opentext™

NICI
Administration Guide

January 2025

Legal Notice

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see https://www.netiq.com/company/legal/.

Copyright © 1996-2019 NetIQ Corporation, a Micro Focus company. All Rights Reserved.

Copyright © 1998-2017 Novell Corporation, a Micro Focus company. All Rights Reserved.

Copyright (c) 1998-2016 The OpenSSL Project. All rights reserved. For more information on OpenSSL License Information, see https://www.openssl.org/source/license.html.

 Copyright © 1990-2013 EMC Corporation. All rights reserved.

 FIPS 140-2 Inside.

TM: A Certification Mark of NIST, which does not imply product endorsement by NIST, the U.S. or Canadian Governments.

About this Book and the Library

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

Intended Audience

This book is intended for network administrators.

Other Information in the Library

NICI is available with OpenText eDirectory, OpenText Identity Console, and Novell Client. The documentation for these products is available at the following resources:

- OpenText Product Documentation Website (https://www.netiq.com/documentation/)
- Novell Client Documentation Website (https://www.novell.com/documentation/windows_client/)

Contents

About this Book and the Library 3

1 Overview 7

Understanding the NICI Modules . . . 7
NICI Modules on Linux Servers . . . 7
NICI Modules on Windows Servers . . . 8

2 What's New 9

Support for 8192 bit RSA Encryption . . . 9
FIPS Mode for NICI . . . 9
Operating NICI in FIPS Mode . . . 9
NICI SDI Health Check . . . 10
Creating an Inherited Rights Mask . . . 10
Dependency Change . . . 10

3 Installing NICI 11

Checklist for Installing NICI . . . 11
Understanding the NICI Installation . . . 11
Prerequisites and Considerations for Installing NICI . . . 11
System Requirements . . . 12
Installing NICI . . . 12
Installing NICI on Linux Servers . . . 13
Installing NICI on Windows Servers . . . 14
Meeting the LSB Compliant Directory Structure . . . 14
Locating the NICI Configuration Files . . . 15
Post-Installation Tasks . . . 16
Configuring the Settings for NICI User Directory . . . 16
Using NICI for Configuring System-Level FIPS Mode . . . 19

4 Upgrading NICI 21

Checklist for Upgrading NICI . . . 21
Prerequisites and Considerations for Upgrading NICI . . . 21
Considerations for Upgrading NICI on Servers with Multiple Instances of OpenText eDirectory . . . 22
Backing Up and Restoring the Current Configuration . . . 22

5 Understanding the NICI Keys 23

Types of Keys . . . 23
Understanding the Key Storage Key . . . 23
Understanding the Session Key . . . 24
Understanding the NICI SDI Key . . . 24
Understanding Tree keys . . . 24

Managing the Tree Keys 25
NDSPKI:SD Key Server DN 25
ACL 26
Synchronizing the NICI SDI Keys 26
Creating an AES 256-Bit Tree Key 27
NICI SDI Health Check 28
Merging Trees 29
Diagnosing SDI Key Synchronization Issues 29

6 Backing Up and Restoring NICI 31

Performing a Backup 31
Performing a Backup on Linux Systems 32
Performing a Backup on Windows Servers 33
Restoring NICI 33
Restoring NICI on Linux Systems 34
Restoring NICI on Windows Servers 35
Special Cases for Windows 35
Automatic Backing Up and Restoring NICI 36

7 Resolving Errors 37

Error Messages 37
Error -1473: NICI E-OpenSSL Failure 37
Error -1460: NICI_E_NOT_FOUND 37
Error -1470: NICI_E_FIPS140CNRG_ERR 37
Error -1471: NICI_E_SELF_VERIFICATION 38
Error -1472: NICI_E_CRYPTO_DOWNGRADE 38
Error -1494: NICI_E_NOT_INITIALIZED 38
Error -1497: CCS_E_AUTHENTICATION_FAILURE 38
Error -670 Error creating/fetching Security Domain key 39

A Documentation Updates 41

February 2018 41
January 2016 41
August 2008 41
Overview 41

1 Overview

Novell International Cryptography Infrastructure (NICI) is 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 (OpenText eDirectory), NetIQ Modular Authentication Service (NMAS), OpenText Certificate Server, and NetIQ SecretStore.

This document helps you to understand and administrator the NICI services which are used by various OpenText, or third-party products. A particular product might use NICI directly or indirectly via another module (DLL, `.so`).

NICI 3.2 (64-bit) is Federal Information Processing Standards (FIPS) 140-2 Inside validated to meet the security requirements of U.S. Federal agencies and customers with highly secure environments.

WARNING: Many actions described in the document could potentially 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, If certain NICI keys are irrecoverably lost, even backed-up data might be useless, because it can't be decrypted without the corresponding key.

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

Understanding the NICI Modules

When you install NICI, the installation process creates NICI modules on your server.

- "NICI Modules on Linux Servers" on page 7
- "NICI Modules on Windows Servers" on page 8

NICI Modules on Linux Servers

The following table lists the modules that the NICI installation process places on your Linux server.

Module	Description
`libccs2.so`	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.

Module	Description
libniciext.so	niciext is shipped with OpenText eDirectory and provides OpenText eDirectory applications with secure communication. It also provides a shared key for common secured storage. .

NICI Modules on Windows Servers

The following table lists the modules that the NICI installation process places on your Windows server.

Module	Description
ccswx64.dll (64-bit) ccs.dll (32-bit)	NICI is a shared library (dll) named ccswx64.dll or ccs.dll. It is typically installed in the %windir%\system32 directory. For example, on a 64-bit box: ◆ ccswx64.dll: c:\Windows\System32 ◆ ccs.dll: c:\Windows\SysWOW64 On a 32-bit box: ◆ ccs.dll: c:\Windows\System32 NICI does not depend on directory services to be installed.
niciextwx64.dlm	niciextwx64.dlm is shipped with OpenText eDirectory and provides OpenText eDirectory applications with secure communication. It also provides a shared key for common secured storage.

2 What's New

NICI 3.2 provides the following features:

- "Support for 8192 bit RSA Encryption" on page 9
- "FIPS Mode for NICI" on page 9
- "Operating NICI in FIPS Mode" on page 9
- "NICI SDI Health Check" on page 10
- "Creating an Inherited Rights Mask" on page 10
- "Dependency Change" on page 10

Support for 8192 bit RSA Encryption

NICI 3.2 (64 bit) supports 8192 bit RSA encryption. For more information, see Using 8192 Bit RSA Keys in Certificates in the *OpenText™ eDirectory CE 24.4 (v9.3) eDirectory Administration Guide*.

FIPS Mode for NICI

NICI 3.2 (64 bit) is FIPS 140-2 Inside validated using the OpenSSL FIPS Object Module. For more information, see the OpenSSL FIPS Object Module.

By default, FIPS mode for NICI is turned off, and must be turned on to operate NICI in FIPS 140-2 mode. For more information about putting NICI in FIPS mode, see "Operating NICI in FIPS Mode" on page 9. In case of OpenText eDirectory, see Operating OpenText eDirectory in FIPS Mode in the OpenText™ eDirectory CE 24.4 (v9.3) Installation Guide.

Operating NICI in FIPS Mode

There are 3 ways that NICI can be put into FIPS mode:

- **Application API:** A NICI enabled application can specify it wants to operate in FIPS mode by setting a parameter on the appropriate API. This gives each NICI enabled application the ability to specify that they are to operate in FIPS mode. Once an application is in FIPS mode it cannot change to a non-FIPS mode until it shuts down completely.

  Process API: A NICI enabled application can specify that all applications in the same process space are to operate in FIPS mode. To enable FIPS mode, an application just need to call the appropriate API. This allows a NICI enabled application to confirm that all applications in the same process space are operating in FIPS mode. For example, a single eDirectory application can call the API resulting in all NICI enabled applications operate in FIPS mode. Once a process space is put into FIPS mode, it cannot change to a non-FIPS mode until the process shuts down completely.

Computer level via NICI Configuration File: NICI can be configured so that all 64-bit NICI applications on that computer will operate in FIPS mode. For more information, see "Using NICI for Configuring System-Level FIPS Mode" on page 19.

The configuration is read and applied at application start-up time. Once an application is in FIPS mode it cannot change to a non-FIPS mode until it shuts down completely.

NICI SDI Health Check

In NICI 3.2, the NICI SDI module (`niciext`) now has a health check which runs each time that the `niciext` module is loaded. The heath check also runs if a new key is created. The output from the health check is output a file, and can also be seen via the `DSTrace` module.

Creating an Inherited Rights Mask

To fully support multiple SDI keys, the NICI health check process runs periodically and creates an inherited rights mask for the `CN=KAP.CN=Security` object. An inherited rights mask is created automatically to address a previous security right issues.

Dependency Change

NICI 3.2 (64-bit) no longer requires 32-bit NICI to be installed. Both 32-bit and 64-bit NICI share the same key files.

3 Installing NICI

This section provides the prerequisites, considerations, and system setup to install NICI in your environment.

Checklist for Installing NICI

It is recommended that you complete the steps in the following checklist:

	Checklist Items
❑	1. Determine which operating system platforms suit your installation. For more information, see "System Requirements" on page 12.
❑	2. Review the considerations for installing NICI. For more information, see "Prerequisites and Considerations for Installing NICI" on page 11.
❑	3. Understand how to install NICI. For more information, see "Understanding the NICI Installation" on page 11.
❑	4. Ensure access the installation files for NICI, located by default in the installation package of the products bundling NICI.
❑	5. Ensure that you have the appropriate permissions for installing NICI. For more information, see the following links: ◆ "Installing NICI as a Root User" on page 13 ◆ "Installing NICI as a Non-Root User With Root Access" on page 13
❑	6. Review the actions that you can perform after installing NICI, see "Post-Installation Tasks" on page 16.

Understanding the NICI Installation

Novell bundles NICI with OpenText eDirectory and OpenText Identity Console in the same installation programs. NICI is also available as part of Novell Client. However, NICI has its own installation program and can be installed separately. It is recommended that NICI be installed by the software program packaging NICI.

Prerequisites and Considerations for Installing NICI

Before installing NICI, review the following considerations:

- You must install NICI as a root user or a root-equivalent user.

- NICI is a shared library (DLL, .so). providing API services to applications. It does not run as a separate program or service.
- NICI provides additional capabilities to OpenText eDirectory via the niciext module that is shipped as part of the OpenText eDirectory distribution and that runs as an OpenText eDirectory service.
- You must install NICI on every workstation that use management utilities for OpenText eDirectory, such as OpenText Identity Console.
- NICI and OpenText eDirectory support RSA key sizes up to 8192 bits.
- You can install NICI separately or with OpenText eDirectory, OpenText Identity Console, or Novell Client on platforms supported by these products. For more information on supported platforms, see "System Requirements" on page 12.
- Uninstalling or reinstalling NICI does not destroy existing keys (i.e. key files are not removed from the file system during uninstall or during reinstall.
- Installing NICI 3.x does not require rebooting the server in most instances. However, if the NICI module (DLL or .so) is in use and cannot be overwritten by the installation program, a reboot might be necessary. Before installing NICI, all applications which are using NICI should be shut down to help avoid reboots.
- Installing a newer version of NICI over an existing NICI installation upgrades NICI. Always upgrade NICI using the NICI installation program (MSI or RPM). Do not copy NICI modules manually. Manual copying will result in a chaotic system, and may cause irreparable damage to the system and/or other products such as PKI, SecretStore/Single Sign-On, NMAS, directory services, etc.

System Requirements

You can install NICI separately or with OpenText eDirectory, OpenText Identity Console, or Novell Client on platforms supported by these products. For more information on which platforms you can install NICI, see the following links:

- OpenText eDirectory (https://www.netiq.com/documentation/edirectory-93/edir_install/)
- OpenText Identity Console (https://www.netiq.com/documentation/identity-console/identity_console-install/data/bookinfo.html)
- Novell Client (https://www.novell.com/documentation/windows_client/windows_client_installqs/data/windows_client_installqs.html)

Installing NICI

This section describes the process for installing NICI. Review the prerequisites and system requirements provided in "Prerequisites and Considerations for Upgrading NICI" on page 21 and "System Requirements" on page 12.

- "Installing NICI on Linux Servers" on page 13
- "Installing NICI on Windows Servers" on page 14

Installing NICI on Linux Servers

By default, the NICI installation file is packaged in the installation kit of the product bundling NICI. While installing the product, select the option to install NICI. Novell recommends installing NICI as root because the required NICI packages are used system-wide. However, if necessary you can delegate access to a different account using the Sudo utility and use that account to install the NICI packages.

NOTE: If you are using NICI 2.7.7, upgrade to the NICI 3.x 32-bit first then to 64-bit.

Depending on your product, follow the installation instructions from one of the following links:

- **OpenText eDirectory:** "Installing NICI" (https://www.netiq.com/documentation/edirectory-93/edir_install/data/a79kg0w.html#bjtfrfr) in the *OpenText eDirectory Installation Guide* (https://www.netiq.com/documentation/edirectory-93/edir_install/data/bookinfo.html)
- **OpenText Identity Console:** Deploying Identity Console (https://www.netiq.com/documentation/identity-console/identity_console-install/data/t498h6ca5qvc.html) in the Identity Console Installation Guide (https://www.netiq.com/documentation/identity-console/identity_console-install/data/bookinfo.html)
- **Novell Client:** "Installing the Novell Client for Windows" in the *Novell Client 2 SP4 for Windows Installation Quick Start*

If the setup program detects a previously installed version of NICI, it might give you the option to stop the installation process, remove, or upgrade the existing installation.

This section describes the following activities:

- "Installing NICI as a Root User" on page 13
- "Installing NICI as a Non-Root User With Root Access" on page 13

Installing NICI as a Root User

To install NICI, enter the following commands from 32-bit and 64-bit NICI directories:

- **32-Bit:** `rpm -ivh NICI_rpm_absolute_path/nici-3.2.0.i586.rpm`
- **64-Bit:** `rpm -ivh NICI_rpm_absolute_path/nici64-3.2.0.x86_64.rpm`

  NOTE: Upgrading OpenText eDirectory from 8.8 to 9.0, does not upgrade the 32-bit NICI.

Installing NICI as a Non-Root User With Root Access

To install NICI on Linux servers, you must login as `root` or have `root` access. The Sudo utility provides an easy way for a system administrator to delegate `root` access to other user to perform specific tasks by editing the `/etc/sudoers` configuration file and adding appropriate entries in it.

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

Most Linux providers have the Sudo utility available on their Web sites. You can also download the Sudo utility from the Sudo Web site. 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`.

The installation program might place the following files on your computer:

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

NICI uses these files to switch NICI into server mode when programs such as OpenText eDirectory are installed. To verify that NICI is set to server mode, you can enter `/var/opt/novell/nici/set_server_mode64`.

Installing NICI on Windows Servers

By default, the NICI installation file is packaged in the installation kit of the product bundling NICI. To install NICI on Windows, perform the following steps:

- **Installing 64-bit NICI:**
  - Double click on `NICI_wx64.msi`.
  - Click **Install**.
- **Installing 32-bit NICI:**
  - Double click on `NICI_x32.msi`.
  - Click **Install**.

Meeting the LSB Compliant Directory Structure

NICI installations for Linux platforms are LSB-compliant. NICI uses the following LSB directory structure:

Table 3-1 *LSB Directory Structure*

File/Directory Type	Directory
Configuration file (nici64.cfg or nici.cfg)	`/etc/opt/novell`
License file and user directories	`/var/opt/novell/nici`
Library file	`/opt/novell/lib64` (if NICI 64-bit is installed)
	`/opt/novell/lib` (if NICI 32-bit is installed)
Man pages	`/opt/novell/man`

Table 3-2 shows the symbolic links for Linux servers.

Table 3-2 *Symbolic Links*

Platform	Symbolic Link
Linux	`/opt/novell/lib/libccs2.so --> /opt/novell/lib/libccs2.so.3.0.0` `/opt/novell/lib64/libccs2.so --> /opt/novell/lib64/libccs2.so.3.0.0`

Locating the NICI Configuration Files

The installation program places the NICI configuration files in the directories listed in Table 3-3.

Table 3-3 *NICI Configuration Directory*

Platform	Library Location	NICI Directory	NICI User Directory
Windows	• **32-bit:** *%SystemRoot%*\System32 • **64-bit:** *%SystemRoot%*\SysWOW64	• **32-bit:** *%SystemRoot%*\System32\Novell\NICI • **64-bit:** *%SystemRoot%*\SysWOW64\Novell\NICI	• **32-bit:** *%SystemRoot%*\System32\Novell\NICI • **64-bit:** *%SystemRoot%*\SysWOW64\Novell\NICI
Linux	• **32-bit:** `/opt/novell/lib` • **64-bit:** `/opt/novell/lib64`	• **32-bit:** `/var/opt/novell/nici` • **64-bit:** `/var/opt/novell/nici`	• **32-bit:** `/var/opt/novell/nici/` • **64-bit:** `/var/opt/novell/nici/`

NOTE: NICI configuration files are shared by both 32-bit and the 64-bit NICI.

Table 3-4 lists additional files that the installation program places on all platforms.

Table 3-4 *NICI Configuration Files*

File	Created by	Description
`NICIFK`	NICI installation	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.

Table 3-5 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.

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

Post-Installation Tasks

In most configurations there is no need to perform any post-installation tasks. However, NICI can be configured to meet the policies and requirements defined by your business processes. Possible post-installation tasks include the following:

- "Configuring the Settings for NICI User Directory" on page 16
- "Using NICI for Configuring System-Level FIPS Mode" on page 19

NOTE: Both the options for Configuring the Settings for NICI User Directory and Using NICI for Configuring System-Level FIPS Mode are optional.

Configuring the Settings for NICI User Directory

NICI creates a new NICI user directory the first time a user uses NICI. NICI sets the rights on each user directory when it creates the directory, so that only the user has access to it.

The default directory for new NICI user directories is:

- **Linux:** `/var/opt/novell/nici/<uid of user>`
- **32-bit Windows:** `Windows\System32\Novell\NICI\<user>`
- **64-bit Windows:** `Windows\SysWOW64\Novell\NICI\<user>`

Changing the Permission of NICI User Directory on Linux Servers

The installation program places `nici64.cfg` (64-bit NICI) and `nici.cfg` (32-bit NICI) configuration files in the `/etc/opt/novell` directory on your Linux operating system.

The NICI configuration file emulates the Windows registry and is a minimally editable text file. Most of the entries in the file are set up when you install NICI and should not be modified. Modifications to some fields will leave NICI inoperable. The entries in the configuration file vary for a 32-bit NICI and 64-bit NICI. Nevertheless, a typical configuration file contains the following entries:

```
ConfigDirectory:s:20:/var/opt/novell/nici
SharedLibrary:s:19:/opt/novell/lib64/libccs2.so
DAC:b:8:1a:aa:6d:49:48:a8:83:98
MkUserDir:s:24:/var/opt/novell/nici/nicimud
NiciVersion:s:5:2.4.0
BuildDate:s:6:020123
NiciStrength:s:2:u0
RestrictionLevel:b:1:00
```

NOTE: For FIPS mode, only modify the last digit of the RestrictionLevel entry.

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) 20 characters. The value is `/var/opt/novell/nici`. Each line is described in Table 3-6.

Table 3-6 *Linux Key Values*

Key	Description
`MkUserDir`	NICI uses this executable to create user directories. `/var/novell/nici/nicimud` is supplied by the NICI installation program. (Do NOT modify)
`NICIVersion`	NICI version string. (Do NOT modify)
`BuildDate`	NICI module's build date; year, month, and day, each in two decimal digits. (Do NOT modify)
`NiciStrength`	u0 for strong, w1 for import restricted (no longer supported). (Do NOT modify)
`RestrictionLevel`	0 for no restriction, 1 for FIPS mode. (Modify only the last digit)
`NICISDI Sync Period`	(Optional) NICISDI synchronization period in minutes, represented in hexadecimal. (Not recommended)

The `libniciext.so` module reads the NICISDI sync period value when OpenText eDirectory loads it. If the value does not exist, or if the period is zero, the module uses an automatic sync period based on a sliding scale that starts with a heavy synchronization and moves towards lighter synchronization. If the value exists and contains a non-zero period, `libniciext.so` reads the value and uses it to determine synchronization periods.

NOTE: You should not use the optional sync period unless support directs you to do so.

The `/var/opt/novell/nici/`*`uid`*`/nicisdi.key` file contains the encrypted security domain keys as discussed in Chapter 5, "Understanding the NICI Keys," on page 23. For example, it is typically 0 for root. Having a `nicisdi.key` file for each user enables multiple instances of OpenText eDirectory running with different user IDs to host multiple trees on the same physical box.

NOTE: The UID is the variable numeric user ID defined by the Linux system.

All users have read and execute (where applicable) rights to the files in the NICI configuration directory (/var/opt/novell/nici). Only the user who installs NICI has full rights to the configuration directory. The setuid executable (nicimud for 32-bit NICI and nicimud64 for 64-bit NICI), creates the NICI directories for users. For example, nicimud will create a directory when a user first uses NICI and will give full rights to only the user creating the directory (0700).

Changing the Permission of NICI User Directory on Windows Servers

The NICI installation program creates and populates a key in the Windows registry. The registry key is different for 32-bit and 64-bit.

- 64-bit registry key: HKEY_LOCAL_MACHINE\SOFTWARE\Novell\nici_x64.
- 32-bit registry key:
  - On 32-bit machine: HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI
  - On 64-bit machine: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Novell\NICI

Table 3-7 *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
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	(Optional - Not Recommended) - NICISDI synchronization period in minutes, represented in hexadecimal.
EnableUserProfileDirectory	DWORD	(Optional) - NICI user files are created in the Application Data\Novell\NICI directory in the user's profile directory.
RestrictionLevel	Binary	(Optional 64-bit only) - 0 for no restriction, 1 for FIPS mode.

- By default, NICI creates users directories in the %SystemRoot%\System32\Novell\NICI (32-bit) and %SystemRoot%\SysWOW64\Novell\NICI (64-bit) directories by the user's name. For example, c:\windows\sysWOW64\novell\nici\administrator.
- If you want to change the root directory in which all user directories are created, navigate to the NICI registry key and create a String value using UserDirectoryRoot as the name and the desired root directory as the value.
- When creating a user directory, NICI uses the name of the user. 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).

- By default the `EnableUserProfileDirectory` key is not created and User Profile functionality is disabled. If you enable user profile functionality, you might need to copy or move the existing NICI user files to the new location. If the user profile directory is enabled, NICI does not set the ACLs on this directory, but 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 on your server and stores all NICI user files in this directory. NICI provides this option to support the dynamic user creation/deletion feature in the Novell ZENWorks® product. It must be set manually or by another application's installation, such as ZENWorks.
- `niciext.dlm` reads the nicisdi sync period value when OpenText eDirectory loads it. If the value does not exist, or if the period is zero, the module 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. We recommend that you do not set the sync period unless directed to by support.
- The `nicisdi.key` file contains encrypted security domain keys as discussed in Chapter 5, "Understanding the NICI Keys," on page 23.
- All users have read, execute, and create rights to the files in the NICI configuration directory (`<SystemRoot>\Novell\NICI`). NICI dynamically creates user directories when a user uses NICI for the first time and provides full rights only to the user creating the directory.

Using NICI for Configuring System-Level FIPS Mode

NICI 3.2 provides the ability to turn on FIPS mode at the computer level. When FIPS mode is turned on, all 64-bit NICI enabled applications, products, and services running on that computer will be able to perform cryptographic operations only using FIPS compliant algorithms. Any attempt to use a non-FIPS compliant algorithm will fail.

To enable the FIPS mode at the computer level, perform the following steps:

- **Linux:** Navigate to the `nici64.cfg` file and change **RestrictionLevel** to 1. This file is located in `/etc/opt/novell` on Linux.

  NOTE: Modify only the last digit of the Restriction Level setting from 00 to 01. Restart the OpenText eDirectory.

- **Windows:** Navigate to the `HKLM\SOFTWARE\Novell\nici_x64` registry and change the restriction Level settings from 0 to 1.

  You will need to perform this action on each server in your tree that you wish to set each server into FIPS mode. Once done, restart the OpenText eDirectory.

IMPORTANT: Enabling FIPS mode on NICI affects all applications that use NICI on that server. If these applications are not supported in FIPS mode, they might not work properly. Novell recommends that you do not use FIPS mode for NICI in OpenText eDirectory 9.1 to 9.2.6 versions. It supports from OpenText eDirectory 9.2.7.

4 Upgrading NICI

This section provides information to help you prepare for upgrading your NICI to the latest version. To install or upgrade the NICI, log in as root or a root-equivalent on the computer where you want to install or upgrade NICI.

Checklist for Upgrading NICI

To perform the upgrade, it is recommended that you complete the steps in the following checklist:

	Checklist Items
❑	1. Learn about version upgrades and compatibility of versions. For more information, see "Prerequisites and Considerations for Upgrading NICI" on page 21.
❑	2. Ensure that you have the latest installation kit to upgrade NICI. For more information, see "Installing NICI" on page 12.
❑	3. Ensure that your computer meets the hardware and software prerequisites for a newer version of NICI. For more information, see "Prerequisites and Considerations for Upgrading NICI" on page 21.
❑	4. Back up the keys and user data in the file system and in system-specific and user-specific directories and files. For more information, see "Backing Up and Restoring the Current Configuration" on page 22.
❑	5. Install or upgrade NICI to the latest version For more information, see one of the following sections: ◆ **OpenText eDirectory:** Installing NICI (https://www.netiq.com/documentation/edirectory-93/edir_install/data/a79kg0w.html#bjtfrfr) in the *OpenText eDirectory Installation Guide* (https://www.netiq.com/documentation/edirectory-93/edir_install/data/bookinfo.html) ◆ **OpenText Identity Console:** Deploying OpenText Identity Console (https://www.netiq.com/documentation/identity-console/identity_console-install/data/t498h6ca5qvc.html) in the OpenText Identity Console Installation Guide (https://www.netiq.com/documentation/identity-console/identity_console-install/data/bookinfo.html)

Prerequisites and Considerations for Upgrading NICI

Before upgrading NICI, review the following considerations:

- Reinstalling NICI does not destroy existing keys.

- Installing NICI 3.x does not require rebooting the server in most instances. However, if the NICI module (`DLL or .so`) is in use and cannot be overwritten by the installation program, a reboot might be necessary. Before installing NICI, all applications using NICI should be shut down to help avoid reboots.
- Installing a newer version of NICI over an existing NICI installation upgrades NICI. Always upgrade NICI using the NICI installation program (MSI or RPM). Do not copy NICI modules manually. Manual copying will result in a chaotic system, and will often cause irreparable damage to the system and/or other products such as PKI, SecretStore/Single Sign-On, NMAS, directory services, etc.

Considerations for Upgrading NICI on Servers with Multiple Instances of OpenText eDirectory

If a Linux server is hosting more than one OpenText eDirectory, each OpenText eDirectory instance typically has its own NICI directory setup. If you upgrade one instance of OpenText eDirectory to 9.0 or later, you will need to upgrade all instances because OpenText eDirectory 8.x will not work correctly using NICI 3.x.

It is recommended to run each instance of OpenText 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).

Backing Up and Restoring the Current Configuration

For information on backing up and restoring NICI, see Chapter 6, "Backing Up and Restoring NICI," on page 31.

5 Understanding the NICI Keys

To help applications securely store and transfer data and keys, NICI provides three types of keys - Key Storage key, NICI Security Domain Infrastructure (SDI) key, and Session key (SASDFM).

- The Key Storage key is a server specific key. This key is unique to the server it is created on, and is intended to be used to securely wrap keys for either internal or external storage. NICI creates this key for the server on which NICI is installed.
- A NICI SDI key is shared by all the servers within a security domain. In OpenText eDirectory a security domain consisting of the whole tree has been established and the associated key is often referred to as the Tree key or sometimes the `W0` key (as the object used to manage this key is `CN=W0.CN=KAP.CN=Security`). In NICI 3.x and OpenText eDirectory 9.x, we have added support for a new AES-256 bit tree key (or W1 key). However, this key is not enabled by default because all servers in the tree must be OpenText eDirectory 9.x or later to support it. All the servers in an OpenText eDirectory tree have the rights to acquire the Tree key.

  Access to SDI keys is governed by OpenText eDirectory rights and attributes. There is a specific set of rights and attributes that allow a server to create and distribute an SDI key. A server with this set of rights and attributes is known as a “Key server”. There is a different set of rights and attributes that allows a server to acquire keys from a Key server.
- NICI provides a Session key (or SASDFM key) to securely communicate between client and server.

Types of Keys

- “Understanding the Key Storage Key” on page 23
- “Understanding the Session Key” on page 24
- “Understanding the NICI SDI Key” on page 24
- “Understanding Tree keys” on page 24

Understanding the Key Storage Key

The server storage key is a computer-specific key. Each server creates a server storage key which is unique to that server and which can be used to securely wrap other keys for either local or remote storage. After a key is wrapped with a server storage key, only code on that server can unwrap the key, this allows the wrapped key to be remain secure even when stored remotely.

Prior to NICI 3.x, the storage keys were Triple Data Encryption algorithm (3DES) keys. NICI 3.x creates AES 256-bit storage keys. Any application that uses the storage keys to securely wrap other keys should be able to handle the new algorithm to encrypt new data. However, any data which is currently wrapped with the older 3DES keys will still be assessable without any changes.

Understanding the Session Key

To securely send data from a client to a server or server to server or vice versa, NICI provides the SASDFM keys which serve as a session key. Prior to NICI 3.x, the session keys were 3DES keys. NICI 3.x supports AES 256-bit session keys, as well as 3DES keys, depending on the capabilities of the applications.

- The client application and the OpenText eDirectory server will use the AES 256-bit session key only if both of them use NICI 3.x.
- If either of them uses a lower version of NICI, they will use a 3DES session key.

Understanding the NICI SDI Key

NICI SDI (Security Domain Infrastructure) is an OpenText eDirectory service which provides and manages shared keys for all servers within a security domain. Access to SDI keys is governed by OpenText eDirectory rights and attributes. There is a specific set of rights and attributes that allow a server to create and distribute an SDI key. A server with this set of rights and attributes is known as a `Key server`. There is a different set of rights and attributes that allows a server to acquire keys from a Key server.

NICI SDI can manage multiple keys of varying strengths and algorithms. Each SDI key can have a different security domain and is controlled by the OpenText eDirectory rights and attributes of the OpenText eDirectory object representing the SDI key known as the SDI key object:

- **Linux:** `libniciext.so`
- **Windows:** `niciext64.dlm`

The security domain keys are not intended for clients.

Understanding Tree keys

Tree keys are a special kind of NICI SDI key. The security domain for tree keys consists of the whole OpenText eDirectory tree, and they are automatically managed by OpenText eDirectory and NICI SDI.

In all OpenText eDirectory versions prior to 9.0, a single security domain consisting of the whole tree has been established and the associated key is often referred to as the Tree key or sometimes the `W0` key (as the SDI key object used to manage this key is `CN=W0.CN=KAP.CN=Security`). This key is a 3DES key, and all the servers in an OpenText eDirectory tree have the rights to acquire this key. This key will continue to be available.

Beginning in OpenText eDirectory 9.0 with NICI 3.0, OpenText eDirectory supports the creation of a new AES 256-bit Tree key. The SDI key object used to manage this new Tree key is `CN=W1.CN=KAP.CN=Security`. It is required that all servers in the tree be upgraded to OpenText eDirectory 9.x before enabling this key. Although OpenText eDirectory 9.x will automatically create this SDI key object, it will not assign a `Key server` and the key will not get created by default. An administrator will need to assign a Key server to the SDI key object, after confirming that all servers in the tree have been upgraded to OpenText eDirectory 9.x, in order to enable the new AES 256-bit Tree key. For more information, see “Creating an AES 256-Bit Tree Key” on page 27.

Although any server can be configured as a Key server for the tree keys, it is recommended that only servers holding a writeable replica of the SDI key object be assigned. It is recommended that the first Key server assigned be the Master replica (for example, the server holding the Master replica of the object `CN=W1.CN=KAP.CN=Security`).

NICI SDI supports having multiple Key servers for any SDI key and it is recommended that multiple Key servers be assigned. In NICI 3.x, once a Key server has been assigned to the Tree key objects, the new Heath-Check feature will automatically add servers holding a writeable replica of the SDI key object). The idea here is that NICI SDI will automatically mirror the Key servers to your OpenText eDirectory replicas.

Various services rely on the availability of Tree keys, including but not limited to SecretStore/Single-Sign-On, PKI (Certificate Server), and NMAS.

NOTE: The NICISDI module is different from the SASDFM module. SASDFM manages session keys between two physical boxes, typically between a client and a server.

Managing the Tree Keys

Tree keys are a special kind of NICI SDI key. Like all SDI keys, tree keys are managed by the associated SDI key object and it's attributes. There are now two Tree key objects, `CN=W0.CN=KAP.CN=Security` which manages the older 3DES Tree key (or the W0 key), and `CN=W1.CN=KAP.CN=Security` which manages the new AES 256-bit Tree key (or the W1 key).

The new object `CN=W1.CN=KAP.CN=Security` will get created when a server holding a writable replica of the `CN=KAP.CN=Security` container is upgraded to OpenText eDirectory 9.x. Although OpenText eDirectory 9.x automatically creates this SDI key object, it will not assign a Key server and the key will not get created by default. An administrator will need to assign a Key server to the SDI key object, after confirming that all servers in the tree have been upgraded to OpenText eDirectory 9.x, in order to enable the new AES 256-bit Tree key. For more information, see "Creating an AES 256-Bit Tree Key" on page 27.

NOTE: Do not delete the W0 object after upgrading the tree.

The W0 and W1 objects contain the following attributes:

- NDSPKI:SD Key Server DN
- ACL

NDSPKI:SD Key Server DN

The NDSPKI:SD Key Server DN attribute is used to determine which servers are Key servers and can create and distribute the SDI key.

This multivalued attribute on the W0 or W1 objects contains the list of the key servers in the tree for the respective SDI key object. There must be at least one server in this list for the SDI key object to be active. The `niciext` module reads this attribute and then connects to each server in this list and requests any new security domain keys from each server in this list. Only servers in this list can create and distribute the tree key.

Adding a server to this attribute makes that server a Key server. Although any server can be configured as a “Key server”, for the tree keys, it is recommended that only servers holding a writeable replica of the SDI key object be configured.

NOTE: If a key server does not hold a writeable replica, additional rights will need to be assigned.

The OpenText eDirectory Install will automatically populate this attribute for the W0 object, so no action is required by an administrator for the W0 object. For the W1 object, an administrator will need to assign a Key server to this attribute, after confirming that all servers in the tree have been upgraded to OpenText eDirectory 9.x, in order to enable the new AES 256-bit Tree key. It is recommended that the first Key server assigned be the Master replica (for example, the server holding the Master replica of the object `CN=W1.CN=KAP.CN=Security`).

ACL

The ACL attribute is used to determine which servers can acquire a copy of the key. Servers need the Read right to the [All Attributes Rights] attribute and the Browse rights to the object in order to be able to acquire a copy of the key.

In 3.x, the NICI SDI health check process runs periodically and creates an inherited rights mask for the KAP.Security object. The inherited rights mask is created automatically to make sure only servers which have been specifically granted rights to a SDI key object are able to acquire the key (objects are not allowed to inherit the rights necessary to acquire an key, they must be specifically assigned the rights).

Synchronizing the NICI SDI Keys

The NICI SDI module (`niciext`) automatically syncs the NICI SDI keys. Normally the module uses an automatic sync period based on a sliding scale that starts with a heavy synchronization and moves towards lighter synchronization.

If a new SDI key is enabled or created, you can speed the synchronization process to each of the servers in the tree by using one of the following methods on each server in the tree:

- **Linux:** Unload and reload the NICI SDI module (`niciext`) using `ndstrace`.
- **Windows:** Use the DHost console to reload and `niciext` module.
- Restarting OpenText eDirectory.
- Rebooting the server.

There is an optional (not recommenced) configuration setting that can adjust the synchronization period. See “Configuring the Settings for NICI User Directory” on page 16 for more details.

NOTE: You should not use the optional sync period unless support directs you to do so.

Creating an AES 256-Bit Tree Key

Tree keys are a special kind of NICI SDI key and are available to all servers in the tree. When multiple servers need access to the same encrypted data, OpenText eDirectory uses the Tree keys to provide access while still keeping the data secure in conjunction with OpenText eDirectory rights. In all prior versions of OpenText eDirectory a single security domain consisting of the whole tree has been established and the associated key is often referred to as the Tree key or sometimes the W0 key (as the SDI key object used to manage this key is `CN=W0.CN=KAP.CN=Security`). This key is a 3DES key, and all the servers in an OpenText eDirectory tree have the rights to acquire this key. This key will continue to be available.

Beginning in OpenText eDirectory 9.0 with NICI 3.0, it supports the creation of a new AES 256-bit Tree key. The SDI key object used to manage this new Tree key is `CN=W1.CN=KAP.CN=Security`. This key will be known as the W1 key. It is required that all servers in the tree be upgraded to OpenText eDirectory 9.x before enabling this key. Although OpenText eDirectory 9.x will automatically create this SDI key object, it will not assign a Key server and the key will not get created by default. An administrator will need to assign a Key server to the SDI key object, after confirming that all servers in the tree have been upgraded to OpenText eDirectory 9.x, in order to enable the new AES 256-bit Tree key.

IMPORTANT: ◆Do not create an AES 256-bit key unless all servers in your tree are upgraded to 9.x.

- Creating an AES 256-bit key with OpenText Identity Manager causes all passwords to be re-synced. For more information, see Re-encrypting Data with AES 256-Bit NICI SDI Key in the *OpenText™ eDirectory CE 24.4 (v9.3) eDirectory Administration Guide*.

When a server holding the master replica of the KAP.Security container is upgraded to OpenText eDirectory 9.x, OpenText eDirectory install creates a W1 object in this container. When all servers in the tree are upgraded to OpenText eDirectory 9.x, the tree administrator can create an AES 256-bit SDI key

1. Log in to the OpenText Identity Console as an administrator with the appropriate rights and goto **Tree View**.
2. Goto **Security Container** under tree view.
3. Click **KAP**.
4. Browse and select the **W1** security object.
5. Click **OK**.
6. In the window that displays, add NDSPKI:SD Key Server DN attribute and set the value to the DN of a server holding a master replica of the partition that contains the W1.KAP.Security object.
7. To create the AES 256-bit SDI key, trigger the NICI health check by performing one of the following actions:
   - **Linux:** Unload and reload the NICI SDI module (`niciext`) using ndstrace.
   - **Windows:** Use the DHost console to reload and `niciext` module.
   - Restart OpenText eDirectory.
   - Restart server.

After the AES 256-bit SDI key is created, the new key will automatically be synchronized to all servers in the tree using the normal synchronization schedule. If the servers in the tree have been up for some time, the automatic synchronization process is likely to be slow because SDI keys are synchronized on a sliding scale depending on how long the SDI module has been running. You can speed the synchronization process to each of the servers in the tree by using one of the following methods on each server in the tree:

- **Linux:** Unload and reload the NICI SDI module (`niciext`) using ndstrace.
- **Windows:** Use the DHost console to reload and `niciext` module.
- Restart OpenText eDirectory.
- Restart server.

IMPORTANT: The NICI SDI key is available to all servers in the tree. Therefore, you must upgrade all servers in the tree to NICI 3.0 before creating the AES 256-bit SDI key.

NICI SDI Health Check

In NICI 3.x, the NICI SDI module (`niciext`) now has a health check which runs each time that the `niciext` module is loaded. The heath check also runs if a new key is created. The output from the health check is output to `NICIext_Health.log` file located in the normal OpenText eDirectory log directory. In addition, the output can be seen in `DSTrace`, if enabled.

The NICI SDI health check process performs the following tasks:

- Creates an inherited rights mask for the KAP.Security object.

  NOTE: The inherited rights mask is created automatically to address a security rights issue.

- Automatically adds servers with a Writable replica of the W0 object to be key servers for the W0 object.
- Automatically creates a new W1.KAP.Security object. NICI uses this object to represent and administer rights to the new AES 256-bit SDI key.

  IMPORTANT: NICI does not automatically create the new AES 256-bit SDI key until a tree administrator performs a specific configuration operation.

- Checks to see if a key server has been assigned to the W1 object. Only if a key server has been assigned, the NICI health Check utility will add servers with a Writable replica of W1 object to be Key Servers for W1. For more information, see “Creating an AES 256-Bit Tree Key” on page 27.
- Checks to see if a key server has been assigned to the W1 object. Only if a key server has been assigned, the NICI health check will mirror the rights for the W0 object to the new W1 object, which will allow all servers in the tree to get access to the new AES 256-bit SDI key.

Merging Trees

In order for a tree merge to be successful, all servers will need to have a copy of the tree key(s) from all the original trees.

Before merging, you should identify the key servers for each of the tree keys (W0 and W1) in all the trees.

After merging, for each of the tree keys (W0 and W1) you should add the names of all the SD key servers to the corresponding object from all the merged trees. For each object (W0 and W1), the final list must contain the names of the SD key servers, for that object, from all the trees being merged.

NOTE: If either of the trees have been created in AES 256-bit tree key, then all servers in both trees MUST be upgraded to OpenText eDirectory 9.x before merging.

Diagnosing SDI Key Synchronization Issues

The addition of the NICI SDI Health check will minimize synchronization issues. Output from the health check will help to identify any issues that can't be automatically fixed by the health check. In addition, more detailed synchronization messages can be seen and captured in `DSTrace`.

Regardless of the platform/OS, the nicisdi.key file is server-unique and should not be copied from one machine to another. Manual creation of a new key typically causes more problems by introducing a new key on the server.

With the advent of the SDI Health check, NICI SDI is designed to fix itself. However, we will continue to provide SDIDiag as a Security Domain Infrastructure diagnostic and repair utility. Among other things, SDIDiag allows an administrator to:

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

For information about 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 OpenText 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.

---

6 Backing Up and Restoring NICI

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 mechanisms 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 Linux 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.

- "Performing a Backup" on page 31
- "Restoring NICI" on page 33
- "Automatic Backing Up and Restoring NICI" on page 36

Performing a Backup

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.

- "Performing a Backup on Linux Systems" on page 32
- "Performing a Backup on Windows Servers" on page 33

Performing a Backup on Linux Systems

In NICI 3.x, the `/var/opt/novell/nici` directory contains all the system and user directories and files.

Most NICI configuration files are located in the `var/opt/novell/nici` directory. A few NICI configuration files are located in the `/etc/opt/novell` 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.

Some of the critical NICI configuration files listed in Table 3-4 are unique to a specific user. Most configuration files are contained within the `/var/opt/novell/nici` directory, which contains common files. Files unique to specific users are contained within subdirectories of this 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.

You must back up the directories and files listed in Table 6-1. Remember to preserve the rights on all the directories and files.

Table 6-1 *Directories and Files for Backup*

Directory/File Name	File Type and Special Instructions
`/etc/opt/novell/nici.cfg` (32-bit)	Configuration file.
`/etc/opt/novell/nici64.cfg` (64-bit)	
`/opt/novell/lib/libccs2.so.*` or `/opt/novell/lib64/libccs2.so*`	NICI library. The version of the library completes the name.
`/var/opt/novell/nici`	Contains all the system keys, user directories, and files, and the programs used to initialize NICI.

On servers where you installed OpenText eDirectory, you can backup the NICI keys using the DSBK utility as instructed in Backing Up NICI in the *OpenText™ eDirectory CE 24.4 (v9.3) eDirectory Administration Guide*.

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 3-4 on page 15 for a list of the configuration files.

Performing a Backup on Windows Servers

Configuration information is kept in the Windows registry. The registry key is different for 32-bit and 64-bit:

- **64-bit registry key:** `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\nici_x64`
- **32-bit registry key:**
  - On 32-bit OS: `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI`
  - On 64-bit OS: `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Novell\NICI`

1. Backup all registry information under the appropriate register key(s) for your installation. If you have both 32-bit and 64-bit NICI installed you will need to backup both registry keys.
2. Backup the directory, including subdirectories, identified by

   ```
   [Your registry key]\ConfigDirectory
   ```

   Both 32-bit and 64-bit NICI share the same config directory, so if both are installed, this value will be the same for both and you will only need to backup one directory.

On servers where you installed OpenText eDirectory, you can backup the NICI keys using the DSBK utility as instructed in Backing Up NICI in the OpenText™ eDirectory CE 24.4 (v9.3) eDirectory Administration Guide.

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.

NOTE:

- Novell recommends to backup NICI using `dsbk` with the `-e` option to ensure that NICI is not in use while performing the backup.
- If `dsbk` is not used, at least OpenText eDirectory should be shutdown before performing the backup.

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.

- "Restoring NICI on Linux Systems" on page 34
- "Restoring NICI on Windows Servers" on page 35
- "Special Cases for Windows" on page 35

Restoring NICI on Linux Systems

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.
4. On server where you installed OpenText eDirectory, you can restore the NICI keys using the DSBK utility as instructed in Restoring NICI in the *OpenText™ eDirectory CE 24.4 (v9.3) eDirectory Administration Guide*.

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.

Review the following guidelines for each file/directory before restoring NICI if NICI is already installed on the server:

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

Restoring NICI on Windows Servers

1. Determine if NICI is already installed on the server by searching the registry for the NICI registry keys mentioned in "Performing a Backup on Windows Servers" on page 33, 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 `[You registry key]\ConfigDirectory` as selected by the operator.
3. On server where you installed OpenText eDirectory, you can restore the NICI keys using the DSBK utility as instructed in Restoring NICI in the *OpenText™ eDirectory CE 24.4 (v9.3) eDirectory Administration Guide*.

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.

Special Cases for Windows

It is possible to configure the registry value `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\NICI\UserDirectoryRoot` 32-bit or `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\ nici_x64\UserDirectoryRoot` 64-bit to indicate that the user configuration files are to 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.

Automatic Backing Up and Restoring NICI

In order to make NICI as stable and self-healing as possible, NICI creates a local backup every time it is successfully ran. The advantage of this is NICI can often automatically detect and recover in the unlikely event of data corruption. The fact that NICI configuration and data files are generally fairly static, means that in most cases no data will be lost even if an automatic recovery is needed.

Although the fact that NICI creates a local backup will eliminate most data corruption issues, it is recommended that you continue to regularly perform an off-line backup of NICI.

7 Resolving Errors

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

Error Messages

- "Error -1473: NICI E-OpenSSL Failure" on page 37
- "Error -1460: NICI_E_NOT_FOUND" on page 37
- "Error -1470: NICI_E_FIPS140CNRG_ERR" on page 37
- "Error -1471: NICI_E_SELF_VERIFICATION" on page 38
- "Error -1472: NICI_E_CRYPTO_DOWNGRADE" on page 38
- "Error -1494: NICI_E_NOT_INITIALIZED" on page 38
- "Error -1497: CCS_E_AUTHENTICATION_FAILURE" on page 38
- "Error -670 Error creating/fetching Security Domain key" on page 39

Error -1473: NICI E-OpenSSL Failure

This is an error generated by the FIPS OpenSSL libraries that NICI uses.

At this time, we do not have any specific workaround for this error code. Contact support for more details.

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.

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 5, "Understanding the NICI Keys," on page 23 for more information.

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. For more details, contact support.

Error -1471: NICI_E_SELF_VERIFICATION

This error condition was introduced with the FIPS 140-certified NICI. 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.

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.

OpenText is shipping the strong NICI worldwide, and stopped shipping the import-restricted version with limited key sizes.

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 Linux `/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 Linux configuration file, and reinstall NICI.

Error -1497: CCS_E_AUTHENTICATION_FAILURE

Typical causes:

- Lack of NICI licensing materials (`.nfk` file copied to the `nicifk` file). NICI on servers (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 OpenText eDirectory license. 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. 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.

Error -670 Error creating/fetching Security Domain key

This error is not unique to OpenText eDirectory 8.6.0, but was first reported during OpenText eDirectory version 8.6.0 upgrade testing, probably because servers are not rebooted during the OpenText 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.

A Documentation Updates

The documentation was updated on the following dates:

- "February 2018" on page 41
- "January 2016" on page 41
- "August 2008" on page 41

February 2018

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

Location	Change
Entire Book	NICI version was updated as part of the NICI 3.1 release.

January 2016

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

Location	Change
Entire Book	Revamped the guide for logical organization of the content.

August 2008

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

Overview

Location	Change
Entire Book	Made editorial changes and updated the guide to current Novell documentation standards.

Location	Change
Chapter 3, “Installing NICI,” on page 11	Added information to this section, including information regarding the following configuration files: ◆ `xmgrcfg.wks` ◆ `nicifk.new` ◆ `set_server_mode` ◆ `set_server_mode.bat`