, IButton, and PayFlex.
5. **Click the arrow between the two listboxes.**

   The selected applet is copied to the Pending Applet Installations listbox, with a check in the checkbox and the name of the smart card displayed. If no card or the wrong smart card is inserted in the card reader, "No compatible devices inserted" is displayed. Insert the appropriate card.
6. **Click Install.**

   A window labeled "Loading Applet to Device" is displayed. It takes a minute or so for the applet to load. When the installation is complete, a window with a confirmation message ("Applet Installation Successful") displays.

   For instructions on loading an applet from the Command-line, see "How to Load an Applet Onto a Smart Card (Command Line)" on page 32.
7. **Click OK to dismiss the confirmation window.**

   The card now stores default values. If the card stored different PIN or user profile values, those values have been overwritten. See "PIN Property" on page 34 and "User and Password Properties" on page 35 for more information.

To Change the PIN and Set Up a User Profile

Do the following to change the PIN on a card and set up a user profile for the authorized user of the card. For more information, see "How to Create User Information on a Smart Card" on page 34.

Note – This is a task that can be performed by an end user.

1. **Insert the smart card you want to configure into the card reader.**
2. **Select Configure Applets in the Navigation pane.**
   The icon for the type of card in the reader is displayed in the Console pane.
3. **Double-click the card icon in the Console pane.**
   The Configure Applets dialog box is displayed.
4. **Select `SolarisAuthApplet` in the listbox.**
   The `SolarisAuthApplet` configuration folders appear on the right side of the dialog box, represented by tabs labeled PIN, User Profiles, RSA Key, and PKI Cert. Only PIN and User Profiles are described here. See "How to Create a Private Key on a Smart Card (Command Line)" on page 48 for information about PKI.
5. **Select the PIN tab.**
   This is optional. You could leave the current PIN on the card. Skip the next three steps if you do not want to change the PIN.
6. **Type and retype a new PIN.**
   A PIN can contain up to eight characters.
7. **Click Change.**
   A popup window labeled "Change PIN" is displayed.
8. **Enter the previous PIN in the pop-up window and click OK.**
   The default PIN, loaded on the card when the `SolarisAuthApplet` was installed on the card, is `$$$$java`.
9. **Select the User Profiles tab in the Configure Applets dialog box.**
10. **Type `dtlogin` in the User Profile Name field.**
    This represents the CDE desktop.
11. **Type user name in User Name field.**
    This is the username of the person who will be using the card. The username cannot be more than eight characters long. Click Get to determine the current username associated with the card.
12. **Type password in Password field.**
    This is the password associated with the username typed above. This password must be in the password database defined by a system's `/etc/nsswitch.conf file` (NIS, NIS+, or local files). The password cannot be more than eight characters long.

Note – If the user's password is changed in the passwd file after you have configured the smart card, you or the user must repeat these steps to store the new password on the smart card. It is not updated automatically.

13. **Click Set.**

    The Set User Profile popup is displayed, asking for the current PIN.

14. **Type the PIN and click OK.**

    The new username and password are stored on the card.

15. **Click OK to dismiss the dialog box.**

To Activate Smartcard Operation

Do the following to activate Solaris Smartcard operation on a system. This must be done on each system that will use Smartcard authentication.

1. **Select OCF Clients in the Navigation pane.**

   The Desktop icon is displayed in the Console pane.

2. **Double-click the Desktop icon.**

   The Configure Clients dialog box is displayed.

3. **Select the Cards/Authentications tab in the dialog box.**

   The three supported smart cards — CyberFlex, IButton, and PayFlex — are listed in the listbox at the left.

4. **Select the radio button labeled "Activate Desktop's Smart Card capabilities."**

Note – As soon as you click OK in the Configure Clients dialog box, Smartcard is activated. Be sure you have a working card reader on the system and a smart card configured with your username and password. And be sure you know the PIN on the card or you will be locked out of the system. If you cannot access your system because of Smartcard, `rlogin` to the system and disable Smartcard by typing, as superuser: `smartcard -c disable`.

5. **Click Apply or OK.**

   Solaris Smartcard is now enabled on the system. See "To Enable Smart Card Operations (Command Line)" on page 38 for command-line instructions for enabling Smartcard operation.

Note – You can disable Smartcard from the Configure Clients dialog box by selecting the radio button labeled "Deactivate Desktop's Smart Card Capabilities" and clicking OK.

6. **Exit CDE to activate the change.**

To Set Smartcard Timeouts

1. **Select OCF Clients in the Navigation pane.**
2. **Double-click the Desktops icon in the Console pane.**
   The Configure Clients dialog box is displayed.
3. **Select the Timeouts tab in the dialog box.**
4. **Adjust the timeouts by sliding the indicator for each timeout with the mouse.**
   The Card Removal timeout specifies the number of seconds the desktop waits after a smart card is removed before locking the screen.
   The Reauthentication timeout specifies the number of seconds the Reauthentication screen is displayed.
   The Card Removal Logout Wait specifies the number of seconds the desktop waits for a smart card to be reinserted when the Reauthentication screen is displayed. If the card is not reinserted in time, the user is logged out. Note that this timeout is relevant only when Reauthenticate After Card Removal (in the Options tab) is set to False.
5. **Click Apply or OK.**
6. **Exit CDE to activate the change.**

To Set Card Removal Options

1. **Select OCF Clients in the Navigation pane.**
2. **Double-click the Desktops icon in the Console pane.**
   The Configure Clients dialog box is displayed.
3. **Select the Options tab in the dialog box.**

4. **Click the checkboxes to toggle them.**

   If Ignore Card Removal is checked, a user is not logged out when a smart card is removed from the reader. Otherwise, a user is logged out when a smart card is removed.

   If Reauthenticate After Card Removal is checked, a user is logged out when a card is removed. If it is not checked, the Card Removal Logout Wait setting (in the Timeouts tab) determines what happens.

5. **Click Apply or OK.**

6. **Exit CDE to activate the change.**

The `ocfserv` Server

The OpenCard Framework (OCF) server, `ocfserv`, is the process that manages smart card communications on your system.

You are prompted to restart `ocfserv` after you have used the Smartcard Console to add or remove a card reader.

If `ocfserv` is not running, start or restart the Smartcard Console.

CHAPTER 3

Setting Up Card Readers

This chapter describes the procedures for setting up and maintaining card readers of various types.

Supported Card Readers

Solaris Smart Cards supports two external card readers, the iButton and the Sun SCRI External Reader 1, and an internal card reader, the Sun SCRI Internal Card Reader 1.

The following table shows the supported card readers and the corresponding values you need to supply to add them.

TABLE 3–1 Card Readers Supported

Reader Type	Card Terminal Factory Name	Reader Model Name
Sun SCRI External Card Reader 1	`com.sun.opencard.terminal.scm.SCMStc.SCMStcCardTerminalFactory`	`SunSCRI`
iButton	`com.ibutton.oc.terminal.jib.iButtonCardTerminalFactory`	`DS1402`
Sun SCRI Internal Card Reader 1	`com.sun.opencard.terminal.scm.SCMI2c.SCMI2cCardTerminalFactory`	`SunISCRI`

Adding a Card Reader (Command Line)

You add a card reader by using the `smartcard -c admin` command with the following syntax:

`smartcard -c admin -t terminal -j` *card_terminal_factory_name* `-x add -d` *device_pathname* `-r` *user_friendly_reader_name* `-n` *card_reader_model*

`-c admin`	Indicates that you are viewing or modifying OCF properties.
`-t terminal`	Indicates that you are about to configure a card reader.
`-j` *card_terminal_factory_name*	Defines the card terminal factory name of the card reader type. See the specific Card Terminal Factory Name in the procedures below.
`-x add`	Indicates that you want to add a card reader.
`-d` *device_pathname*	Specifies the device port where you have plugged in the card reader.
`-r` *user_friendly_reader_name*	Specifies a unique name for the reader.
`-n` *reader_model_name*	Designates the model name of the card reader. See the specific card reader model name in the procedures below.

Refer to the `smartcard`(1M) man page for more information.

▼ How to Add an iButton Reader (Command Line)

1. **Become superuser on the system where you are attaching the card reader.**
2. **Ensure that you have already physically attached the external card reader to the system.**

   Physically attach the external smart card reader to the serial port, following instructions in the card reader documentation.
3. **Add the iButton reader by typing the following command on one line.**

   For example:

```
# smartcard -c admin -t terminal
-j com.ibutton.oc.terminal.jib.iButtonCardTerminalFactory
-x add -d /dev/cua/b -r MyButtonReader -n DS1402
```

`-c admin`	Indicates that you are viewing or modifying OCF properties.
`-t terminal`	Indicates you are configuring a card reader.

`-j com.ibutton.oc.terminal.jib.iButtonCardTerminalFactory`	Identifies the card terminal factory name of the iButton reader. Be careful to type the card terminal factory name following `-j` option exactly as shown in the procedure above, with no spaces or returns between characters.
`-x add`	Indicates that you want to add a card reader.
`-d /dev/scmi2c0`	Defines the device port where the card reader is attached.
`-r` *MyButtonReader*	Specifies a unique name for the iButton reader.
`-n DS1402`	Indicates the model name for the iButton card reader.

4. **Stop** `ocfserv`**.**

```
# pkill ocfserv
```

The `ocfserv` process is restarted the next time you use the Smart Card Console or the `smartcard` command.

▼ How to Add a Sun SCRI External Card Reader 1 (Command Line)

1. **Become superuser on the system where you are attaching the card reader.**
2. **Ensure that you have already physically attached the external card reader to the system.**

   Physically attach the external smart card reader to the serial port, following instructions in the card reader documentation.

3. **Add the Sun SCRI External Card Reader 1 by typing the following command on one line.**

   For example:

```
# smartcard -c admin -t terminal
-j com.sun.opencard.terminal.scm.SCMStc.SCMStcCardTerminalFactory
-x add -d /dev/cua/b -r MyExternalReader -n SunSCRI
```

`-c admin`	Indicates that you are viewing or modifying OCF properties.
`-t terminal`	Indicates you are configuring a card reader.

`-j`
 	
`com.sun.opencard.terminal`
`.scm.SCMStc.SCMStcCard`
`TerminalFactory`
	The card terminal factory name of the Sun SCRI External Card Reader 1.
	
	Be careful to type the card terminal factory name following `-j` option exactly as shown in the procedure above, with no spaces or returns between characters.

`-x add`	Indicates that you want to add a card reader.
`-d /dev/scmi2c0`	Defines the device port where the card reader is attached.
`-r` *MyExternalReader*	Specifies a unique name for the SCRI External Card Reader 1.
`-n SunSCRI`	Indicates the model name for the SCRI External Card Reader 1.

4. **Stop** `ocfserv`**.**

   ```
   # pkill ocfserv
   ```

   The `ocfserv` process is restarted the next time you use the Smart Card Console or the `smartcard` command.

▼ How to Add a Sun SCRI Internal Card Reader 1 (Command Line)

1. **Become superuser on the system where you are attaching the card reader.**
2. **Add the Sun SCRI Internal Card Reader 1 by typing the following command on one line.**

   For example:

   ```
   # smartcard -c admin -t terminal
   -j com.sun.opencard.terminal.scm.SCMI2c.SCMI2cCardTerminalFactory
   -x add -d /dev/scmi2c1 -r MyInternalReader -n SunISCRI
   ```

`-c admin`	Indicates that you are viewing or modifying OCF properties.
`-t terminal`	Indicates you are configuring a card reader.

`-j`
`com.sun.opencard.terminal.`
`scm.SCMI2c.SCMI2cCard`
`TerminalFactory`
	The card terminal factory name of the Sun SCRI Internal Card Reader 1.
	
	Be careful to type the card terminal factory name following `-j` option exactly as shown in the procedure above, with no spaces or returns between characters.

`-x add`	Indicates that you want to add a card reader.
`-d /dev/scmi2c0`	Defines the device port where the card reader is attached. For example, `/dev/scmi2c`*n*, where *n* in `scmi2c`*n* is the *n*th SunISCRI reader on the system.
`-r` *MyInternalReader*	Specifies a unique name for the SCRI Internal Card Reader 1.
`-n SunISCRI`	Indicates the model name for the SCRI Internal Card Reader 1.

3. **Stop** `ocfserv`**.**

```
# pkill ocfserv
```

The `ocfserv` process is restarted the next time you use the Smart Card Console or the `smartcard` command.

Removing a Card Reader

You might need to remove an external card reader from a system, for example, when a user no longer needs to use a smart card, or when you want to move the card reader to another system. Be sure to remove the card reader logically before you disconnect the physical device.

▼ How to Remove a Card Reader (Console)

1. **Click Card Readers on the Navigation pane.**
2. **Select the card reader in the Console pane that you want to remove.**
3. **Select Remove Terminal from the Action menu.**
4. **Click OK to remove the card reader.**
5. **Restart** `ocfserv`**, if prompted.**

▼ How to Remove a Card Reader (Command Line)

1. **Become superuser on the system with the card reader to be removed.**

2. **Remove the card reader.**

   # **smartcard -c admin -t terminal -r** *user_friendly_reader_name* **-x delete**

3. **(Optional) Unplug the external card reader from the port.**

4. **Stop** `ocfserv`**.**

   # **pkill ocfserv**

   The `ocfserv` process is restarted the next time you use the Smart Card Console or the `smartcard` command.

CHAPTER 4

Setting Up a Smart Card

This chapter provides an overview of setting up a smart card.

You can set up a smart card either with the Smart Card Console or the command line.

The tasks in this chapter assume that you have identified how you will implement smart cards at your site. The tasks also assumes that you have set up a card reader on all systems that will use smart cards.

Activating or Deactivating Card Services

The Card Services dialog box lets you verify that services for a particular card are active. You can also use this dialog box to deactivate a card service.

Adding or Changing the ATR on a Smart Card

The answer-to-reset (ATR) property contains numeric values that identify the smart card version. Smart card manufacturers supply the ATR property.

Supported smart cards have the following ATR property values:

```
PayFlex.ATR      = 3B690000579202010100010 0A93B691100000057920201010001 00
IButton.ATR      = 008F0E000000000000000000000040000034909000
CyberFlex.ATR    = 3B169481100601810F 3B169481100601811F
```

When you set up smart cards, you need to identify the ATR on the smart card to the OCF server.

You need to change the answer-to-reset property (ATR) on a system if the manufacturer of the smart card used by your site issues a new card type with a different ATR. Change this property on every system that needs to accept the new card.

You might also need to add an ATR if you are loading an applet from the Smart Card Console and the following message is displayed:

```
No compatible devices inserted
```

See "To Add Support for a New Card Type (New ATR)" on page 18 for step-by-step instructions on adding or changing the ATR on a new smart card.

Loading the `SolarisAuthApplet` Applet

You must add the default `SolarisAuthApplet` applet to the card before you can add the user profile information.

▼ How to Load an Applet Onto a Smart Card (Command Line)

Use this command to load the `SolarisAuthApplet` applet onto all card types supported by Solaris Smart Cards.

1. **Insert the smart card into the reader.**
2. **Become superuser.**
3. **Load the `SolarisAuthApplet` applet onto a smart card.**

   ```
   # smartcard -c load -i /usr/share/lib/smartcard/SolarisAuthApplet.capx
   ```

   When the `smartcard -c load` finishes, the following message displays:

   ```
   Operation successful.
   ```

▼ To Change a PIN on a Smart Card – Command Line

Caution – Be sure to type the new PIN correctly because you will not be prompted to confirm it.

1. **Insert the smart card into the card reader.**
2. **Become superuser.**
3. **Change the PIN.**

```
# smartcard -c init -A A000000062030400 -P '$$$$java' pin=001234
```

Caution – Enclose the default PIN, `$$$$java`, or any PIN containing shell special-characters (such as `$`) within single quotes. Otherwise, the shell tries to interpret the PIN as a variable, and the command fails.

Creating User Information on a Smart Card

After the default applet has been loaded, create the user profile information on the card. The user profile information identifies personal information about the user, such as login name and password, a personal identification number (PIN), and the protected application.

See "How to Create User Information on a Smart Card" on page 34 for step-by-step instructions on creating user information on the smart card.

▼ How to Create User Information on a Smart Card

Example—Creating User Information on a Smart Card (Command Line)

This command is appropriate for all smart cards devices supported by Solaris Smart Cards. Make sure the card is in the card reader.

Set the PIN, login name, password, and application for the card by typing the following on one line.

```
# smartcard -c init -A A000000062030400 -P '001234' user=nigel
password=changeme application=dtlogin
```

Defining Authentication Properties on a Smart Card

You set the properties on each smart card based on the user's requirements, your site's security policies, and the limitations of the type of smart card used. Using the Configure Applets dialog box, you define corresponding properties for each smart card. The client and server programs on the system read the properties on the smart card to determine whether to give the user access to a particular application.

Note – These properties apply only to cards initialized with the `SolarisAuthApplet` applet provided with Solaris Smart Cards. If your site uses a different smart card applet, the available properties might differ. Refer to the `smartcard(1M)` man page for more information.

PIN Property

The PIN property is an authentication property that defines a personal identification number (PIN) for the card. The default PIN created on the card is `$$$$java`. Either you or the user can change `$$$$java` to a personalized PIN. Consider giving all users at your site the same default PIN name (for example, `changeme`). Then make sure each user changes the PIN to a value known only to that user.

See "To Change a PIN on a Smart Card – Command Line" on page 33for step-by-step instructions on changing the PIN on a smart card.

Users can later change this PIN.

User and Password Properties

The user and password properties are authentication properties that identify the user and associate the user with the smart card's PIN. To set these properties, you must know the user's login name and password.

How the User and Password Properties Work

On systems using the default authentication mechanism of PIN, `ocfserv` verifies the authenticity of the PIN as explained in "How the PIN Works" on page 37. Next, `ocfserv` reads the user and password properties on the card. If the password on the smart card matches the user's entry in the system's password database, `ocfserv` gives the user access to the application.

Application Property

Use the application authentication property to designate which applications the user needs to log in to with a login name and password. For example, to require a smart card login to the desktop, you must specify `dtlogin` as the application associated with the login name and password on the card. You can also require a smart card login for an application specific to your site, such as a financial package or personnel database, by specifying its name as the application property.

Before initializing an application on the card, find out which applications a user needs to access through smart card authentication. This step is particularly important when preparing a smart card for a system administrator or other user who might need to log in to an application as root or another restricted login name.

Note – Payflex cards do not support multiple profiles; they cannot be used in cases where a user needs to login to the desktop and one or more secure applications or to use multiple user names.

How the Application Property Works

The application property on the smart card works in tandem with the other authentication properties. For example, suppose you initialized a smart card for user Frank with the following information:

- `A000000062030400` – The `SolarisAuthApplet` applet.
- `'$$$$java'` – The default PIN for this card, which user Frank can change later.
- `dtlogin` – The application requiring the smart card login.
- `frank` – The name that Frank must provide to log in to the desktop.
- `changeme` - The password that Frank must type to log in to the desktop.

The preceding information would be entered on the command line, as follows:

```
# smartcard -c init -A A000000062030400 -P '$$$$java' application=dtlogin
user=frank password=changeme
```

When Frank inserts his card into the reader and tries to log in to the desktop (`dtlogin`), `ocfserv` reads the card to determine whether any authentication properties are associated with `dtlogin`. The `ocfserv` server finds that the user and password properties are associated with `dtlogin`.

The `ocfserv` server prompts Frank for his PIN, and the typed PIN is compared with the PIN stored on the smart card assigned to the `dtlogin` application. Also, `ocfserv` uses the login name and password on Frank's card, along with the passwords in the system's password database, to verify that Frank is who he claims to be. If these properties match, Frank is logged in to the desktop.

Setting Up the Default Authentication Mechanism for the OCF Server and Client Applications

The `authmechanism` property defines the authentication mechanism that the client application uses on the local system. Solaris Smart Cards offers three authentication mechanisms:

- **Password** – A password associated with the individual's login name must be on the smart card or typed when the user tries to access a client application.
- **PIN** – An individual's personal identification number (PIN) must be on the smart card or typed when the user tries to access a client application.
- **Challenge-Response** – A challenge-response authentication sequence must occur between the system and the smart card before the individual can access the client application.

The default `authmechanism` property is Pin.

How the PIN Works

The default authentication mechanism for `ocfserv` and client applications is PIN. In this scenario, the user tries to log in to an application, such as the desktop. The application requests that the user type a PIN.

The `ocfserv` server verifies the authenticity of the user by comparing the PIN the user types in to the PIN on the smart card. If the PINs match, either the user is given access to the application, or `ocfserv` reads additional authentication properties on the card.

Default Authentication for Client Applications

You set up authentication on the local system so that it denies access to anyone who does not have the corresponding authentication mechanisms on the smart card. The server's authentication mechanisms are the default mechanisms used by `ocfserv` during smart card login.

If the user tries to access a client application with a different sequence set, `ocfserv` uses the client authentication mechanisms during login.

`ocfserv` authentication mechanisms must be activated before you can configure client authentication properties. By default, all authentication mechanisms supported by Solaris Smart Cards are activated when the Solaris 9 release is installed. These mechanisms are:

- Password
- PIN
- Challenge-Response

You must define properties for the default smart cards and the default authentication sequence used by the individual OCF client application. You protect sensitive applications running on the local system by configuring them to require login with a smart card. The default application to be protected is `dtlogin`, the application that controls logging into the Common Desktop Environment (CDE).

An application does not need to have the same authentication sequence as `ocfserv`. The client authentication sequence takes precedence over a different authentication sequence assigned to `ocfserv`. For example, you can configure "password" as the default authentication mechanism for `ocfserv`. However, you might want to add "PIN authentication" into the authentication sequence for anyone trying to access a client application, such as the Solaris desktop.

Enabling Smart Card Operations

The final step in setting up a smart card is to enable smart card operations.

See "To Enable Smart Card Operations (Command Line)" on page 38 for step-by-step instructions on enabling smart card operations.

You cannot log in through `dtlogin` if you enable smart cards and either of the following conditions is true:

- You do not have a working smart card, or
- You have not configured a smart card successfully

If you enable smart card operations before you have established a working smart card configuration, do the following:

1. Login in to the system remotely with the `rlogin` command.
2. Become superuser (root).
3. Disable smart card operations.

```
# smartcard -c disable
```

▼ To Enable Smart Card Operations (Command Line)

The user must use the accepted smart card for the system and possibly type a PIN to successfully log in to this system after smart cards are enabled.

1. **Become superuser on each system to be used in smart card operations.**
2. **Stop the desktop.**

```
# /etc/init.d/dtlogin stop
```

3. **Turn on smart card operations.**

```
# smartcard -c enable
```

4. **Restart the desktop.**

```
# /etc/init.d/dtlogin start
```

CHAPTER 5

Additional OCF Server and Client Configuration (Tasks)

This chapter describes additional OCF server and client configuration tasks that you might want to perform after initially setting up a smart card. You can complete these tasks from the Smart Card Console or the command line.

Additional Server Configuration Tasks

OCF server properties define operations of `ocfserv` on each system. You can change these properties using either the OCF Server Configuration dialog box or the `smartcard -c admin` command. To change OCF server properties using the command line, use the following basic steps.

1. Become superuser on the system where you want to change properties.
2. Change the default server property.

```
smartcard -c admin -x modify "property_name=property_value"
```

`-x modify`	Indicates that you want to modify a property.
property_name=property_value	Represents the property to be modified and the value you want to assign to it.

▼ How to View OCF Server and Client Properties (Command Line)

1. **Become superuser on the system that you want to configure.**

2. **Display the configurable properties.**

   # **smartcard -c admin**

   Your screen should look more-or-less like the following:

```
Client Properties:
  ClientName.PropertyName   Value
  ----------------------    -----
  default.validcards      = CyberFlex IButton PayFlex
  default.authmechanism   = Pin=UserPin
  default.defaultaid      = A000000062030400

Server Properties:

  PropertyName             Value
  ------------             -----
  authmechanism           = PIN
  OpenCard.terminals      = com.sun.opencard.terminal.scm.
        SCMStc.SCMStcCardTerminalFactory|MySCM|SunSCRI|/dev/cua/b
  ocfserv.protocol        = rpc
  PayFlex.ATR             = 3B690000579202010100010 0A9 3B691100000057920201010001 00
  authservicelocations    = com.sun.opencard.service.auth
  OpenCard.services       = com.sun.opencard.service.cyberflex.CyberFlexServiceFactory
        com.sun.opencard.service.ibutton.IButtonServiceFactory
        com.sun.opencard.service.payflex.
PayFlexServiceFactory abc.class com.sun.services.scm.SCMStcCardTerminalFactory
  initializerlocations    = com.sun.opencard.cmd.IButtonInit
  IButton.ATR             = 008F0E0000000000000000000004000034909000
  cardservicelocations    = com.sun.opencard.service.common
  CyberFlex.ATR           = 3B169481100601810F 3B169481100601811F
  country                 = US
  debugging.filename      = /tmp/ocf_debugfile
  language                = en
  debugging               = 0
```

▼ How to Change the Valid Smart Cards for the Server (Console)

By default, all three card types are considered valid by the OCF server.

1. **Select OCF Server from the Navigation pane.**
2. **Double-click the icon representing the local system.**
3. **Select Valid Smart Cards from the Available Resources list.**
4. **Click the check box in the list that you want to deselect or select as a valid smart card type.**

5. **Activate the card services for the cards you selected as being valid.**
   For instructions on activating card services, see [link].
6. **Click Apply or OK.**

▼ How to Change the Default Smart Card for the Server (Console)

1. **Select OCF Server from the Navigation pane.**
2. **Double-click the icon representing the local system.**
3. **Select Default Smart Cards from the Available Resources list.**
4. **Click the check box in the list that represents the card type you want as the default.**
   None is selected by default, which means there is no default smart card type.
5. **Activate the card services for the cards you selected as the default.**
   For instructions on activating card services, see [link].
6. **Click Apply or OK.**

Additional Client Configuration Tasks

Before performing the tasks in this section, you must have:

- Configured at least one card reader for the system.
- Activated card services on the system.
- Decided on the default authentication mechanism to use at your site and the sequence in which each mechanism must occur.
- Determined which applications running on the system must be protected by smart card login.

▼ How to Define the Default Smart Card for the Client (Console)

1. **Select OCF Clients from the Navigation pane.**
2. **Double-click the CDE icon.**
3. **Select the Defaults folder.**
4. **Select Smart Card from the Available Resources list.**
5. **Select the radio button for the smart card that will serve as the default for the client. You can select only one default card type.**

Note – The card type you select for the default card type must also be defined as a valid card. See the "How to Change the Default Smart Card for the Server (Console)" on page 41.

---

6. **Click Apply or OK.**

▼ How to Define the Default Smart Card Reader for the Client (Console)

1. **Select OCF Clients from the Navigation pane.**
2. **Double-click the CDE icon.**
3. **Select the Defaults folder.**
4. **Select Card Reader from the Available Resources list.**
5. **Select the radio button for the card reader that will serve as the default for the client. You can pick only one default card reader.**

Note – The card reader you choose must accommodate the default smart card you previously defined.

---

6. **Click Apply or OK.**

▼ How to Change the Default Client Authentication Sequence for Valid Cards (Console)

1. **Select OCF Clients from the Navigation pane.**
2. **Double-click the CDE icon.**
3. **Select one or more valid smart card types on the Smart Cards Used list.**

   The *card_name* Authentications list shows PIN as the default authentication mechanism assigned by Solaris Smart Cards. The Tag column lists a lookup value assigned to the application.
4. **Click Add to display a combo box.**
5. **Pull down on the arrow to display the authentication mechanisms active on the OCF server, and choose additional mechanisms as needed.**
6. **Repeat this procedure for each card type selected as a valid card.**
7. **Click Apply or OK.**

▼ How to Change the Valid Smart Cards for a Client Application (Command Line)

1. **Become superuser.**
2. **Change the default valid cards by typing the following on one line:**

```
# smartcard -c admin -a default -x modify validcards="IButton|CyberFlex|PayFlex"
```

IButton|*CyberFlex*|*PayFlex*	Indicates any one or a combination of these values.

For example, to define the valid smart card types as CyberFlex and Payflex for all applications, type the following on one line:

```
# smartcard -c admin -a default -x modify validcards="CyberFlex Payflex"
```

▼ How to Assign a Default Smart Card to a Client Application (Command Line)

The `application_name.authmechanism` property enables you to assign an authentication mechanism to a particular application.

1. **Become superuser on the system whose client properties you want to modify.**
2. **Assign a default smart card type to an application.**

```
# smartcard -c admin -a application_name -x add defaultcard=card_name
```

application_name	The application for which you want to define a default smart card type.
card_name	The smart card type that must be used to log in to this application, either `CyberFlex`, `PayFlex`, or `IButton`.

For example, to define iButton as the default card type for a system's desktop, type:

```
# smartcard -c admin -a dtlogin -x add defaultcard=IButton
```

Thereafter, when you run `smartcard -c admin`, you see the following client properties:

```
dtlogin.defaultcard       = IButton
default.validcards        = CyberFlex PayFlex
```

▼ How to Define Client Application and Card Removal Timeouts (Console)

1. **Select OCF Clients from the Navigation pane.**
2. **Double-click the CDE icon.**
3. **Select the Timeouts folder.**
4. **Slide the indicator to change the amount of time for any of the following timeout values.**
   - Card Removal Timeout
   - Re-authentication Timeout
   - Card Removal Logout Wait Timeout

▼ How to Change the Client Application Behavior When a Card is Removed (Console)

1. **Select OCF Clients from the Navigation pane.**
2. **Double-click the CDE icon.**

3. **Select the Timeouts folder.**

4. **Enable or disable the following options:**

   - Ignore Card Removal
   - Re-authenticate After Card Removal

Note – The user must log out of the current session and log in again in order for these changes to take effect.

CHAPTER 6

Additional Smart Card Administration (Tasks)

This section describes additional smart card administration and maintenance tasks. It includes procedures for performing these tasks from both the Smart Card Console and from the command line.

Smart Card Administration

▼ How to Verify a PIN for a Smart Card (Command Line)

This procedure is appropriate for all cards supported by Solaris Smart Cards.

1. **Insert the smart card into the card reader.**
2. **Verify the PIN for the smart card.**

   `# smartcard -c init -A A000000062030400 -P 'PIN_number'`

   where *PIN_number* represents the PIN set for the card.

   For an invalid PIN, an `Invalid PIN` message is displayed. A valid PIN results in no output.

▼ How to Create a Private Key on a Smart Card (Command Line)

This procedure is appropriate for the Java-based iButton and Cyberflex smart cards. You cannot store a private key on the Payflex card.

To use this feature, you must have a public-key infrastructure (PKI) set up at your site.

1. **Create a public/private-key pair for the user using the appropriate commands for your PKI.**
2. **Export the private-key part of the key pair into a separate file.**
   Record the fully qualified path name of the file because you have to specify it later when setting up the private-key property.
3. **Initialize the card by typing the following on one line:**

```
# smartcard -c init -A A000000062030400 -P 'PIN_number' privatekey=key_file_name
```

PIN_number	Represents the PIN assigned to the card.
key_file_name	The full path name of the file containing the user's private key.

Note – The certificate property is not fully implemented by the `SolarisAuthApplet`.

Using a Smart Card on Multiple Systems

When you run the `smartcard -c init` command to initialize a user's smart card, you create a symmetric key on the system and on the smart card. `ocfserv` creates a file called `/etc/smartcard/.keys` that contains information about all secret keys configured on a system. If the user needs to access systems other than the system where the smart card was created, you need to export the `/etc/smartcard/.keys` file to all systems the user must access.

▼ How to Export a System's Keys File (Command Line)

To export the `/etc/smartcard/.keys` from the system where the card was created, use the following procedure:

1. **Become superuser on the system where the card was created.**
2. **Create a separate key file for the user in question. This file should contain only the user's keys as shown in** `/etc/smartcard/.keys`**.**
3. **Export the** `/etc/smartcard/.keys`**:**

```
# smartcard -c admin -k challenge_response -E -o key_file_name
```

key_file_name	The file containing the user's symmetric key, either `/etc/smartcard/.keys` or another file specifically for that user.

▼ How to Import a User's Keys File (Command Line)

Use this procedure to import the user's symmetric key onto a different system than the system where the user's card was created.

1. **Become superuser on the system to which you want to import the user's key file.**
2. **Import the key file to the new system.**

```
# smartcard -c admin -k challenge_response -I -i key_file_name
```

key_file_name	Either `/etc/smartcard/.keys` or another file that you created for the user.

3. **Repeat the first two steps on every system that the user must access through the smart card.**

Troubleshooting

See the sections below if you have trouble logging in with your smart card.

Setting Up Debugging Properties

You can debug smart card operations on a system by setting the debugging properties. Solaris Smart Cards offers standard debugging and a detailed trace of your operations, if specified.

If enabled, debugging information is logged to a file. You can control the level and amount of debugging information on 0–9 scale. Debugging is disabled by default.

▼ How to Enable Debugging (Console)

Use the Debug folder if you want to set up the `ocfserv` debugging property. Setting up debugging is optional.

1. **Select OCF Server from the Navigation pane.**
2. **Double-click the icon representing the local system.**
3. **Select the Debug folder.**
4. **Slide the indicator for the OCF Debug Level slider to the right to indicate the level of debugging you want on the OCF Server.**
5. **Slide the indicator for the Open Card Trace Level slider to the right to indicate the trace level you want on the OCF Server.**
6. **(Optional) Specify an alternate name for the debug file.**
   a. **Click Browse to view the file systems on the system.**
   b. **Type the fully qualified path name for the debug file in the OCF Debug File Location field.**
7. **Click Apply or OK.**

Enabling Debugging From the Command Line

The following debugging properties are defined for `ocfserv` by default:

```
debugging.filename        = /var/run/ocf.log
debugging                 = 0
OpenCard.trace            = com.sun:9 opencard.core:9
```

Note – If you are running a previous Solaris 8 release, the debugging log file might be called `/tmp/ocf_debugfile`.

`/var/run/ocf_log`	The name of the file to contain debugging information.
`debugging = 0`	Means that debugging is disabled. Debugging is enabled if `debugging = 1`.

`OpenCard.trace`	The OpenCard trace level.

▼ How to Enable Debugging (Command Line)

Use the following procedure to enable smart card debugging.

1. **Become superuser.**
2. **Enable smart card debugging by setting** `debugging=1`**.**

   ```
   # smartcard -c admin -x modify debugging=1
   ```

   In the following example, the location of the `ocfserv` debugging file is changed by specifying the `-x modify debugging.filename` option and a fully qualified file name for the debugging file.

   ```
   # smartcard -c admin -x modify debugging.filename=/var/tmp/sc.debug
   ```

▼ How to Disable Smart Card Operations (Command Line)

You might need to disable smart card operations on a system if a smart card configuration error does not allow a user to log in with a smart card, or if a system no longer needs a smart card login.

1. **Become superuser.**
2. **Disable smart card operations.**

   ```
   # smartcard -c disable
   ```

▼ How to Resolve Smart Card Configuration Problems

The `/etc/smartcard/opencard.properties` file stores important smart card configuration information. This file requires no administration and should not be edited manually. However, if you inadvertently introduced a problem in your smart card configuration by using either the Smart Card Console or the command line, you can restore the previous version of the `/etc/smartcard/opencard.properties` file from the command line.

1. **Become superuser.**

2. **Change to the** `/etc/smartcard` **directory.**

3. **Save the current version first.**

```
#  cp opencard.properties opencard.properties.bad
```

4. **Copy the previous version to the current version.**

```
# cp opencard.properties.bak opencard.properties
```

▼ How to Resolve Applet Downloading Problems

1. **If you see the following message while trying to download the applet on the card, it is possible that you have not added the ATR of the smart card inserted in the reader to the list of valid ATRs the system can accept.**

```
SmartcardInvalidCardException
```

2. **Try updating the card's ATR by following the procedure in "To Add Support for a New Card Type (New ATR)" on page 18.**

▼ How to Resolve Missing Smart Card ATR Problems

When you try to add the smart card by using the Smart Card Console, a screen displays the ATR of the card inserted in the reader. If the ATR displayed does not exist in the list of valid ATRs, add the ATR to the *card-name*`.ATR` property.

See "To Add Support for a New Card Type (New ATR)" on page 18 for more information.

Example—Adding a Missing ATR of a Smart Card (Command Line)

Display `ocfserv` properties to see if the *card_name*.ATR property exists.

```
# smartcard -c admin
```

For example, `ocfserv` lists a property `MySCM.0.ATR`, where `MySCM` is the user-friendly name of the card reader. This property reflects the ATR of the smart card inserted in the reader. This property is temporary and is added by `ocfserv` only for the time the card is in the reader. This property is removed when the card is removed.

Add this ATR to the *card_name*.ATR property if the ATR displayed by this property does not exist in the list of valid ATRs.

▼ How to Resolve Smart Card Login Problems

After you have enabled smart card operations and logged of the system, the CDE login screen displays the following prompt:

```
Please insert Smart Card
```

1. **If you are unable to log into your system using a smart card because of smart card setup problems, try logging in remotely with the `rlogin` or `telnet` commands.**
2. **Become superuser, then attempt to disable smart card operations, rather than try to re-install the system first.**

   After smart card operation is disabled, the CDE screen displays the following prompt:

   ```
   Enter User Name
   ```

Glossary

Answer to Reset (ATR)	A property assigned to each smart card type by the manufacturer that identifies the version of the smart card. An equivalent property is stored on the system to assist in authentication.
ATR	See Answer to Reset.
authentication	The process of verifying a user's identity.
CDE	See Common Desktop Environment (CDE).
challenge-response	A form of authentication whereby the smart card is loaded with a DES key used in response to a random number generated by the system and sent to the card when the card is inserted in the card reader.
Common Desktop Environment (CDE)	A desktop application used in the Solaris operating environment.
Console pane	The pane in the Smart Card Console that contains icons for various management tasks.
Information pane	The pane in the Smart Cards Console that contains a brief description of the category or icon just clicked, as well as instructions for beginning the task associated with that category or icon.
Navigation pane	The pane in the Smart Card Console that lists major categories of tasks involved in setting up smart cards.
personal identification number (PIN)	A unique number used to identify a user.
PIN	See personal identification number (PIN).
private key	A type of security that works in a public-key infrastructure, involving pairs of key strings. The private key part of this pair is stored on the smart card.
Solaris Smart Cards	Name of the software that enables the use of smart cards in a Solaris operating environment.

smart card	A plastic card that has been initialized in such a way as to allow the user to access a system by inserting the card into a card reader.
Smart Card Console	The GUI tool that enables an administrator to manage Solaris Smart Cards.
symmetric key	Another term for the DES key described in challenge-response authentication method.

Index

R

S

U

V