

ZENworks & Y2K @ Novell

A Beigepaper

by Grettir Asmundarson (grettir@neticus.com)
Last Revised: December 16, 1999

A Service Of Novell's Information Services & Technology Technical Architecture Group

© 1999 by Grettir Asmundarson. All rights reserved. Any part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without any permission whatsoever...as long as you give me credit. All brands and product names mentioned are trademarks or registered trademarks of their respective companies. Patent pending. All sales final. Void where prohibited by law. No purchase necessary. Low in saturated fat. Do not read this document if the printed seal is broken or missing. Do not drive a vehicle or operate heavy machinery while reading this document. Call a physician if swelling persists. The author's wanton use of the "@" symbol does not necessarily reflect the sensibilities of the author himself, who is personally opposed to the gratuitous use of internet-related typographic symbols. :-)

About The “@ Novell” Series

Most documentation starts as hastily scrawled notes from sleep-deprived developers who weren't necessarily hired for their keen communication skills. Those notes are then fleshed out by recently graduated English majors who have spent their last four years immersed in works of fiction. The results are then passed on to the marketing department whose job it is to make sure that no word or phrase, even if it's true, will reflect unfavorably on the product (“I don't think that the word ‘Basic’ properly communicates the exciting nature of the product. Why don't we call it ‘Visual Zesty!?!’”). It is then beset by lawyers who finish the job by making sure that they haven't explicitly promised that the product will actually do anything.

By the time the documentation gets into your hands, it has been so sanitized for your protection and generalized beyond recognition that you usually have to go out and buy a 3rd-party manual (that was, more likely than not, written by the same non-technical technical writer who wrote the original documentation) in a vain attempt to get an unbiased, unexpurgated, and/or unfiltered view of just how you're really supposed to use the stuff.

That's where the “@ Novell” series comes in. Rather than the vague, generalized, and wholly fictional examples found in most documentation, we're going to tell you exactly how we use our own products to run our own company. After all, we are not a small, tidy computing environment suitable for documentation. We are a big, sprawling, untidy environment made up of over 500 production servers and 20,000 workstations in 130 locations throughout the world. In other words, we're probably an awful lot like you.

And it's not that we're necessarily any smarter than you are, we just have a distinct advantage. By the time you get your hands on one of our released products, we've already been using it to run our business for quite some time. For instance, a month before NetWare 5 shipped, well over half of our 500 production file servers had already been upgraded to NetWare 5. (Keep in mind that these were production servers. These were not test servers that we had safely tucked away in antiseptic labs. These were real-world servers in a real-world environment solving real-world problems.) And two months before NetWare 5 shipped, we'd already converted one of our buildings to IP Only. That means that we've probably gained some insights into implementing our products in a big, sprawling, untidy environment, and this paper is an attempt to share those big, sprawling, untidy insights with our customers.

But keep in mind that this document may be a little rough. It wasn't conceived by a committee, written by a committee, or approved by a committee, so it hasn't been edited, re-edited, tidied up, sanitized, and whitewashed. Don't think of this as an official whitepaper. It's more like a beigepaper.

How *Not* To Get On The 10 O'Clock News

During the Melissa virus brouhaha, a Salt Lake City television station contacted Novell in hopes of interviewing someone about the destruction and mayhem being caused by the outbreak. The

television station had heard reports of Fortune 500 companies whose mail systems had been brought to their knees and I think they were salivating about the idea of a local example of destruction and mayhem that would:

- a. Horrify their viewers with another example of technology run amok.
- b. Bring that horror closer to home.
- c. Increase ratings with tantalizing tag lines like: “Local company’s e-mail system constipated by rogue virus! Details at 10!”

Novell’s PR folks contacted me and asked me how we were holding up under the onslaught.¹ I informed them:

- a. Since GroupWise wasn’t vulnerable to the replication problems that plagued Microsoft Outlook, we weren’t subject to the congestion that had been such a problem for the other companies.
- b. We had already used ZENworks to roll out a new version of our anti-virus client to those users who didn’t already have it and a new virus pattern file to those who already did.
- c. Therefore there was no destruction or mayhem to report.

Being a PR representative, our PR representative tried to put a positive spin on our non-story in hopes that the station would highlight the role of ZENworks in our state of non-emergency. But the reporter was appalled at our utter lack of destruction and mayhem and quickly lost interest in us.

Y2K: Another Example Of Our Utter Lack Of Destruction & Mayhem

When this beigepaper was first proposed, it was suggested that we play up “The Y2K Story,” meaning our use of ZENworks and Greenwich Mean Time’s Check 2000 application to test and ensure Y2K compliance on all of our desktops. The idea was that “The Y2K Story” was much sexier than just talking about ZENworks generally, but if you ask me it’s just another example of our utter lack of destruction and mayhem.

One of the hardest things about dealing with Y2K issues is just figuring out what’s out there. Even in the best-managed environments, I’m amazed at the amount of crap (both hardware and software) that can filter in over time. ZENworks and GMT’s Check 2000 can make the job of establishing what’s out there much easier. Both applications have the ability to inventory your workstations for BIOS, OS, and application revisions, but Check 2000 goes one step further,

¹ Our PR representative informed me that I wasn’t allowed to talk with the reporter from the local television station directly because I hadn’t gone through “Spokesperson Training.” I think they were afraid that I would belch or scratch my armpits during the interview. The next morning I might have turned on CNBC to find Maria Bartiromo announcing, “Scrawny engineer’s unkempt appearance sends Novell stock plummeting in after-hours trading.”

providing you with recommended Y2K solutions for over 5,000 applications. And once you know what you have and what you need to do to fix any problems, rolling out those solutions is a piece of cake with ZENworks.

Here @ Novell, we used Check 2000 to do the preliminary inventory in order to formulate our internal Y2K initiative. Once we had established what patches and upgrades were required, we were able to distribute them quite easily using the ZENworks Application Launcher.

That's all there was to it. Boring, huh? No destruction, no mayhem. But that's the way we like it. And with "The Y2K Story" out of the way, we can now get into the nuts and bolts of the product.

Installation

Like most other Novell products, the installation of ZENworks consists of extending the NDS schema and then installing the product itself. But rather than installing the product on every server in the tree, we install it on a single server and then ZIP up the ZENworks components in the SYS:\PUBLIC\ directory (and its sub-directories) into a single archive. Then we can manually blast those files to the other servers or any new servers that we install into the tree.

One of the most important aspects of ZENworks has nothing to do with the server, however. It's the client piece. The thing you need to keep in mind is that there are two different versions of the Novell clients: the regular Novell clients and the ZENworks clients. To many people, they appear to be the same thing and, indeed, they are almost identical. But the ZENworks version of the client contains a few key components that are not part of the regular Novell client.

Now, if you're like me (and I hope you're not), you might be asking yourself, "Why in the world are there two different versions of the Novell clients? Surely, that just complicates things, doesn't it?" And, if you're like me (and I hope you're not), you'd be wrong. It simplifies things.

If new ZENworks-specific pieces of the client were being pushed out with every new Service Pack, it could break things very easily. For instance, the ZENworks client for Windows 95/98 uses a file called WM95INV.DLL to publish inventory information to a database. In ZENworks 1.1, the client published its data to NDS. But in ZENworks 2.0, the client publishes its data to a Sybase database.

Having users upgrade the ZENworks-specific components of their clients willy-nilly can create a bit of chaos if they end up trying to use the ZENworks 1.1 client to write to a ZENworks 2.0 database. (Or vice-versa.) It just isn't going to work very well.

To make things a little easier, these components have even been separated from the client itself. While the enhanced inventory capabilities of ZENworks 2.0 may be very attractive to people doing last-minute Y2K work, if you've already got 10,000 people running the ZENworks 1.1 client, the thought of rolling out 10,000 copies of the ZENworks 2.0 client this close to Y2K may leave you clammy. (As well it should.) But ZENworks 2.0 ships with an AOT that will roll out

only those few pieces of the new client that are necessary to use the new ZENworks 2.0 inventory feature.

The ZENworks Application Launcher

The Gartner Group recently announced that it costs a company \$43,627.43 every time a technician has to visit a desktop.² So, to keep the accountants happy, we try to keep desktop visits to a minimum. For my money (all \$43,627.43 of it), one of the best ways to accomplish this is by using the ZENworks Application Launcher.

Here @ Novell, we use it to deploy nearly every piece of software to the desktops. We use it to roll out Windows 95/98 updates and Windows NT service packs. We use it to automatically install and configure Web browsers.³ We use it to push out software patches and virus pattern file updates. You name it.

One of the nicest aspects of ZENworks 2.0 is the ability to control access to software applications on a workstation level, not just by user. Limiting access to specific users was great, but what would sometimes happen is that you would assign an especially sexy application to an individual user. Then that user would suddenly become the most popular person in the department. They would be invited to one office after another, making new friends as they went.

(That sort of thing would never happen here, of course...)

Here are the User-level defaults we use:

Display Icon On Desktop	No
Enable Time Refresh	Yes
Read Groups for Applications	No
Set Application Inheritance Level	1
Set Refresh Frequency	14400
Use As Top Of Configuration Tree	✓

We set Application Inheritance Level to "1" in order to prevent users from walking the tree looking for applications during the launch of the Application Launcher. Without it, you can see significant performance degradation, especially if you are walking the tree across very slow links.

² O.K., I made up that \$43,627.43 figure. But it's some outrageous amount that no one quite believes, but can't really dispute.

³ ...and/or integrated components of the operating system that can't be removed for fear of hurting consumers.

We also have the Application Launcher configured to refresh every 14400 seconds (4 hours). That way if we need to push out a new piece of software we'll have the opportunity to touch every user's desktop at least once during their workday.⁴

Another nice aspect of the ZENworks 2.0 Application Launcher that we are especially fond of is Criteria-based Distribution. This allows you to specify what software and can't be installed based on specific criteria.

For example, (and I'll make my bosses happy by using a Y2K example) if you are running Microsoft Office 97, you need to deploy Service Release 2 in order to gain some semblance of Y2K compliance. The trouble is that Service Release 2 requires Service Release 1. To minimize the confusion, you can have the Application Launcher check what version of Office 97 is currently installed and only display those options that are valid for each set of criteria. For instance:

Detected	Option Displayed
Office 97	Service Release 1 (SR-1) Install
Office 97 SR-1	Service Release 2 (SR-2) Install

There are all sorts of different uses for this nifty feature. If you're switching anti-virus programs, you really shouldn't install the new one until you've uninstalled the old one. So, you could hide the application object that installs the new anti-virus program until they've run the application object that uninstalls the old one. Or you can automatically check a file date or version and, if it doesn't meet the established criterion, force out a virus pattern file update if a user doesn't have the latest installed. It's really quite handy.

Creating snAppShots

When creating application snAppShots, it's important to remember that less is more. Start with systems that have a bare minimum of software installed. Our snAppShot systems have only the base OS and the Novell client installed.

We then create a snAppShot with the following options:

Folder and File Entries	Copy if newer
INI Entries	Create if does not exist
Registry Entries	Create if does not exist

These settings assure that your snAppShot performs consistently and doesn't overwrite newer files with older ones. It also prevents unnecessary writes to the INI files and the registry.

⁴ ...unless, of course, they're in management. What with the Off-Site Management Retreat (read "11:00 a.m. tee time") and Team-Building Activity (read "2:30 p.m. *Phantom Menace* screening"), you might miss that 10-minute window when they are actually in their office.

After the snAppShot has been created, we go back in and delete any extraneous rubbish that might have been included. This might include DHCP settings that were assigned/changed during a reboot, Workstation Manager log files, and temporary files that might have been left lying around.

Importing Workstations

We import workstations on a daily basis either manually using NWAdmin or automated via the WSImport utility. Workstations are imported using the following naming conventions:

USERNAME-WORKSTATIONNAME-OS

For instance:

MDELL-IBMTHINKPAD600E-WIN98

This pretty much assures a unique name for every workstation in the tree, even if a user has multiple workstations.

We also purge workstations from the tree that are 21 days old (or older) to minimize the amount of workstation clutter that can accumulate over time.

A Side Note About Container Packages

We put container packages in every container of our tree rather than just placing them at the root. Why? Again, we are trying to minimize tree walking. Each user can get all of the information that they need from their local replica, to which they already have a default connection.

And each container package is configured to minimize tree walking in the following ways:

1. We set the search policy to PARTITION. This allows the workstation to access any information they need since they are already attached to their local partition for login. No tree walking, no additional connections necessary.
2. The search level needs to stay at the default of "0." That keeps them in their current context for their policies.
3. On the "Search Order" tab we make sure that the order is first set to "Object" and then "Container." This allows policies to first be effective by direct association and then by inheritance from their container.

Remote Control

The enhanced remote control capabilities of ZENworks 2.0 are also something that we are starting to use in a more meaningful way. We are using it mainly as a tool for our IS&T technicians. After all, if it costs \$127,346.92 every time a technician has to visit a desktop, we can avoid quite a few of those desktop visits by using remote control to resolve the trivial issues and saving the desktop visits for the important stuff.

We set the User Package to require a password for all remote services. We then set the Workstation Package to *not* require a password. This allows the Help Desk to assist people who aren't able to log in, since the Remote Agent can now run as a service under Windows 9x/NT. But if a user is already logged in, then the User Package policy will require that they grant permission to the Help Desk before they can do anything. This gives us a great amount of flexibility in handling problems while assuring the users a certain amount of privacy and control.

Inventory

Now that ZENworks 2.0 allows us to inventory software as well as hardware, we're using it to verify the results of GMT's Check 2000, and we'll be running follow-up inventories on a regular basis to keep an eye on things as we approach Y2K.

In our larger offices (50+ users), we run the Scanner, Gatherer, and Storer from a local server and publish to a local database server. But in our smaller sites, we run the scanner locally, but the Gatherer/Storer and the inventory database are centralized in a regional office.

Solving Real-World Problems In Non-Obvious Ways

Here @ Novell, our use of policies has not been very extensive. Policies are very useful in locking down workstations or handling roaming users. But since we have a fairly open computing environment here and roaming isn't a real issue, there aren't a lot of instances where policies provide real benefits.

There is one instance, however, where we found policies to be very useful. We had a few database front-ends that were experiencing some performance degradation unless the user had a four-processor workstation, 1GB of RAM, and a direct gigabit connection to the database server. At least that's what I think the developers must have had in mind when they developed the product, because performance was pretty pathetic on anything less.

People trying to access these databases across a less-than-stellar WAN link (let alone a modem connection) were finding that they could start a query, go have lunch, come back from lunch, visit with friends, finish the work day, go home, have dinner, pack their bags, go visit relatives for a fortnight, and return to the office just in time to see their query results displayed on the screen.

To solve the problem, we used Citrix to give these users access to a beefy Windows NT box that had a zippy connection to the database servers. They could then perform their queries in a reasonable amount of time no matter what kind of workstation or network connection they had.

The only problem was that Citrix relies on Windows NT accounts/domains for user administration. In a rather brilliant display of IS&T ingenuity, we were able to get around this by using a “dynamic local user policy” in ZENworks which would dynamically create Windows NT accounts on the Citrix server whenever an authorized user logged in.

It was nothing earth shattering. But I think it’s a neat example of using our products to solve real-world problems in non-obvious ways.

Acknowledgments

Nothing in this beigepaper represents original thought on my part. I couldn’t have written a word without the generous help and input from everyone in Novell’s IS&T Global Technical Architecture group. (I’d name them all individually but they’d probably get spammed.)

Send all comments, questions, corrections, and/or complaints to:

grettir@neticus.com

Tasty baked goods can be sent to:

Grettir Asmundarson
PRV-C-122
122 E. 1700 S.
Provo, UT 84606

And please note that Grettir Asmundarson is just a ridiculous pseudonym, so don’t bother trying to call. You’ll only confuse our receptionist.