Novell International Cryptographic Infrastructure (NICI)

2.6x

www.novell.com

ADMINISTRATION GUIDE

December 19, 2003

Novell®

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

You may not export or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside.

U.S. Patent No. 5,157,663; 5,349,642; 5,455,932; 5,553,139; 5,553,143; 5,572,528; 5,594,863; 5,608,903; 5,633,931; 5,652,854; 5,671,414; 5,677,851; 5,692,129; 5,701,459; 5,717,912; 5,758,069; 5,758,344; 5,781,724; 5,781,733; 5,784,560; 5,787,439; 5,818,936; 5,828,882; 5,832,274; 5,832,275; 5,832,483; 5,832,487; 5,859,978; 5,870,561; 5,870,739; 5,873,079; 5,878,415; 5,884,304; 5,893,118; 5,903,650; 5,903,720; 5,905,860; 5,910,803; 5,913,025; 5,913,209; 5,915,253; 5,925,108; 5,933,503; 5,933,826; 5,946,002; 5,946,467; 5,956,718; 5,956,745; 5,964,872; 5,974,474; 5,983,223; 5,983,234; 5,987,471; 5,991,810; 6,002,398; 6,014,667; 6,016,499; 6,023,586; 6,029,247; 6,052,724; 6,061,726; 6,061,740; 6,061,743; 6,065,017; 6,081,774; 6,081,814; 6,094,672; 6,098,090; 6,105,062; 6,105,069; 6,105,132; 6,115,039; 6,119,122; 6,144,959; 6,151,688; 6,157,925; 6,167,393; 6,173,289; 6,216,123; 6,219,652; 6,233,859; 6,247,149; 6,269,391; 6,286,010; 6,308,181; 6,314,520; 6,324,670; 6,338,112; 6,345,266; 6,353,898; 6,424,976; 6,466,944; 6,477,583; 6,477,648; 6,484,186; 6,496,865; 6,510,450; 6,516,325; 6,519,610; 6,532,451; 6,532,491; 6,539,381; RE37,178. Patents Pending.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.

www.novell.com

NICI 2.6x Administration Guide
December 19, 2003

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

Third-Party Trademarks

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.

- Chapter 1, “Introduction,” on page 9
- Chapter 2, “NICI Modules,” on page 11
- Chapter 3, “NICI Setup,” on page 13
- Chapter 4, “NICISDI: Security Domain Infrastructure,” on page 17
- Chapter 5, “Error Resolution,” on page 21
- Chapter 6, “Installing and Upgrading,” on page 25

Documentation Updates

For the most recent version of the *NICI 2.6x Administration Guide*, see the NICI Administration Guide Web site (http://www.novell.com/documentation/lg/nici20/treetitl.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.

1 Introduction

Novell® International Cryptography Infrastructure (NICI) is Novell's 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 may 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 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%\System 32\Novell\NICI (See Chapter 3, "NICI Setup," on page 13)	%SYSTEMROOT%\System 32\Novell\NICI (See Chapter 3, "NICI Setup," on page 13)
UNIX*	/usr/lib	/var/novell/nici (See Chapter 3, "NICI Setup," on page 13)	/var/novell/nici/ (See Chapter 3, "NICI Setup," on page 13)

2 NICI Modules

NICI is shipped or will be shipped on multiple platforms, including NetWare®, Windows 95/98/Me/NT/2000, Linux* (ix86), Solaris* (SPARC*), and AIX*. 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.

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 loaded by 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. It is a typical symptom to get a 1460 error when this module is not loaded.

- sasdfm.xlm (sasdfm.nlm)

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

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 not maintained anymore.

- 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.

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 of behalf of NICI. The shared object is shipped with eDirectory distributions.

3 NICI Setup

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, perhaps remove other items such as certificates that a customer has purchased, and reinstall NICI properly.

NICI Configuration Files

NICI configuration files are located in the NICI directories listed on Table 1, "NICI Configuration Directory" on page 7. 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.

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.

NICI User Configuration Files

A NICI user directory is created by NICI 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.

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.

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. 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 ys:/system/nici/nicisdi.key file contains encrypted security domain keys as discussed in Chapter 4, “NICISDI: Security Domain Infrastructure,” on page 17.

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.

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)

Key	Type	Description
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 Application Data\Novell\NICI directory in the user's profile directory.

Users' directories are created, by default, in %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 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 17.

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.

UNIX Configuration (/etc/nici.cfg)

The /etc/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/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 column (:). 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/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, if it is not covered in the Windows registry section.

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 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 sys:\system\nici\<uid>\nicisdi.key file contains the encrypted security domain keys as discussed in Chapter 4. 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 per 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/novel/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 give full rights only to the user creating the directory (0700).

4 NICISDI: Security Domain Infrastructure

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 has nothing to do with the SASDFM module. SASDFM manages session keys between two boxes, typically between a client and a server. The modules are both loaded during autoexec.ndf 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.

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 “Directory Objects” on page 17 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 revoking 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.

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:

NDSPKI:SD Key Server DN

This multi-valued 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.

NDSPKI:SD Key List

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

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 13 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 would 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.

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.

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

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. A 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 not to use this NLM, and let NICI 2.4 or later manage such keys.

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

5 Error Resolution

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

Error Messages

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 17 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. We don't anticipate this error will occur.

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.

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.

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 don'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.

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 that basically disabled NICI. With the new directory services introduced with Tao for the first time, DS uses NICI for a variety of cryptographic functionality, so a simple upgrade from an earlier version of DS to a newer version renders DS 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 is thrown away. 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.

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.

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 DS is restarted. The problem is duplicated in other environments by restarting DS (without rebooting and allowing NICI to reinitialize) on servers listed in the W0 object.

Workarounds:

- Avoid restarting DS 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 will allow NICI to reinitialize, but you still need to be careful not to restart DS.)
- Upgrade to NICI version 2.4 or later.

6 Installing and Upgrading

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.

Version Upgrade 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™, DS, 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 co-exist with newer NICI versions. If you want to do this, always install the newer version after the old version of NICI. For example, install NICI 2.4 after NICI 1.5.7.

NICI Transfer (NUWNICI)

As part of 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 handy 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.

In such an event, we highly recommend that you delete the NICI configuration on the old server.

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) 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/200 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.

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

The nicisdi.xlm module shipped as part of NICI 2.*x* or later does a better 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 up LDAP/etc.

NICI Backup and Restore

A backup and restore of the NICI configuration directory, along with its subdirectories and files, is sufficient for backing up NICI configuration on all platforms.

On versions of NICI before 2.*x* on NetWare platforms, some of the NICI configuration files are stored in the sys:_netware directory. A backup utility might or might not include this directory. We recommend an upgrade to NICI 2.0.1 for a more streamlined NICI configuration backup. NICI versions 2.*x* and later store NICI configuration files in the sys:\system\nici directory.

NICI Upgrade to version 2.*x* on UNIX Systems

A hybrid version (mixing features of NICI 1.2 and NICI 1.5) of NICI was shipped with the Tao version of eDirectory. 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 /var/nds1 and /var/nds2 directories, respectively. The runf2dc tool must be run on 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. On the contrary, encrypted data in eDirectory depends on keys stored in NICI configuration files. Such encrypted data (such as user private keys, certificates, secret store data, and NMAS store data) will not be 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).