

Novell[®]

NetWare[®] 6 and multiprocessing

Networking Services

www.novell.com

TECHNICAL WHITE PAPER

N



Novell[®]

table of contents

Novell NetWare 6
and multiprocessing

2	NOVELL NETWARE 6 AND MULTIPROCESSING UNIT
2	MULTIPROCESSING VOCABULARY
3	INTEL'S MULTIPROCESSOR SPECIFICATION (MPS) 1.4
3	MULTIPROCESSING TECHNOLOGIES
5	NETWARE 6 MP SUPPORT
9	PERFORMANCE AND SCALABILITY THROUGH MULTIPROCESSOR SUPPORT
11	SUMMARY OF MULTIPROCESSING ADVANTAGES IN NETWARE 6

Novell® NetWare® 6 and multiprocessiNg unit



Multiprocessing may be the left-out middle child of Novell® NetWare® 6's trio of performance advances. Stuck between the considerable improvements in NSS (Novell Storage System) and NCS (Novell Clustering Services™), multiprocessor support rarely gets mentioned on its own. Often an afterthought, the story becomes "New file services power improved clustering, and, oh yes, improved multiprocessor support" in NetWare 6.

This slight does a disservice to the impressive technical improvements displayed in NetWare 6 with considerably improved multiprocessing support. A totally separate technology advance from the file system and clustering, multiprocessing in NetWare 6 deserves more of a spotlight. After all, performance improvements in server throughput, application support power, and load balancing across a cluster all owe much to the technical foundation of multiprocessor support.

MULTIPROCESSING VOCABULARY

There are a few new terms to understand when discussing systems using multiple processors.

Most of these terms are clear and to the point, but the limited vocabulary set available can lead to confusion. Seemingly half the terms start with "multi" so misunderstandings are common.

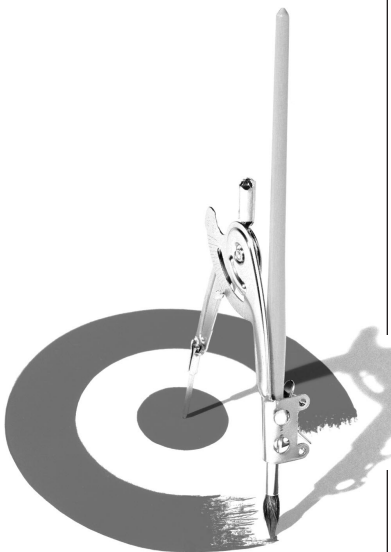
SMP (Symmetrical MultiProcessing)—An SMP system uses a single copy of the operating system to manage more than one equivalent processor. It is "tightly coupled" in that the processors all share the operating system as well as memory and I/O channels.

MPK (Multiprocessing Kernel)—The NetWare MPK replaces the standard operating system kernel for SMP servers. This replacement kernel loads

automatically when the operating system finds more than one processor when the hardware discovery process runs during installation.

Thread—The simplest explanation is that a thread is a unit of execution, but that leaves holes in the definition. More completely, a thread is the information (unit of execution) needed to serve one particular service request, whether from an application or a user. Threads and tasks are somewhat similar and confused by some, but they shouldn't be. A thread usually includes multiple tasks, and may allow itself to be interrupted between tasks.

Run queue—A data structure controlled by the kernel that contains threads ready to process.



Details in the rest of the paper use these definitions as building blocks to help explain more involved multiprocessor concepts and technologies. Unfortunately, not everyone follows the same exact definitions as everyone else. Where confusion exists between different vendors, we'll offer the explanation that works best with NetWare.

INTEL'S MULTIPROCESSOR SPECIFICATION (MPS) 1.4

Intel, the world's largest microprocessor developer, defines multiprocessing rules and conditions for their processors. Their most recent specification, version 1.4, outlines the rules Novell engineers followed developing the MPK (MultiProcessing Kernel) for NetWare 6.

Processors must be functionally identical—It may seem cost-effective to team a lower-cost and lower-powered processor as a member of a multiprocessing system to handle less demanding operations, but it doesn't work that way. Processors must be (functionally) identical, and you always see multiprocessor motherboards that support identical processors.

Processors must have equal status—Nothing in the hardware must allocate one processor to a higher or lower level than any other processor. The operating system makes those designations based on processor load, but not the processor or supporting hardware.

Processors must communicate with each other—Each processor in an SMP system must have full connections for communication to all other

processors in the system. Coordination of processor activities requires constant communications.

Processors must share the same I/O subsystem—SMP processors share everything equally, hence the "symmetrical" tag. Research into asymmetrical processing systems provides fascinating study, but doesn't apply to NetWare 6. Each processor must be functionally identical, including I/O access.

Processors must share the same memory space—The pool of server memory (which should be substantial in an SMP server) will be accessed by all processors. Again, functional equivalence.

Processors must use the same memory addresses—Coordination of processor functions relies heavily on shared memory to pass operations back and forth if necessary; therefore, all processors must use the same memory pool and the same memory addresses to function properly.

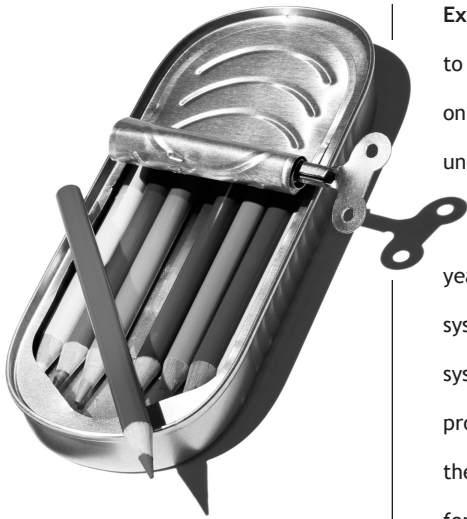
These aren't all the specifications from Intel, of course, but they show the broad outlines of what must be done to build an SMP system. Novell engineers work closely with Intel engineers, and have for years.

If you don't believe Novell engineers understand the lowdown details of microprocessors as well or better than anyone else in the computer business, your historical memory needs updating. Who was the first company to use Intel's 386 processor in 32-bit protected mode to access the expanded memory and address space? Novell.

MULTIPROCESSING TECHNOLOGIES

Buzzword alert: NetWare 6 is a multithreaded, multitasking, MP-enabled operating system.





Explanation: NetWare 6 uses multiple processors to improve performance. How that works depends on a variety of technologies working together under control of the MPK.

Novell engineers prepared for multiprocessing years ago by using threads in the operating system. While multiple threads in a uniprocessor system must obviously run through the single processor, multithreading techniques developed in the operating system are then ready and waiting for a multiprocessor system. Development time drops drastically, since much of the operating system already supports a key component of multiprocessor support.

Multithreading

As just mentioned, multithreading does not mean multiprocessing. NetWare has been multithreading for over a decade before multiple processors became generally available for use in a NetWare server.

A thread, or unit of execution, is not the code itself, it is a stream of control that executes instructions independently (from other threads). This stream of instructions consists of tasks in the thread.

The kernel controls the run queue, essentially a stack of threads ready to run. A multithreaded program is one where two or more threads can execute concurrently. Let's explain "concurrently" a little better.

People often call things "simultaneous" when they mean concurrent. Simultaneity is often thought to mean two or more of the same things

happening at the same time. A physicist will tell you two things doing the same thing at the same time are really the same thing, and that you really mean concurrent. Let's hope your physicist tells you this politely; many don't.

Concurrently means two different things (threads) running at the same time. How is this possible? After all, a single processor can only handle one instruction operation at a time, so how can it run two threads concurrently?

Modern processors switch between multiple threads so quickly it appears that two or more things are happening at once. Threads can be preempted, or suspended, while they wait on something. The speed of modern processors slice a second into so many millions of slices that multiple threads can run concurrently perfectly well.

Assume you launch a thread to find and display a file. Since the processor runs many times faster than memory and storage systems, the thread will go into suspension each time the file system can't find the desired data in memory and must wait for a disk I/O response. Since many clock cycles in the CPU will go by while waiting for a hard disk to read and respond, many other threads may be serviced while your file fetching thread waits for results.

Multithreaded programs do not support multiple processors automatically. Unless the developers specify the ways to use a multiprocessing system, a multithreaded program will never run on more than one processor.

Multitasking

This brings us to multitasking, but not full multiprocessing just yet. Multithreaded operating

systems that support multitasking can execute threads from different multithreaded programs concurrently on a single processor.

Multitasking makes it seem that multiple threads are executed concurrently, but they really aren't. The round-robin approach of the processor shifts from one thread to another every time a thread allows itself to be suspended; however, only one thread at one time can have an instruction being processed in a uniprocessor system. One CPU, one executing thread, period.

Fast processors make it appear multiple threads execute at exactly the same time, but they don't. When you get way down into the processor's time scale, millions of operations a second, you still see only a single thread active in the processor. One CPU, one executing thread; there's no way to "massage" that by clever marketing.

MP-enabled

Once developers have written their programs to be multithreaded and multitasking, they still must take the next step and make them MP-enabled. Support for multiprocessor operations must be built into the operating system and applications involved. Without explicit MP-enabled instructions, applications will remain bound by a single processor.

Ah, but when a multiprocessor system runs an MP-enabled, multithreaded, multitasking application, miracles do happen. One thread can be executing at exactly the same time on each processor in the system. A single application, written properly, suddenly supports multiple thread execution on an SMP system. The appearance of

multitasking applications gives way to the reality of true multitasking applications.

NETWARE 6 MP SUPPORT

Multiprocessor support in NetWare is not brand new with NetWare 6, and, in fact, the majority of MPK features in NetWare 6 were available in NetWare 5 and some date back to NetWare 4. Most improvements in NetWare 6 multiprocessor support come outside the MPK because many NetWare functions are newly multiprocessor enabled.

Critical NetWare 6 MP Components

Plainly, multiprocessor support includes quite a bit more than a few kernel modules to recognize and use more than one processor in the server. Many software components inside NetWare and from application developers must work together to increase server performance through the use of multiple processors.

Scheduler

The traffic cop inside the MPK, the Scheduler uses the MPS 1.4 Platform Support Module (MPS14.PSM) to determine the number of processors in a server during installation. By watching processor activity, the Scheduler decides where to send new threads requesting execution.

Programs written that do not explicitly use multiple processors but are deemed MP safe will be spread out among available processors. Depending on how many threads they use and whether the developer used serialization techniques, any processor can handle the threads.

Every program in a single processor system runs on Processor 0 because the zero processor is the only one there. Programs not MP safe automatically run on Processor 0.

Some developers explicitly request certain processors in the program itself. This falls outside the realm of smart programming techniques, but it is done now and then. Typically, these programs request Processor 0 even when MP-enabled. No matter how much operating system vendors strongly suggest this technique can cause problems, developers still do it occasionally so NetWare supports it.

The Scheduler, after dealing with exceptions, takes each new thread as it appears and allocates that thread to the first idle processor. Processor 2 not doing anything? The thread goes there.

Once a thread runs on a particular processor, the Scheduler tries hard to keep that thread on that same processor for reasons of efficiency. There are two primary exception states where the Scheduler moves a thread:

A thread not MP-enabled gets moved to Processor 0.

The load-balancing gets far out of balance.

Two other rare situations also can cause the Scheduler to move a thread. If a processor is stopped by a management command, those threads must obviously move to other processors. Threads which specify processors by number will also be moved. Both of these situations are rare.

When a MP-enabled program runs, here's how NetWare 6 runs the program:

Scheduler checks to see which (if any) processors are idle

Scheduler sends the waiting thread to the lowest-numbered idle processor

Scheduler repeats the process with each waiting thread

Returning thread requests stay on the processor where they started if at all possible

"Processor affinity" is the term for the technique to keep threads on the same processor whenever possible. Unless one of the two exceptions occurs, the Scheduler follows the processor affinity rule and leaves threads alone to execute on their particular processor.

Funneling

The fancy (er, official) name for moving non-SMP programs to Processor 0 is funneling. If a thread from an older application gets assigned to Processor 1 or above by some chance, such as not identifying itself as non-SMP and not appearing to be a legacy application, the funneling process within the Scheduler takes over and moves the thread to Processor 0.

Threads which get funneled do so because:

The thread is in an MP state

The thread is executing MP enabled code

The thread calls an MP-unsafe procedure

When the above conditions occur, the Scheduler will funnel the thread to Processor 0. Once the thread finishes the MP-unsafe procedure, Scheduler will return the thread to the original processor.

Run Queues

A run queue, a data structure inside the operating system kernel, holds threads ready for processing. Uniprocessor systems have a single run queue, since they have a single processor.

Multiprocessor systems demand a new way to organize threads in a state of readiness, and two options lead the way: global run or per-processor run queues.

A global run queue provides a single run queue that distributes ready threads across all processors. Since the global run queue always has threads ready to process, no processor stays idle for long. Unfortunately, as the number of processors increase, the global run queue itself can become a system bottleneck.

Per-processor run queues provide an advantage in maximizing throughput per processor due to using thread's processor affinity. Threads almost always run on the same processor they ran previously, keeping high speed cache information for the thread close at hand. No single queue blocks access to all processors, eliminating the bottleneck possibility of a global run queue.

On the other hand, per-processor run queues must have some overhead built in to maintain load balancing. A single processor's run queue can become heavily loaded, but the per-processor run queue can't itself compare its load with that of other processors. An outside mechanism (like Novell's Scheduler) must help the balancing remain distributed.

NetWare's kernel uses the per-processor run queue, one reason NetWare multiprocessor systems scale so well. Each processor picks up waiting

threads from its own processor run queue, allowing each added processor to provide more total system horsepower. Yet some outside mechanism must help load balancing to maintain the performance increase with each processor.

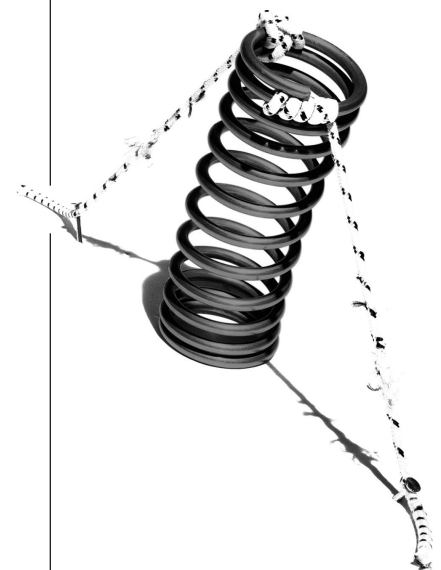
Load Balancing

NetWare uses a sophisticated load balancing algorithm to maintain relatively equal performance across multiple processors. The two critical components of any balancing scheme are the ability to distribute processing load quickly, yet the stability to not overreact to small load imbalances.

The Scheduler in NetWare uses a threshold to maintain system load balancing stability. In fact, two thresholds feed information to the Scheduler: high trigger load and low trigger load. This option provides the optimum balance between processor inactivity and excessive thread migration using two system measurements.

Periodically, the NetWare Scheduler calculates the system-wide load and the mean load (mid-point between the highest and lowest loads). This calculation result gets applied to each individual processor to determine if that processor is underloaded or overloaded. The threshold margin maintain system productivity by allowing some leeway before thread migration.

Note: although the threshold can be changed in the NetWare Remote Manager, Novell engineers strongly recommend against making any changes. If you must make changes, note the optimum value so you can reset the system when you realize Novell engineers give good advice which should be heeded.



Never Enough Cache

Memory vendors make new memory chips faster all the time, but no external memory chips can process data as fast as memory built into the processor itself: onboard cache memory. Running at the same speed as the CPU, and with no delays for off-chip I/O, onboard cache truly blazes new speed records.

Why does processor affinity receive so much attention by NetWare engineers? To utilize onboard cache, of course.

Cache misses occur when the Scheduler forces a thread to migrate from one processor to another. This forces a cache flush, where the data needed by the migrated thread must be written out of the first processor (flushed) into system RAM. The new processor executing the thread then reads system RAM for the thread data. As you can guess, performance engineers groan when calculating the drop in thread performance speed with every cache miss.

There are three types of cache:

L1 (Level 1): internal to the processor chip core and just as fast as the processor itself

L2 (Level 2): eternal to the processor chip core, yet often inside the processor chip housing (or cartridge), this cache is almost as fast as the processor.

L3 (Level 3): Typically external to the processor chip and chip housing (or cartridge).

Processors with large L1 and L2 caches cost quite a bit more money than processors with smaller onboard caches. Where a Pentium* III chip may have an L2 cache of 256KB, the same speed Pentium III XEON processor may have 1MB of

onboard cache. Now you understand why servers with XEON or other high-cache processors cost so much more, and why a cheap server won't perform as well under load as one of the servers with larger processor caches.

Data in processor cache must always be written out to system RAM sooner or later, of course, so other processors can take advantage of the data if necessary, and to keep the system current and data in balance. NetWare 6 uses a lazy-write algorithm for normal cache data copies to RAM. When the cache management circuits realize the cache has no more room for more data, the system writes the information out to RAM. This puts all the modified data out where all other processors can use the data, but on the processor's terms, not when forced by a cache miss.

Improvements Since NetWare 5.1 MP Support

There are improvements in the MPK between NetWare 5.1 and NetWare 6, but no tremendous leap of innovation. The biggest improvements in multiprocessor support came between NetWare 4.11 and NetWare 5, when the entire MPK upgraded considerably for the newer, more powerful processors available and new motherboards to support them.

NetWare 6 MP Enabled Components

Since NetWare 5, the multiprocessor engineers at Novell have been busy upgrading critical server functions to better utilized multiprocessor servers. The list of MP-enabled components may surprise you. There are so many we need to group them:



Specialized Servers and Critical Components

NDS: eDirectory™

Novell JVM (Java® Virtual Machine)

Search engine

Web engine

Servlet interface in NetWare Enterprise
Server**Protocol Stacks**

NetWare Core Protocol™ (of course)

TCP/IP (complete IP stack family)

HTTP

WebDAV (Web-based Distributed Authoring
and Versioning)

LDAP (Lightweight Directory Access Protocol)

SLP2 (Service Locator Protocol)

Gigabit Ethernet, 100 Megabit Ethernet,
10 Megabit Ethernet

Token Ring 16

NNTP (NetWare News Server running
Network News Transport Protocol)**Storage and Data Transfer**

NSS (Novell Storage Services™)

DFS (Distributed File Services)

Fibre Channel disk support

Transport service request dispatcher

Protocol service request dispatcher

Security Features

Authentication

NICI (Novell International Cryptographic
Infrastructure)

GUI Audit (new ConsoleOne® snap-in module)

Novell MP-Enabled Products

BorderManager®

GroupWise®

ZENworks® for Desktops

ZENworks for Servers

No others

Different customers will utilize different MP-enabled applications and utilities, but every customer will benefit by running NetWare on a multiprocessor server. Throughput, one of the bottlenecks for servers today, gains a considerable increase with TCP/IP becoming MP-enabled. Storage services always need more speed, at least according to users.

With NetWare 5, SMP systems provided performance improvements for specific applications. NetWare 6 increases performance many ways on multiprocessor servers, speeding the entire user experience through improved MP-enabled functions within NetWare.

**PERFORMANCE AND SCALABILITY
THROUGH MULTIPROCESSOR SUPPORT**

The list of MP-enabled components within NetWare show the dedication Novell applied to improving user experience with NetWare 6 on multiprocessor servers. Notice how Novell's other products take advantage of multiprocessor servers as well, greatly improving performance of those products.

Novell recognizes network manager's workload increases on a daily basis. One way to improve application performance for less money and less management time comes from multiprocessing certain applications. A server with four processors, while requiring capital investment, always costs

less than four servers with a single comparable processors. An application needing more horsepower will gain that horsepower from a multiprocessor server while keeping investment down and management time under control. After all, even with NDS, it's always easier to manage, control, and physically secure a single server than four individual servers.

Automatic support for 1-32 server processors

NetWare 6 SMP support automatically recognizes and supports multiple processors in the hardware server. The NetWare kernel, as we mentioned earlier, efficiently utilizes multiple processors in a server without requiring any installation configuration changes. Aside from a few mentions of multiple processors, and some extra information in the NetWare management utilities, there is no management time difference between a server with one or many processors.

Most hardware vendors push their four processor servers as their top end. Earlier multiprocessor operating systems tended to waste the extra processing power when the processor count headed north of four processors, and hardware vendors took that hint. Now that more powerful and more modern operating systems provide better multiprocessor support (like NetWare 6), vendors are starting to push their processor count up to at least eight.

Processing Power Improvement Scale

The complicated operating system dance to coordinate multiple processors requires

considerable engineering expertise. Overhead associated with multiple processors has, in the past, negated the horsepower gains provided by more processors. NetWare 6 improves on that history considerably, giving customers more bang for their processing buck than ever before.

Novell recommends the following guidelines for processing power increase with each new processor in a single server:

Two processors = 1.8 times more power than one processor

Four processors = 3.5 time more power than one processor

Six processors = 5.2 times more power than one processor

Eight processors = 6.1 times more power than one processor

Plenty of variables come into play here. Processors with large onboard cache memory perform better than those with less cache memory. Applications developed with an appreciation of multiprocessor environments provide more performance on such systems. Total system memory (always) impacts system performance; the more memory, the more performance.

Beware quick and easy calculations for increased server performance by adding extra processors. If a data-intensive application runs out of horsepower supporting 100 users, adding a second processor will not automatically support 200 users, or halve the response time for the first 100 users. Disk throughput speeds and system I/O

bandwidth may have more to do with database performance than just the number of processors. But when a database can't easily be distributed across multiple servers, adding more processors (and attendant memory and I/O bandwidth) will certainly boost performance.

As the car ads used to say, "your mileage may vary" depending on the multiprocessor application and how well the developers did their jobs. Multiprocessing doesn't help applications with large amounts of disk or network I/O, because that's so much slower than the processors. For instance, basic file and print operations don't really benefit from multiprocessing. Yet applications which require plenty of processor power and are intelligently developed for multiprocessing see tremendous performance gains with NetWare 6 running on a multiprocessor server.

Intel's Pentium IV platform will improve multiprocessing support as new servers roll out with the new processor. Not only will the processor itself speed up, but Pentium IV servers will include a faster memory bus and more local cache. Faster processors with faster memory will increase multiprocessor performance even more than today.

SUMMARY OF MULTIPROCESSING ADVANTAGES IN NETWARE 6

We ask much more from our servers than in the past. NetWare 6 and NDS eDirectory manages multiple servers easier and more securely than ever before, but there are often reasons to beef up one server rather than spread applications across multiple servers. When you need more server horsepower, MP-enabled NetWare 6 performs.

The NetWare 6 kernel automatically discovers and configures the operating system for multiple processors, meaning easier installation and ongoing management. Greatly improved support among all the component parts of NetWare 6 means multiprocessing performance enhancements cover the gamut of network services. Increased developer support for Novell's multiprocessor SDK (Software Development Kit) means more applications can now take full advantage of NetWare 6 running on multiprocessor hardware.

Sports cars and servers all need plenty of horsepower. NetWare 6 on a multiprocessor server provides plenty of performance while maintaining the great handling for an enthusiastic driving, er, serving experience. Take a multiprocessor server powered by NetWare 6 out for a test drive today.



NOVELL OFFICES WORLDWIDE

Novell Corporate Headquarters
1800 South Novell Place
Provo, Utah 84606
USA
Tel: (801) 861 7000
Toll-free: (800) 453 1267

AMERICAS REGION

Novell Canada
3100 Steeles Avenue East
Suite 500
Markham, Ontario L3R 8T3
Canada
Tel: (905) 940 2670
Fax: (905) 940 2688

Novell de Argentina S.A.
Ing. Butty 240, piso 6
Edificio Laminar
(1001) Capital Federal
Argentina
Tel: (54) 11 4 510 2626
Fax: (54) 11 4 510 2656

**Novell do Brasil
Software Limitada**
Avenida das Nações
Unidas, 12.995
8º Andar
04578-000 São Paulo - SP
Brasil
Tel: (55) 11 5505 4040
Fax: (55) 11 5505 4041

Novell Chile
Edificio Nuevo Centro
Providencia
Av. 11 de Septiembre 1881,
Of. 615
Providencia, Santiago
Chile
Tel: (56) 2 3769211
Fax: (56) 2 3769776

Novell de Colombia
Teleport Business Park
Calle 114 No. 9 - 45
Torre B - Of. 709
Santafé de Bogotá
Colombia
Tel: (57) 1 629-2969
Fax: (57) 1 629-3509

Novell de México
Blvd. Manuel Avila
Camacho #138-1
Col. Lomas de Chapultepec
Mexico D.F., C.P. 11000
Tel: (52) 5 284 2700
Fax: (52) 5 284 2799

Novell Central America
One East Broward Blvd.
Suite 700
Ft. Lauderdale, FL 33301
USA
Tel: (954) 713 2869
Fax: (954) 356 0409

Novell de Panamá
Calle 53 Marbella,
World Trade Center
Piso 2, Oficina 10
Ciudad de Panamá
Panama
Tel: (507) 206-8714
Fax: (507) 206-8777

Novell del Perú
Pasaje Martir Olaya 129,
Of. 1701, Miraflores
Lima 18
Peru
Tel: (511) 214 1340
Fax: (511) 214 1087

EUROPE, MIDDLE EAST, AFRICA REGION

Novell Austria
Heiligenstädter Lände 27c
1190 Vienna
Austria
Tel: (+43) 1 367 74 44
Fax: (+43) 1 367 74 44 20

Novell Belgium NV
Koningin Astridplein 5,
(3rd floor)
2018 Antwerpen
Belgium
Tel: (32) 3 206 1793
Fax: (32) 3 206 1799

Novell Praha s.r.o.
Praha City Center
Klimentska 46
110 02 Praha 1,
Czech Republic
Tel: (420) 2 2185 66 11
Fax: (420) 2 2185 66 22

Novell Danmark a/s
Slotsmarken 12
DK 2970 Hørsholm
Danmark
Tel: (+45) 45 16 00 20
Fax: (+45) 45 16 00 40

Novell Finland
Lars Sonckin kaari 14
02600 Espoo
Finland
Tel: (358) 9 502 951
Fax: (358) 9 502 95 300

Novell France
Tour Framatome
1, place de la Coupole
92084 Paris La Défense cedex
France
Tel: (33) 1 47 96 60 05
Fax: (33) 1 47 78 94 72

Novell Germany
Monschauer Strasse 12
40549 Düsseldorf
Germany
Tel: (49) 211 5631 0
Fax: (49) 211 5631 250

Novell Hungary
East-West Business Center
Rákóczi út 1-3
1088 Budapest
Hungary
Tel: (36) 1 235 7656
Fax: (36) 1 266 6360

Novell Israel
Ackerstein Building
Medinat Hayehudim St 103
Herzliya 46776
Israel
Tel: (972) 99 51 44 55
Fax: (972) 99 51 44 66

Novell Italia
Piazza Don Mapelli 75
20099 Sesto San Giovanni
Milano
Italy
Tel: (39) 02 26295.1
Fax: (39) 02 26295 800

Novell Nederland
Barbizonlaan 25
2908 MB Capelle a/d IJssel
PO Box 85024
3009 MA Rotterdam
The Netherlands
Tel: (31) 10 286 44 44
Fax: (31) 10 286 40 10

Novell Norge AS
Østensjøveien 34
P.O. Box 6555 Etterstad
0606 Oslo
Norway
Tel: +47 23 37 77 70
Fax: +47 23 37 77 71

Novell Polska
ul. Wspólna 47/49
00-684 Warszawa
Poland
Tel: (48) 22 537 5000
Fax: (48) 22 537 5099

Novell Portugal
Centro Empresarial Torres
de Lisboa
Rua Tomás da Fonseca,
Torre G
1600-209 Lisboa
Portugal
Tel: (351) 21 723 06 30
Fax: (351) 21 722 35 33

Novell Russia and CIS
Suite 524, Business Center
2, Berezhkovskaya
Naberezhnaya
121059 Moscow, Russia
Tel.: +7 095 941 8075/8073
Fax: +7 095 941 8066

Novell South Africa
Morning Wedge Office Park
Novell House
255 Rivonia Road
Morningside
Sandton
PO Box 1840
Rivonia 2128
Republic of South Africa
Tel: (27) 11 322 8300
Fax: (27) 11 322 8341

Novell Spain, S.A. (Madrid)
Paseo de la Castellana, 95
Planta 27
Edificio Torre
Europa
28046 Madrid
Spain
Tel: (34) 91 555 65 67
Fax: (34) 91 555 29 15

Novell Sweden
Kronborgsgränd 1
164 87 Kista
Sweden
Tel: +46 (8) 477 41 00
Toll-free: +46 (20) 35 30 30
Fax: +46 (8) 477 41 01

Novell (Schweiz) AG
Leutschenbachstrasse 41
8050 Zürich
Switzerland
Tel: (41) 1 308 47 47
Fax: (41) 1 302 04 01

Novell United Kingdom Ltd.
Novell House
1 Arlington Square
Downshire Way Bracknell
Berkshire RG12 1WA
United Kingdom
Tel: +44 (0)1344 724000
Fax: +44 (0)1344 724001

ASIA PACIFIC REGION

Novell Pty Ltd
Level 18, 201 Miller Street
North Sydney NSW 2060
Australia
Tel: +61 2 9925 3000
Fax: +61 2 9922 2113

Novell New Zealand Limited
L12, 44 - 52 Wellesley Street
Auckland 1
New Zealand
Tel: +64 9 308 1400
Fax: +64 9 308 1409

Novell China
Floor 11 Canway Building
No. 66 Nan Li Shi Road
Beijing 100045, China
Tel: (86) 10 68028855
Fax: (86) 10 68028720

Novell Hong Kong
Room 4601-5
China Resources Building
26 Harbour Road
Wanchai
Hong Kong, China
Tel: (852) 2 588 5288
Fax: (852) 2 827 6555

**Onward Novell
Software (I) Ltd.**
62 MIDC, 13th Street
Andheri (East)
Mumbai 400 093
India
Tel: +91 (022) 8342244
Fax: +91 (022) 8342223

Novell Japan Ltd.
Toei Mishuku Bldg.
1-13-1 Mishuku
Setagaya-Ku
Tokyo 154-8561
Japan
Tel: (81) 3 5481 1294
Fax: (81) 3 5481 1934

**Novell Corporation (Malaysia)
Sdn Bhd**
Unit 501, Level 5, Uptown 1
1 Jalan SS21/58
Damansara Uptown
47400 Petaling Jaya
Selangor Darul Ehsan
Malaysia
Tel: (60) 3 7722 6100
Fax: (60) 3 7722 6155

Novell Philippines, Inc.
28th Floor, Tower 2,
The Enterprise Center
6766 Ayala Avenue,
Paseo de Roxas
1226 Makati City, Philippines
Tel: (632) 849 3878
Fax: (632) 849 3861

Novell Singapore Pte Ltd
8 Temasek Boulevard
#32-01/02 Suntec Tower Three
Singapore 038988
Tel: (65) 395 6888
Fax: (65) 395 6777

**Novell Software
(Thailand) Ltd.**
16th Fl., TISCO Tower
48 North Sathorn Rd.
Silom, Bangrak,
Bangkok 10500
Thailand
Tel: (662) 638 0310
Fax: (662) 638-0311

Novell Taiwan
The MORE Int'l Business Centre
23F., No. 105, Section 2
Tun Hwa South Road
Taipei 106
TAIWAN R.O.C.
Tel: (886) 2 2784 1613
Fax: (886) 2 2702 5050

© 2001 Novell, Inc. All rights reserved.
Novell, NetWare, BorderManager,
ConsoleOne, GroupWise, NDS and
ZENworks are registered trademarks,
and eDirectory, NetWare Core Protocol,
Novell Cluster Services and Novell
Storage Services are trademarks of
Novell, Inc. in the United States and
other countries.

*Pentium is a registered trademark of
Intel Corporation. Java is a registered
trademark of Sun Microsystems, Inc.
All other third-party trademarks are
the property of their respective owners.

Novell Product Training and Support Services

For more information about
Novell's worldwide product
training, certification programs,
consulting and technical support
services, please visit:

www.novell.com/services

For More Information

Contact your local
Novell Authorized Reseller,
or visit the Novell Web site at:
[www.novell.com/products/
netware](http://www.novell.com/products/netware)

You may also call Novell at:

1 888 321 4272 US/Canada
1 801 861 4272 Worldwide
1 801 861 8473 Facsimile

Novell, Inc.
1800 South Novell Place
Provo, Utah 84606 USA

www.novell.com

Novell