AUTHORIZED DOCUMENTATION

Administration Guide

Novell® International Cryptographic Infrastructure (NICI)

2.7x

July 24, 2008

www.novell.com

Legal Notices

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see the Novell Documentation Web page (http://www.novell.com/documentation).

Novell Trademarks

Third-Party Materials

Contents

About This Guide

This guide describes the structure and functionality of Novell® International Cryptographic Infrastructure (NICI), how to set it up, and how to manage it. This guide also documents NICI error messages.

Audience

This guide is written primarily for network administrators.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comment feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For the most recent version of the *NICI 2.7x Administration Guide*, see the NICI Administration Guide Web site (http://www.novell.com/documentation/lg/nici27x/index.html).

Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

Introduction

1

Novell® International Cryptography Infrastructure (NICI) is the Novell solution to a cross-platform, policy-driven, independently certified, and extensible cryptography service. NICI is the cryptography module that provides keys, algorithms, various key storage and usage mechanisms, and a large-scale key management system.

NICI controls the introduction of algorithms and the generation and use of keys. NICI allows a single commodity version of security products to be produced for worldwide consumption that supports strong cryptography and multiple cryptographic technologies. Initial services built on this infrastructure are Directory Services (Novell eDirectory™), Novell Modular Authentication Service (NMAS™), Novell Certificate Server™, Novell SecretStore®, and TLS/SSL.

NICI first shipped with NetWare® 5.0. This document is provided to help resolve NICI issues found in the field or during testing of various Novell or third-party products. A particular product might use NICI directly or indirectly via another module (NLM™, DLL, so, etc.).

WARNING: All actions described here can cause unrecoverable data loss and must be executed with the full knowledge of such an action. Most NICI problems, as well as solutions, have implications in other products. It might not be easy to predict the effects of taking a NICI action. NICI is one of the most critical services in the system and if it is inoperable, it typically renders the system inoperable, as well as causing permanent and unrecoverable damage. If certain NICI keys are irrecoverably lost, even backed-up data might be useless, because it can't be decrypted.

The contents of this document do not guarantee a fix. All information is advisory.

Table 1-1 *NICI Configuration Directory*

Platform	Startup Directory	NICI Directory	NICI User Directory
NetWare	`c:\nwserver`	`sys:\system\nici`	`sys:\system\nici`
Microsoft* Windows*	*SystemRoot*`\System32`	*SystemRoot*`\System32\Novell\NICI` (See Chapter 3, "NICI Setup," on page 15)	*SystemRoot*`\System32\Novell\NICI` (See Chapter 3, "NICI Setup," on page 15)
UNIX*	`/opt/novell/lib`	`/var/opt/novell/nici` (See Chapter 3, "NICI Setup," on page 15)	`/var/opt/novell/nici/` (See Chapter 3, "NICI Setup," on page 15)

NICI on UNIX platforms is LSB-compliant starting with version 2.7.0.

NICI Modules

2

NICI supports multiple platforms. It is a shared library (DLL, so, etc.) except on the NetWare®, where it is comprised of multiple signed NLM™ programs called XLMs. On platforms other than NetWare, NICI has another module running in the DHost environment in server mode distributed as part of a Novell® eDirectory™ release.

This release of NICI supports the following server platforms:

- NetWare
    - NetWare 6.5 SP2 or later
- Windows
    - 2000 Server SP4 or later
    - 2000 Adv Server SP4 or later
    - 2003 Server or later
- Linux*
    - SUSE® Enterprise Server (SLES) 8.x
    - SLES 9
    - SLES 9 SP1
    - Red Hat* Linux Advanced Server AS 3.0
- Solaris*
    - Solaris 9 on Sun SPARC
    - Solaris 10 on Sun SPARC
- AIX*
    - AIX 5L Version 5.2 with all recommended AIX patches

The following modules are used in NICI:

- Section 2.1, “NetWare,” on page 11
- Section 2.2, “Windows,” on page 12
- Section 2.3, “UNIX,” on page 13

2.1 NetWare

NICI on NetWare has multiple signed NLM programs called XLMs. The `MODULES` command displays the NLM names, not the XLMs. The startup directory is typically `c:\nwserver`.

- `ccs.xlm` (`ccs.nlm`)

  This file is located in the startup directory, and is the only XLIB module that exports APIs used by other NLM programs.
- `xmgr.xlm` (`xmgr.nlm`)

  This module is located in the startup directory. It has no usable APIs by other NLM programs.

- `expxeng.xlm` (`xengnul.nlm`, `xengexp.nlm`, `xngaexp.nlm`)

  This module is located in the startup directory. The presence of these NLM programs identifies the availability of weak/exportable cryptography.

- `domxeng.xlm` (`xengnul.nlm`, `xengexp.nlm`, `xengusc.nlm`, `xngausc.nlm`)

  This module is located in the startup directory. The presence of these NLM programs identifies the availability of strong/domestic cryptography. As of NICI 2.*x*, Novell ships strong cryptography worldwide.

- `xsup.xlm` (`xsup.nlm`)

  This module is located in the startup directory.

- `nicisdi.xlm` (`nicisdi.nlm`)

  This module is present in the `sys:\system` directory and is loaded by the `autoexec.ncf` file. This is the Security Domain Infrastructure management module. If this module is not loaded, then security domain keys (such as the tree key) are not loaded into NICI and they are not available. You usually see a 1460 error when this module is not loaded.

- `sasdfm.xlm` (`sasdfm.nlm`)

  This module is present in the `sys:\system` directory and is loaded by the `autoexec.ncf` file. This is the SAS Data Flow Manager file; it is responsible for handling NCP™ communications for session key setup, as well as handling client NICI initialization requests. The absence of this module disables session key support in NICI. Typical symptoms are not being able to export user certificates in ConsoleOne®, or not being able to use NMAS™ to log in to eDirectory.

2.2 Windows

- `niciccs.sys` and `niciccs.vxd`

  NICI versions before NICI 2.0 are kernel drivers. On Windows NT*/2000 systems, it was called `niciccs.sys` and was located in the `drivers` directory under the `system32` directory. On Windows 95/98 systems, it was called `niciccs.vxd`. Kernel versions of NICI are no longer maintained.

- `ccsw32.dll`

  NICI versions newer than 2.*x* have a DLL named `ccsw32.dll`. These are the FIPS 140 level 1 and level 2 certified modules. Refer to the security policy document for more on the FIPS 140 evaluations. Simply copying the DLL into a directory does not make NICI operational, because it requires Windows registry and configuration file setup. Additionally, a NICI module self-verifies, so most components are coupled with the distributed DLL, and usually are not distributable alone. NICI does not depend on directory services to be installed.

- `niciext.dlm`

  In the DHost environment, NICI has a DLM called niciext.dlm, which manages NCP connections and other Novell eDirectory services on behalf of NICI. The DLM is shipped with eDirectory distributions.

2.3 UNIX

- `libccs2.so`

  The first version supported on all UNIX platforms is 2.3.0. NICI is a shared object (`.so`) named `libccs2.so`. Typically, it is a symbolic link to the actual file named per platform and version. NICI does not depend on directory services to be installed.

- `libniciext.so`

  In the DHost environment, NICI has a shared object called `libniciext.so` loaded by DHost to carry out communications and other directory services on behalf of NICI. The shared object is shipped with eDirectory distributions.

NICI Setup

3

We strongly encourage using the NICI install program provided on each platform to install and configure NICI. NICI installed by other means can cause irreparable damage. It might be necessary to remove NICI, and perhaps remove other items such as certificates that a customer has purchased, and reinstall NICI properly.

3.1 NICI Configuration Files

NICI configuration files are located in the NICI directories listed on Table 1-1 on page 9. The NICI configuration files listed in the following table are present on all platforms. Platform-specific files and other configuration details are explained in following sections.

Table 3-1 *NICI Configuration Files*

File	Created by	Description
`NICIFK`	Novell® eDirectory™ install	NICI license material for server-mode operation.
`xmgrcfg.wks`	NICI Install	NICI license material for client-mode operation. Not used if NICIFK is present.
`xmgrcfg.nif`	First use of NICI or by install by a privileged user	NICI per-box unique keying material generated locally.
`xarchive.000`	First use of NICI by a privileged user	NICI master archive.

The NICI configuration files are signed and partially encrypted. An invalid license file (`NICIFK`) or a client license file (`xmgrcfg.wks`) renders NICI nonfunctional.

There are two other configuration files that might be present, which are used to switch NICI into server mode when programs such as eDirectory are installed. The files are:

- `nicifk.new`
- `set_server_mode` (Linux/UNIX)

  or

  `set_server_mode.bat` (Windows)

3.2 NICI User Configuration Files

NICI creates NICI user directory when a user first uses NICI, if the directory does not already exist. NetWare® does not have user directories, because the system has only one user: the server itself. Likewise, user directories are not created on single-user systems like Windows 95/98/Me, if multi-user capability is not configured. NICI sets the rights on each user directory when it creates the directory, so that only the user has access to it.

The system administrator (such as the Administrator on Windows or `root` on UNIX) must typically take the ownership of a user directory, and then change its permissions accordingly. Refer to the operating system's file management utilities for more details.

Table 3-2 *NICI User Configuration Files*

File	Created by	Description
`xmgrcfg.ks2`	First use	User-specific key materials and other configuration materials.
`xmgrcfg.ks3`	First use or update	User-specific state data, updated occasionally.
`xarchive.001`	First use or update	NICI user archive.

3.3 NetWare Configuration

The `sys:/system/nici/nicisdi.cfg` file is used to configure the NICISDI module's operation parameters. By default, this file does not exist. When NICI is in server mode, it creates the file on-demand. At present, the only configurable parameter is the synchronization period the `nicisdi.xlm` module checks for new security domain keys. A typical file contains the following:

```
# This is a sample NICISDI.CFG file for NetWare systems.
# There is only one configuration parameter; all others are ignored.
# The pound sign in the first column marks the
# entire line as a comment, and the line is ignored.

# The time in minutes NICISDI.XLM module polls.
NICISDI Sync Period = 60
```

The `nicisdi.cfg` file is read when the `nicisdi.xlm` module is loaded (as part of `autoexec.ncf` processing). If the file does not exist, does not contain the sync period, or if the sync period is zero, NICISDI does not attempt to read it again. If the file exists and contains a non-zero sync period, the file is read once in a period before synchronization. You can disable the background synchronization process by deleting the file, setting the period to zero, or commenting out the sync period line.

The `sys:/system/nici/nicisdi.key` file contains encrypted security domain keys as discussed in Chapter 4, "NICISDI: Security Domain Infrastructure," on page 21.

3.4 Windows Configuration

The NICI install creates and populates a key in the Windows registry. The location of the key is `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI`. The table below describes each value.

Table 3-3 *Windows Key Values*

Key	Type	Description
`ConfigDirectory`	String	Location of NICI configuration files
`DAC`	Binary	NICI module's digital authentication code.
`SharedLibrary`	String	The name of the library, such as `ccsw32.dll`
`Strength`	String	U0 for strong, W1 for import restricted (no longer supported)
`UserDirectoryRoot`	String	(Optional). Name of a directory where user directories are created. Defaults to ConfigDirectory.
`Version`	DWORD	NICI version, such as 0x00002400 for 2.4.
`NICISDI Sync Period`	DWORD	NICISDI synchronization period in minutes, represented in hexadecimal.
`EnableUserProfileDirectory`	DWORD	NICI user files are created in the `Application Data\Novell\NICI` directory in the user's profile directory.

Users' directories are created, by default, in the `<systemroot>\system32\novell\nici` directory by the user's name, for example, `c:\winnt\system32\novell\nici\administrator`. To change the root directory in which all user directories are created, edit the string type registry entry `UserDirectoryRoot` in the NICI registry key, and set it to the desired root directory. For example, use `c:\documents and settings` to create the NICI user configuration files in each user's local profile path on a Windows 2000 system.

The username is the name of the user owning the process that started NICI. If it is a local user, NICI uses the username. If it is a remote or a domain user, NICI forms the username as the combination of username and domain separated by a dot (userName.domainName).

`EnableUserProfileDirectory` is not created by the NICI install, so it is disabled. If it is set, existing NICI user files might need to be copied or moved to the new location. If the user profile directory is enabled, NICI does not set the ACLs on this directory. It relies on existing security properties (ACLs, inheritance, and ownership) of the user's profile directory. Use this option very carefully, because you can disclose all users' NICI keys. NICI creates the `Application\Novell\NICI` directory if it is not present, and stores all NICI user files in this directory. This option is provided to enable the dynamic user creation/deletion feature in the Novell ZENWorks® product. It must be set manually or by another application's install, such as ZENWorks.

The `niciext.dlm` module reads the nicisdi sync period value when DHost loads it. If the value does not exist, or if the period is zero, NICIEXT does not attempt to read it again. If the value exists and contains a non-zero period, the value is read once in a period before synchronization. You can disable the background synchronization process by deleting the value, or setting the period to zero.

The `<systemroot>\system32\novell\nici\nicisdi.key` file contains encrypted security domain keys as discussed in Chapter 4, "NICISDI: Security Domain Infrastructure," on page 21.

All users have read, execute, and create rights to the files in the NICI configuration directory (`<SystemRoot>\Novell\NICI`). NICI dynamically creates user directories upon first use of NICI by that user, and give full rights only to the user creating the directory.

3.5 UNIX Configuration

The `/etc/opt/novell/nici.cfg` file emulates the Windows registry in an editable text file. Most of the entries are set up by NICI install. A typical `/etc/opt/novell/nici.cfg` file is shown below.

```
ConfigDirectory:s:16:/var/novell/nici
SharedLibrary:s:19:/usr/lib/libccs2.so
DAC:b:8:1a:aa:6d:49:48:a8:83:98
MkUserDir:s:24:/var/novell/nici/nicimud
NiciVersion:s:5:2.4.0
BuildVersion:s:11:4001101.23
BuildDate:s:6:020123
NiciStrength:s:2:u0
NICISDI Sync Period:b:1:3c
```

Each line can have multiple entries all separated by a colon (:). The first entry in a line is the name, followed by its type. The second is the length in decimal, followed by the actual value. There are two types, string (s) and binary (b). For example, the name of the first line in the sample above is `ConfigDirectory`, of type string (s) of 16 characters. The value is `/var/opt/novell/nici`. The name of the last line is `NICISDI Sync Period`, of type binary (b) of 1 hexadecimal digit; its value is 0x3c, or 60 in decimal, which represents minutes for this particular parameter.

Each line is described in the table below, or in Table 3-3 on page 17.

Table 3-4 *Unix Key Values*

Key	**Description**
`MkUserDir`	This executable executed to create user directories. `/var/novell/nici/nicimud` is supplied by NICI install.
`NICIVersion`	NICI version string.
`BuildVersion`	NICI build version string.
`BuildDate`	NICI module's build date; year, month, and day, each in two decimal digits.
`NiciStrength`	u0 for strong, w1 for import restricted (no longer supported).
`NICISDI Sync Period`	NICISDI synchronization period in minutes, represented in hexadecimal.

The `libniciext.so` module reads the NICISDI sync period value when ndsd loads it. If the value does not exist, or if the period is zero, NICIEXT does not attempt to read it again. If the value exists and contains a non-zero period, the value is read once in a period before synchronization. You can disable the background synchronization process by deleting the value, or setting the period to zero.

The `/var/opt/novell/nici/`*`uid`*`/nicisdi.key` file contains the encrypted security domain keys as discussed in Chapter 4, “NICISDI: Security Domain Infrastructure,” on page 21. The UID is the numeric user ID defined by the UNIX system. For example, it is typically 0 for root. Having a `nicisdi.key` file for each user enables multiple instances of eDirectory running with different user IDs to host multiple trees on the same physical box.

All users have read and execute (where applicable) rights to the files in the NICI configuration directory (`/var/opt/novell/nici`). Only the installing user has full rights in the configuration directory. User directories are created by a setuid executable (nicimud, meaning the NICI Make User directory) provided by NICI install by user IDs. The nicimud creates a user directory upon the first use of NICI by that user, and gives full rights only to the user creating the directory (0700).

NICISDI: Security Domain Infrastructure

4

NICISDI stands for NICI Security Domain Infrastructure. This module is responsible for managing domain keys, where a domain is typically defined as the whole tree. In the future, a directory partition or custom domains will be able to be defined.

Up to NICI version 1.5.*x*, NICI supports one single partition key, the partition being the whole tree. Starting with NICI version 2.0.1, NICI can manage multiple partition keys of varying strengths and algorithms. Such keys are called Security Domain keys. On NetWare®, Windows, and `libniciext.so` on UNIX platforms, the module manages security domain keys in coordination with NICI. Various other services rely on the availability on security domain keys, including but not limited to SecretStore/Single-Sign-On, PKI (Certificate Server), and NMAS™.

The NICISDI module is different from the SASDFM module. SASDFM manages session keys between two boxes, typically between a client and a server. The modules are both loaded during `autoexec.ncf` processing on NetWare. Multiple loading of these modules is controlled and should not cause problems if NICI 1.5.5 or newer is installed on the system.

Security domain servers manage security domain keys. Any server can be configured as a security domain server. There can be multiple security domain servers in a tree. Security domain keys are not intended for clients.

One tree key is installed by an eDirectory installation. The tree key is created or retrieved from the security domain key server during the server installation.

- Section 4.1, “Tree Merging and Splitting,” on page 21
- Section 4.2, “Directory Objects,” on page 22
- Section 4.3, “Key Synchronization,” on page 22
- Section 4.4, “Initsdi.nlm,” on page 22
- Section 4.5, “SDIDiag,” on page 23

4.1 Tree Merging and Splitting

Merging two or more trees with NICI versions before NICI 2.0.1 caused problems in various components including PKI, NMAS and Novell® SecretStore®. With NICI 2.0.1, multiple security domain key support and automatic key synchronization is added, reducing such problems short of rebooting a server and adding a server name to a directory attribute. See Section 4.2, “Directory Objects,” on page 22 for more details.

Tree splits do not cause major problems like tree merges do. Nevertheless, it is strongly recommended that existing security domain keys are revoked, and new ones created after a tree split, so old security domain keys cannot access encrypted data protected by such keys. However, new data must be encrypted with one of the new security domain keys to facilitate cryptographic tree separation. A tool is being developed for administration of security domain keys.

4.2 Directory Objects

In the directory, the Security.KAP.W0 container off the root has a list of attributes to aid in security domain key management. These attributes are described below:

- Section 4.2.1, "NDSPKI:SD Key Server DN," on page 22
- Section 4.2.2, "NDSPKI:SD Key List," on page 22

4.2.1 NDSPKI:SD Key Server DN

This multivalued attribute contains the list of SD key servers in the tree. There must be at least one server in this list. NICI 2.0.1 and newer versions, which are distributed with NetWare 6 or later, make use of this attribute. NICISDI or NICIEXT reads this attribute on each loading (typically server boot). Then NICISDI or NICIEXT connects to each server in this list, and requests any new security domain keys from each server in this list. Existing security keys are also checked for revocation. However, deletion of a security domain key is not automatically done. Only new key retrieval (not creation) and key revocation are automatically done on every loading of NICISDI or NICIEXT, or periodically as configured by the NICISDI sync period.

For a tree merge, add the name of the new SD key server's name to this list after trees are merged, and reboot all the servers in the tree unless periodic synchronization is enabled. The final list must contain the names of SD key servers in all trees. We strongly recommend that NICI version 2.0.1 or newer be installed on servers.

4.2.2 NDSPKI:SD Key List

This attribute is reserved for future use to hold the list of security domain key identifiers.

4.3 Key Synchronization

NICISDI or NICIEXT can be configured to periodically synchronize its keys with each SD key server. This feature is disabled by default. See Chapter 3, "NICI Setup," on page 15 for setup information.

The sync period value can be updated while the server is up, and the server does not need to be rebooted for the change to take effect. The new period value takes effect in the next scheduled synchronization time. Setting this value to zero or removing it entirely causes the termination of the background thread at the next scheduled execution. Thus, further changes of this value to a nonzero value have no effect unless the server reboots.

Starting with NICI 2.4.0, NICI creates a domain key automatically on a server with WRITE rights to the domain's object in the Security.KAP container. It is designed to support multiple domains created in the Security.KAP container. At present, there is only one domain represented by W0 in the Security.KAP container.

4.4 Initsdi.nlm

This obsolete NLM™ was provided to create or to retrieve a tree key (the only security domain key at the time) during installation. This NLM can be used to create and retrieve a tree key as a standalone utility.

In order to create or retrieve a tree key, the security domain key file, `nicisdi.key`, must be deleted. The `nicisdi.key` file, regardless of the platform/OS, is server-unique, and should not be copied from one machine to another. Copying it would not make the key available. Manual creation of a new key typically causes more problems by introducing a new key on the server. It is run differently from the actual tree key other servers have. We strongly recommend that you do not use this NLM, and let NICI 2.4 or later manage such keys.

If you need to create a new tree key on the local box, run

```
INITSDI -new logFile errorFile serverName
```

To retrieve the tree key from a server in the same tree, run

```
INITSDI -get logFile errorFile serverName treeName
```

For instance, to receive a key from server server.novell in the novell tree, load

```
INITSDI -get sys:\sdi.log sys:\sdi.err server.novell tree
```

The `initsdi.nlm` is obsolete with NICI 2.4, because it provides auto-sync and auto-create capabilities. The file might not work if the target server has NICI versions 2.0.1 or later.

4.5 SDIDiag

SDIDiag is the Security Domain Infrastructure diagnostic and repair utility. Among other things, SDIDiag allows an adminstrator to:

- Run CHECK to verify that all Security Domain servers have a consistent key set
- View the various keys within an eDirectory container or Tree
- Ensure that all servers are synchronized with consistent keys.

For information on using this utility, see TID #319224010081773 (http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3192240&sliceId=SAL_Public&dialogID=2494558&stateId=1%200%202492907).

IMPORTANT: If you have installed eDirectory to use a non-standard port, you must specify the port number with the IP address when you run SDIDiag. For example, xxx.xxx.xxx.xxx:port.

Installing and Upgrading

5

NICI has a platform-dependent installation program. Reinstalling NICI does not destroy existing keys. Except on NetWare®, the NICI 2.0 or later install does not require rebooting the server in most instances. However, if the NICI module (DLL or .so) is in use and can't be overwritten by the install program, a reboot might be necessary.

- Section 5.1, "Version Upgrades and Compatibility," on page 25
- Section 5.2, "NICI Transfer (NUWNICI)," on page 25
- Section 5.3, "NICI Backup and Restore," on page 26
- Section 5.4, "Upgrading NICI to Version 2.x on UNIX Systems," on page 26
- Section 5.5, "Using Sudo to Allow Non-Root Users to Install NICI on UNIX Servers," on page 26

5.1 Version Upgrades and Compatibility

Installing a newer version of NICI on top of an existing NICI installation upgrades NICI. Always upgrade NICI using its install program. Freely copying NICI modules often results in a chaotic system, and consequences of such an action often cause irreparable damage to the system and other products such as PKI, Novell® SecretStore®/Single Sign-On, NMAS™, directory services, and others.

Applications developed for NICI 1.*x* are not compatible with newer NICI versions (2.*x* or later). To provide backward compatibility, NICI 1.*x* on Windows platforms can coexist with newer NICI versions. If you want to keep this coexistence, always install the newer version after the old version of NICI. For example, install NICI 2.4 after NICI 1.5.7.

5.2 NICI Transfer (NUWNICI)

As part of the NetWare upgrade wizard utility, the NICI team provided an NLM™ (`nuwnici.nlm`) to encrypt and transfer NICI configuration files from one physical server to another. The encrypted files are written to a floppy diskette, and the floppy is physically transported to the target server. `nuwnici.nlm` can also be used as a standalone NICI transfer utility. It has multiple phases. The first phase (Phase 1) is executed on the target (new) server. Phase 2 is executed on the source (old) server. Phase 3 is executed on the target (new) server. After phase 3 is completed, the target (new) server must be rebooted for the transfer to take effect.

There is also a phase 4 executed on the target (new) server. Phase 4 is basically a tool to check if NICI is working properly on the new server after the reboot.

A help screen is displayed by using the -h command line option.

On platforms other than NetWare, copying the NICI configuration files to the new box transfers NICI keys and keying materials to the new server.

If you do this, we highly recommend that you delete the NICI configuration on the old server.

5.3 NICI Backup and Restore

For information on backing up and restoring NICI, see Chapter 6, “Backing Up and Restoring NICI,” on page 27.

5.4 Upgrading NICI to Version 2.*x* on UNIX Systems

A hybrid version of NICI (mixing features of NICI 1.2 and NICI 1.5) was shipped with the Tao version of eDirectory™ (eDirectory 8.5). In order to migrate the NICI configuration files from the hybrid version to 2.*x*, an upgrade utility (runf2dc) is provided.

If a UNIX (Solaris, Linux, or AIX) server is hosting more than one eDirectory, each eDirectory instance typically has its own NICI directory setup. Both instances of NICI configuration files must be migrated with the provided tool. For instance, assume two eDirectory instances run on a single Solaris host in the `/var/nds1` and `/var/nds2` directories, respectively. The runf2dc tool must be run on both the `/var/nds1/nici` and `/var/nds2/nici` directories to migrate each instance separately.

Materials in the NICI configuration files don’t depend on the contents of eDirectory files. Instead, encrypted data in eDirectory depends on keys stored in NICI configuration files. This encrypted data (such as user private keys, certificates, secret store data, and NMAS store data) is not available if NICI files are not migrated properly.

We strongly recommend running each instance of eDirectory on the same host with different user IDs to separate their cryptographic materials using the host system’s security mechanisms. NICI does not require a special user to run, except for installation, when a privileged user who can install setuid programs must install NICI (a one-time operation).

IMPORTANT: When upgrading to NICI v2.7.0 or later on all UNIX platforms, you must first remove the existing NICI installation.

5.5 Using Sudo to Allow Non-Root Users to Install NICI on UNIX Servers

To install NICI on UNIX servers, the person doing the installation must either be logged in as `root` or have `root` access. The Sudo utility provides an easy way for a system administrator to delegate `root` access to another user to perform specific tasks.

As with any delegation of rights, the system administrator should take the necessary precautions to ensure that the company's standards of security are followed.

Most UNIX providers have the Sudo utility available on their Web sites. You can also download the Sudo utility from the Sudo Web site (http://www.sudo.ws). The Sudo documentation provides information on how to install and configure the utility. When it is configured properly, the user can install NICI by entering `sudo` *`install_command`*.

Backing Up and Restoring NICI

6

NICI stores keys and user data in the file system and in system-specific and user-specific directories and files. The NICI installation program protects these directories and files by setting the proper permissions on them, using the mechanism provided by the operating system.

Uninstalling NICI from the system does not remove these directories and files; therefore, the only reason to restore these files to a previous state is to recover from a catastrophic system failure or a human error. Also, overwriting an existing set of NICI user directories and files might break an existing application.

Backing up and restoring NICI requires two things:

1. Backing up and restoring directories and files
2. Backing up and restoring specific user rights on those directories and files

The exact sequence of events required is platform-dependent.

When you back up and restore NICI, it is critical that you maintain the exact permissions on the directories and files. NICI's operation and the security it provides depends on these permissions being set properly.

Typical commercial backup software should preserve permissions on the NICI system and user directories and files. You should check your commercial backup software to see if it does the job before you do a custom backup of NICI.

You should always back up the existing NICI directory structure and its contents, if any, before doing a restore. If you lose the machine key, it is unrecoverable. Because the user data and keys could be encrypted by using the machine key, losing it results in a permanent loss of user data.

To do a restore of NICI only, you must understand which specific files must be restored. During restoration, it is important that the correct access rights be restored for the correct owner. On UNIX and Windows systems, the name of the user-specific directory reflects the ID of the owner, but on both systems the owner ID might change between the time of the backup and the time of the restore. It is important for security reasons that you know which account is being restored and that you assign the directory name and access rights accordingly. The mere existence of a user account on the system with the same ID as what was backed up does not mean that the current account is the actual owner of the information being restored.

- Section 6.1, "Linux/UNIX Systems," on page 27
- Section 6.2, "NetWare," on page 30
- Section 6.3, "Windows," on page 30

6.1 Linux/UNIX Systems

In NICI versions earlier than 2.7.0, the `/var/novell/nici` directory contains all the system and user directories and files.

- Section 6.1.1, "Performing a Backup," on page 28
- Section 6.1.2, "Restoring NICI," on page 29

6.1.1 Performing a Backup

The NICI configuration files are located in the `var/opt/novell/nici` directory. The configuration files are associated with each user account on the operating system. In order to back up a user's configuration files, you must preserve the contents of the novell configuration directory located in the following tables and the user-specific subdirectory within it (alternatively, back up everything within the directory). You might find some executables in the directory. They do not need to be backed up.

Applications that use NICI to perform cryptography might have dependencies on data that NICI manages. If so, it might be necessary to back up the NICI configurations files in order to recover the encrypted data, or just to preserve the state of the files as part of an incremental backup. This section assumes that you have other means to perform disaster recovery or rebuild a system and just need to know which files must be backed up and restored in order to preserve critical NICI data that is not recoverable by simply reinstalling NICI. You should consult the individual application documentation to determine if NICI data is critical to the application. If it is, the NICI files should be backed up at the time the application data is backed up.

The critical NICI configuration files are listed in Table 3-1 on page 15. Some of those files are unique to a specific user. The configuration files are all contained within one directory, identified in the last row of Table 6-1, which contains common files. Files unique to specific users are contained within subdirectories of that directory. For simplicity, you can back up the entire directory structure or back up the common files and specific user files, whichever is most convenient. Be sure that you can restore the access rights on the directories and files later. When you restore the files you can make decisions about exactly which files must be recovered. Be sure to note which version of NICI is installed at the time, because the configuration files may not be compatible with earlier versions.

The following directories and files should be backed up. Remember to preserve the rights on all the directories and files.

Table 6-1 *For NICI Versions Earlier than 2.7.0*

Directory/File Name	File Type and Special Instructions
`/etc/nici.cfg`	Configuration file.
`/usr/lib/libccs2.so`	Symbolic link to the actual library in `/usr/lib/`.
`/usr/lib/libccs2.so.*`	NICI library. The version of the library completes the name.
`/var/novell/nici`	Contains all the system keys, user directories, and files, and the programs used to initialize NICI.

Table 6-2 *For NICI Versions 2.7.0 and Later*

Directory/File Name	File Type and Special Instructions
`/etc/opt/novell/nici.cfg`	Configuration file.
`/opt/novell/lib/libccs2.so.*`	NICI library. The version of the library completes the name.

Directory/File Name	File Type and Special Instructions
`/var/opt/novell/nici`	Contains all the system keys, user directories, and files, and the programs used to initialize NICI.

NOTE: Depending on your operating system and the version of NICI installed, there might be additional files, particularly executable files, within the directories. Those additional files, which are created during NICI installation, do not need to be backed up. See Table 3-1 on page 15 for a list of the configuration files.

6.1.2 Restoring NICI

At some point it might be necessary to recover NICI configuration files so that the information they contain can be used to decrypt data for an application or simply to restore NICI to a previous state. We assume that you backed up the NICI configuration files at the same time you backed up the application.

WARNING: Overwriting existing NICI configuration files can cause critical data to be lost. If an application has used NICI to encrypt data and the NICI configuration files are lost, it might not be possible to recover the encrypted data. Always keep copies of any files you overwrite. Different applications might have conflicting needs and you might need to recover the data for one application, then restore the system again to recover the data for a second application or continue with normal operations.

1. Reinstall NICI to a known good state.
2. Determine which user files must be restored.

   It might be necessary to recover files from one user directory and place them in a different user directory if the users on the system have changed. For example, if Bob originally encrypted data, then the data should not accidentally be revealed to Mary.
3. Recover the common configuration files and the appropriate user-specific files.

   This may invalidate the configuration files for other users not recovered from the same backup. It might be appropriate to just delete all the configuration files before attempting to restore any specific user files. Re-establish the correct access rights so that each user has approved access to the correct configuration files.

The administrator should follow the above steps. But a knowledgeable operator might choose to restore individual files or directories, possibly changing the names of the files or directories and assigning new access rights.

This can be done if the `nicifk` and `xmgrcfg.wks` files haven't changed from those on the backup store.

The following guidelines for each file/directory are recommended when restoring if NICI is already installed on the server:

Filename	Guidelines
`xarchive.000`	Can be restored over an existing file.

Filename	Guidelines
`xmgrcfg.nif`	Can be restored over an existing file.
User-specific directories and files	Make sure that the user ID in the backup is the same as the user on the box. If the user directory already exists, then it must be determined if the user wants to keep the current files or restore them to a previous state. Normally, user configuration files should be restored as a group rather than individually. Be sure to restore the user files under the correct user's user ID and to restore the rights on the user directory and contents. For example, if BOB had user ID 1000 at the time of the backup but now has user ID 5000, then the files in the backed up directory 1000 should be restored to directory 5000, or BOB's UID must be changed back to 1000. So, the restore process must not just blindly restore the user directories without input from the operator. In either case, a backup of the existing NICI user directory needs to be done.

6.2 NetWare

For versions earlier than NICI 2.*x*, the configuration files were kept in `sys:\_NetWare` and different procedures apply. These instructions are valid only for NICI versions 2.*x* or later.

6.2.1 Performing a Backup

Back up the `sys:\system\NICI` directory and any subdirectories along with access rights. There is only one user on NetWare, so the problem of backing up and restoring the user directories does not exist.

6.2.2 Restoring NICI

1 Determine if NICI is already installed on the server by searching for the `sys:\system\nici\nici.cfg` file, then do one of the following:

- If NICI is not installed on the server, just restore the `sys:\system\NICI` directory and its contents.
- If NICI is installed on the server, make a backup of the existing setup and remove NICI from the server. Then copy the whole backup structure from the backup store.

Selective restoration can be done only if the `nicifk` file hasn't changed from the one on the backup store. If it hasn't changed, you can restore whatever files in the `sys:\system\NICI` directory you choose. Generally, the files should be restored as a group, but if you are knowledgeable, you might choose to restore only certain files or subdirectories.

6.3 Windows

Configuration information is kept in the system registry under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI.
```

A second key identifies the version of NICI currently installed. For example:

`HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI (Shared) U.S./Worldwide (128 bit).`

6.3.1 Performing a Backup

1. Back up any registry information under

   `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI*.`

   NICI* indicates all registry keys that begin with NICI. There might be more than one.
2. Back up the directory, including subdirectories, identified by

   `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI\ConfigDirectory`

   As with UNIX systems, you should remember the access rights on that directory and all subdirectories.

On Windows systems, if commercial software is used to do the backup, make sure the backup program itself runs as a system process. This ensures that the program can access all the directories and subdirectories.

6.3.2 Restoring NICI

1. Determine if NICI is already installed on the server by searching the registry for the NICI registry keys mentioned in Section 6.3.1, "Performing a Backup," on page 31, then do one of the following:
   - If NICI is not installed, restore all the registry information first.
   - If NICI is installed, remove NICI and overwrite the registry information from the backup store.
2. Restore the files and directories within `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI\ConfigDirectory` as selected by the operator.

It is recommended that all the files be restored as a group. But if you are knowledgeable, you can choose to restore individual entries. This can be done only if the `nicifk` and `xmgrcfg.wks` files did not change from the files in the backup store. If this is the case, be sure to adjust the access rights based on the new owner of the user configuration directories. The individual directories are named after the owner, but access rights are controlled by the SID. For example, just because a subdirectory is named BOB does not automatically mean that the current user BOB is the correct owner of the information being restored.

6.3.3 Special Cases for Windows

It is possible to configure the registry value `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI\UserDirectoryRoot` to indicate that the user configuration files be placed in the user's personal configuration directory. In this case, you should be prepared to back up and restore the user information independently as part of normal backup and restore operations. If NICI has been configured in this manner, you should be aware of it and be prepared to do individual backups.

This special case for the Windows user directory is enabled by creating the registry value EnableUserProfileDirectory rather than just pointing the directory path there. When the user profile directory is enabled, the directory might be automatically deleted when Windows is configured to automatically create and delete user accounts. In this case, backup and restore is necessary only for those specific users who are permanent.

The default path is the `Application Data\Novell\Nici` directory branch of the user's directory in Documents and Settings.

Error Resolution

7

This section provides NICI error messages and information on how to resolve the errors.

7.1 Error Messages

- "Error -1460: NICI_E_NOT_FOUND" on page 33
- "Error -1470: NICI_E_FIPS140CNRG_ERR" on page 33
- "Error -1471: NICI_E_SELF_VERIFICATION" on page 33
- "Error -1472: NICI_E_CRYPTO_DOWNGRADE" on page 33
- "Error -1494: NICI_E_NOT_INITIALIZED" on page 34
- "Error -1497: CCS_E_AUTHENTICATION_FAILURE" on page 34
- "NICI Module Corruption (NetWare): Abend" on page 34
- "Error -670 Error creating/fetching Security Domain key" on page 35

7.1.1 Error -1460: NICI_E_NOT_FOUND

If returned when trying to initialize NICI on a Windows platform, this error typically means that NICI is not installed, or the NICI device (in 1.*x* device driver versions) is not running. If the NICI device is not running, you can try to run it by entering `net start niciccs` on a Windows NT/ 2000 console. If it fails, reboot the system. Otherwise, reinstall NICI.

This error is returned when a security domain key (such as a tree key) is not found on the system. The API is CCS_GetPartitionKey. See Chapter 4, "NICISDI: Security Domain Infrastructure," on page 21 for more information.

7.1.2 Error -1470: NICI_E_FIPS140CNRG_ERR

This is an error in NICI's internal random number generator as defined by FIPS 140. NICI will try to recover, and returns this error if it can't. The solution is to retry, reload, or restart the application. We don't anticipate this error will occur.

7.1.3 Error -1471: NICI_E_SELF_VERIFICATION

This error condition was introduced with the FIPS 140-certified NICI, and is present regardless of the certification level of NICI on platforms other than NetWare®. Upon loading or being instantiated by a process, NICI runs a set of tests for module integrity as well as cryptographic process integrity. If one of these tests fails, NICI puts itself in an inoperable state and returns this error. The typical cause of this problem is module verification failure. The solution is to reinstall NICI, or to uninstall and then reinstall NICI.

7.1.4 Error -1472: NICI_E_CRYPTO_DOWNGRADE

This error was introduced in NICI version 2.0.1. The most likely cause is installation of a weak NICI version on a strong NICI installed base. The solution is to install strong NICI.

Novell® is shipping the strong NICI worldwide, and stopped shipping the import-restricted version with limited key sizes. We don't anticipate seeing this error anymore.

7.1.5 Error -1494: NICI_E_NOT_INITIALIZED

Similar to error -1497, this is typically caused by the lack of NICI license materials or configuration files. Reinstalling NICI typically solves the problem. If it does not, first try removing the NICI registry key on Microsoft Windows, deleting the UNIX `/etc/nici.cfg` configuration file, and then installing NICI. Reinstalling NICI does not remove existing keys. If this doesn't solve the problem and you won't lose data by deleting the NICI configuration files and keys, then delete the NICI configuration directory together with the registry on Microsoft Windows or the UNIX configuration file, and reinstall NICI.

7.1.6 Error -1497: CCS_E_AUTHENTICATION_FAILURE

Typical causes:

- Lack of NICI licensing materials (`.nfk` file copied to the `nicifk` file). NICI on servers (NetWare, DHost, or equivalent environment on other platforms) must have a NICI foundation key file in order to initialize key materials. NICI license materials are part of a Novell eDirectory™ license. Earlier NetWare installs had the option of installing eDirectory without licenses, which basically disabled NICI. With the new directory services introduced with Tao (eDirectory 8.5) for the first time, directory services uses NICI for a variety of cryptographic functionality, so a simple upgrade from an earlier version of directory services to a newer version renders directory services unusable because of NICI. NICI does not operate without a NICI licensing materials, or a proper configuration file. The solution is to install a license (this can be the installation of the same license), or copy the `.nfk` file from the license diskette to the `nicifk` file, then reboot the server or restart the DHost process.
- Lack of or corrupted NICI configuration files, especially on NetWare servers. A corrupted NICI configuration file is not fixable; it must be deleted. An effort was made to minimize this problem starting with NICI version 1.3.*x*. It is less likely for this to occur with NICI 2.*x* or later.
- Cryptography module downgrade.

7.1.7 NICI Module Corruption (NetWare): Abend

On NetWare, all NICI modules are signed NLM™ programs, and they have the `.xlm` extension. These modules are loaded by `xim.xlm`, which is in turn loaded by `xldr.xlm` as part of `server.exe` execution. The XIM module verifies multiple digital signatures during XLM loading. NetWare abends if any of the signatures is invalid. This is intentional, and not a problem or a bug. It makes sure that the cryptographic and key management modules are not tampered with, and that the module integrity is in place. We have seen corrupted XLMs because of CD burner and other copying problems.

The NICI license materials file (`nicifk`) is also signed. An invalid license file renders NICI dysfunctional.

7.1.8 Error -670 Error creating/fetching Security Domain key

This error is not unique to Novell eDirectory 8.6.0, but was first reported during Novell eDirectory version 8.6.0 upgrade testing, probably because servers are not rebooted during the Novell eDirectory version 8.6.0 upgrade, but directory services is restarted. The problem is duplicated in other environments by restarting directory services (without rebooting and allowing NICI to reinitialize) on servers listed in the W0 object.

Workarounds:

- Avoid restarting directory services on the servers listed in the W0 object without also initializing NICI.
- Restart the server identified by the W0 object before requesting the security domain key (A restart allows NICI to reinitialize, but you still need to be careful not to restart directory services).
- Upgrade to NICI version 2.4 or later.

Troubleshooting

8

The information in this section is provided to help you troublehsoot problems with NICI.

- Section 8.1, “Windows NT/2000: chkdsk,” on page 37
- Section 8.2, “NetWare 5.x and 6.x Install Issues,” on page 37

8.1 Windows NT/2000: chkdsk

NICI 1.*x* versions are implemented as a kernel driver on Windows systems. Because of an improper registry configuration, the `niciccs.sys` kernel driver on Windows NT/2000 systems might prevent a check disk (chkdsk) from running during system reboot (initial blue screen). Alternatively, the system might try to run chkdsk on every system reboot. The NICI version 1.5.5 install fixed this problem. However, you can also check the Windows NT/2000 registry to make sure that your system does not have this problem. Check the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NICICCS` key’s Start DWORD value. It must be set to 2 to prevent chkdsk volume access errors.

8.2 NetWare 5.*x* and 6.*x* Install Issues

The `nicisdi.xlm` module shipped as part of NICI 2.*x* or later does an improved job of authenticating and checking rights, among other enhancements in conjunction with Novell eDirectory™. Together with the directory services’ rights management changes at install time, some changes were inevitable, and backward compatibility is broken. This is an issue when installing a new server with a version of NICI earlier than 2.*x*, such as NetWare 5.*x* server, into a tree with a Security Domain Server (the server listed in the Security.KAP.W0 container) running NICI 2.*x* or later. The new server being installed into the existing tree fails when trying to connect and get a copy of the tree key. This error occurs during the final file copy and shows up as part of the certificate server installation.There will not be a fix for this error, because fixing it would reduce the overall security. However, there is a workaround. These steps assume that a NetWare 5.*x* server is installed into a 6.*x* tree.

1. Install the NetWare 5.1 server in its own tree.
2. Update to NICI 2.0.1 or later (the one that shipped with NetWare 6.0) after the installation is completed.
3. Uninstall the directory on the NetWare 5.*x* server.
4. Delete the `sys:system\nici\nicisdi.key` file.
5. Install the NetWare 5.*x* server into the NetWare 6.0 directory tree.
6. Create the server certificates via the PKI management console for this server.
7. Configure LDAP,/etc.

UNIX Installation File Locations

A

Installations of NICI v2.7.0 and later for UNIX platforms are LSB-compliant. The installation moves the existing installation (if any) to an LSB-compliant directory structure and makes links to the old directories in order to not break anything. It uses the following LSB directory structure:

Table A-1 *LSB Directory Structure*

Directory	**File/Directory Type**
`/etc/opt/novell`	Configuration file
`/var/opt/novell/nici`	License file and user directories
`/opt/novell/lib`	Library file
`/opt/novell/man`	Man pages

Symbolic links are used to provide compatibility with the paths used by previous versions of NICI.The following table shows the symbolic links by platform:

NOTE: The Man pages might have links from `/usr/share/man/*` to `/opt/novell/man/*`.

Table A-2 *Symbolic Links*

Platform	**Symbolic Link**
Linux	`/etc/nici.cfg--> /etc/opt/novell/nici.cfg`
	`/var/novell/nici--> /var/opt/novell/nici`
	`/usr/lib/libccs2.so--> /opt/novell/lib/libccs2.so.2.7.0`
	`/opt/novell/lib/libccs2.so--> /opt/novell/lib/libccs2.so.2.7.0`
Solaris	`/etc/nici.cfg--> /etc/opt/novell/nici.cfg`
	`/var/novell/nici--> /var/opt/novell/nici`
	`/usr/lib/libccs2.so--> /opt/novell/lib/libccs2.so.2.7.0`
	`/opt/novell/lib/libccs2.so--> /opt/novell/lib/libccs2.so.2.7.0`

Platform	Symbolic Link

HP_UX	`/etc/nici.cfg--> /etc/opt/novell/nici.cfg`
	`/var/novell/nici--> /var/opt/novell/nici`
	`/usr/lib/libccs2.so--> /opt/novell/lib/libccs2.so.2.7.0`
	`/opt/novell/lib/libccs2.so--> /opt/novell/lib/libccs2.so.2.7.0`

AIX	`/etc/nici.cfg--> /etc/opt/novell/nici.cfg`
	`/var/novell/nici--> /var/opt/novell/nici`
	`/usr/lib/libccs2.so--> /opt/novell/lib/libccs2.so.2.7.0`
	`/opt/novell/lib/libccs2.so--> /opt/novell/lib/libccs2.so.2.7.0`

Documentation Updates

B

The documentation was updated on the following dates:

- Section B.1, "July 24th, 2008," on page 41

B.1 July 24th, 2008

Updates were made to the following sections. The changes are explained below.

B.1.1 Overview

Location	**Change**
Entire Book	Made editorial changes and updated the guide to current Novell documentation standards.
Section 3.1, "NICI Configuration Files," on page 15	Added information to this section, including information regarding the following configuration files: ◆ `xmgrcfg.wks` ◆ `nicifk.new` ◆ `set_server_mode` ◆ `set_server_mode.bat`