

7

Upgrading to Notes/Domino 8

After you have decided to upgrade to Domino 8, you will need to create an upgrade plan. Most companies are not able to upgrade all users and servers at once. There are several things that you need to consider before you upgrade. This chapter explains these things. Overall, the upgrade process from Notes 6 or Notes 7 client is relatively easy. You can use the SmartUpgrade process to help upgrade these clients. (See chapter 6 for more details on SmartUpgrade.)

This chapter is divided into two main sections. In the first, we look at the Notes/Domino upgrade process in general, discussing concepts and steps that should be considered whenever you upgrade to any major release of Notes/Domino. In the concluding section, we look at upgrade issues that are specific to Notes/Domino 8.

The Domino/Notes Upgrade Process

The Domino/Notes upgrade process consists of a number of phases, as follows:

- Vision and direction.
- High-level architecture analysis
- Use cases.
- Requirements.
- Agreements.
- Final target architecture.
- Creating the design and upgrade plans.
- Creating test plan.
- Testing.
- Creating upgrade process document and plans.

- Executing logistics plans and schedules.
- Creating the pilots.
- Updating and final changes.

We discuss each of these phases below.

Vision and direction: This phase is where you define your goals for the upgrade. These goals can include your business needs, a basic idea of your current IT architecture, and some rough time lines for the upgrade. A simple vision charter might read something like this:

THE COMPANY will upgrade their ND5/6/7 architecture to Domino 8 in X months, taking advantage of new Domino 8 features, and will also consolidate several servers during the upgrade.

High-level architecture analysis: Before you upgrade, make sure you know what you have. Experience tells us that most companies cannot identify 100% of their environment. A good review is prudent so as to keep surprises to a minimum. Take the time to obtain a list of applications, including email applications and custom applications, backup systems, virus scanners, and web-based services and appliances. Build an inventory of all things that "touch" Domino. This will help you identify any items that may be affected impacted by the upgrade.

Use cases: A use case, in this context, is a statement and description of a system/ service that defines the use and behavior of an environment. A basic use case should include the following elements:

- Upgrade steps
- Description of requirements
- Goals to help target requirements
- Identification of "actors" (the people using the system: users, administrators, operators, and so on)
- Identification of associations between use cases and actors

These documents will help you build a set of requirements. In each use case, you should also identify various states of the upgrade. Examples include upgrading the servers, and enabling the new mail policy feature once all of the clients and server have been upgraded.

A use case can point to the need for:

- Client upgrade
- Server upgrade
- ODS upgrade (optional with Domino 8)
- Communications and transformation management
- Application upgrade
- Custom API upgrades
- Calendaring and Scheduling (including rooms and resources)
- Administration tool upgrade
- SMTP service upgrade
- Security impacts
- Directory impacts
- Process upgrade
- Help desk

A sample use case is included at the end of this chapter.

Requirements: When all the use cases have been created and agreed on, you can summarize them into a total list of requirements. These use cases and requirements can be used to determine upgrade steps, use of new features, systemic impacts, budgets, and time-lines. These requirements will be used to create the "draft" target architecture.

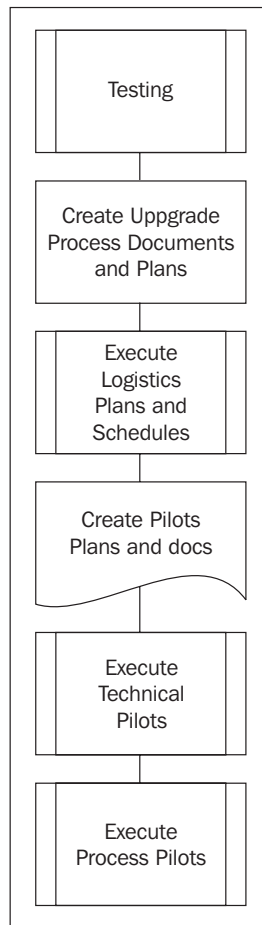
Agreements: This is where you will build out your budgets, build out decision records, and obtain agreements from all interested parties in your organization. After all of the agreements have been approved and signed, then your target architecture can be finalized.

Final target architecture: At this point, the final target architecture can be created. In most organizations, there will normally be a phased approach. It can take several iterations to get to this final architecture. One example would be new Domino 8 programming functions. In order to take advantage of this feature, you will need to have both servers and clients upgraded before the new functions are enabled.

Create the design and upgrade plans: This step is where you start the process to detail the upgrade process. Also, you begin documenting the process that will be used as a step-by-step upgrade guide.

Create test plan: Remember the identification of new features and requirements? This is where you create a test plan to test each of the upgrade elements, which includes the server, clients, applications, custom tools, and other items

Testing: The flowchart below shows the testing and pilot process:



Each part of the upgrade should be tested before you actually put any new technology into a production environment. Most companies execute what are known as "unit" or component tests. These tests are the basic components of the new technology. For example, you might choose to test the Notes 8 client on a sampling of your current PCs. This particular test verifies that Notes 8 will run on your exiting hardware, and does not impact any other applications and/or PC environment. As testing progresses, you will start to include each other element into the environment, for example, Notes 8 on the network, Notes 8 on applications, Notes 8 client that will access a Domino 6 or 7 server, and so on.

The goal is to test Notes/Domino in a holistic test environment that replicates various part of your production environment.



One very important step is to contact each vendor for any third-party tools and utilities. The upgrade process will make changes to the directory, and then to each server. Be sure to contact every vendor and determine whether or not Domino 8 (or any new release of Notes/Domino) is supported by that vendor. Double-check to verify that APIs have been recompiled (as needed) by the vendor, and that the new directory is supported. Then do your own testing, to make sure all is working as advertised by the vendor.

Create upgrade process document and plans: Create all of the upgrade steps, procedures, and schedules, training, and Frequently Ask Questions documents. Some of these documents will be the actual upgrade steps and checklists. If you are upgrading a large number of servers and users, then you can use a tracking database and/or spreadsheet. The results of the testing will be manifest in the upgrade process. Also, communications plans should be created at this time.

Execute logistics plans and schedules: This is where you order any equipment, hire any additional staff, and start the overall upgrade process. Included in the scheduling process will be the execution of the pilots. (See the following step.)

Create the pilots: Next, create and document each pilot that is needed. You will need two pilot types: non-production pilots (technical pilots, process pilots), and production pilots (shown in the diagram later in this chapter).

As we mentioned earlier, you should test as much as possible in a test environment before executing a production roll-out. These non-production pilots are an opportunity to test each step of a process. These should include:

- Upgrade steps.
- Training and education.
- Communications.
- Help desk testing and FAQ.
- Executive help staff.

One important step of the pilots is "lessons learned". Each pilot is an opportunity to modify upgrade steps and processes.

Non-production pilots are normally separated into two types, technical pilots and process pilots. Technical pilots verify that each holistic step of the upgrade works correctly. Process pilots verify that the actual checklists and documents are correct.

Transformation management: When upgrade/migration projects fail, it is always people issues, not technical problems, which are the root cause. Transformation management (TM) is a formal process to help identify and mitigate the people issues associated with each project. Change is implied by any upgrade and/or migration. TM takes into considerations the various work environment issues that occur during the upgrade/migration project. One core fundamental part of any TM plan is communication.

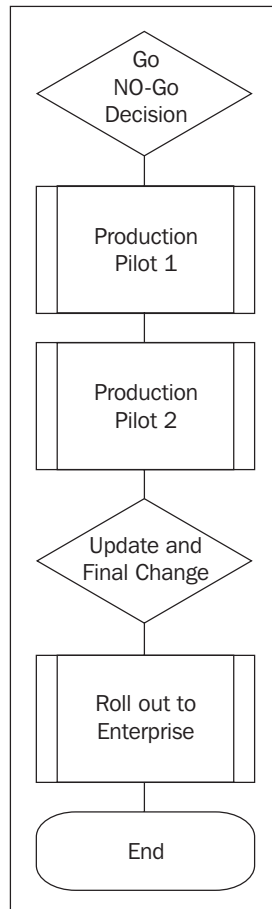
Here is a simple example that could form the basis of the communication part of a TM plan:

Your company is taking on a migration/upgrade to Domino 8. Your company may experience a few changes to the architecture, some changes to end user clients, and possibility a few changes to the administration teams. In some cases the end user and administrators may require some training on Notes and Domino 8. Your company should consider the following activities as part of the administration and migrations team's transformation management activities:

1. Develop a team name: for example, *The Domino8 Upgrade Team*. (Also create a team logo.)
2. Develop a team charter: for example, Migrate/upgrade x number of users in n number of weeks.
3. Announce the date of the final "migration done" party.
4. Create an intranet web site with a list of FAQs, and the names and pictures of the migration team members.
5. Create a Red/Yellow team to isolate the migration team from the end users (post-migration issues).
6. Develop two sets of communications from the migration team to end users. This should be added to the overall TM planning.
7. Introduce users to the migration team. These users will be notified about the migration team and their purpose. Also users should be instructed about whom they should contact with questions about the upcoming migration. LPS/ISSL recommends that the help desk be trained about the migration and possible end user questions.
8. A pre-migration FAQ should be created and hosted on the Intranet.
9. Print posters with the name of the team and the logo on the top of the poster, and place them where they are likely to be seen (break rooms, elevators, and so on).

10. After users have been migrated, send out a weekly message to migrated users, with "Tips of the week" and other relevant information.
11. Three milestone meetings will be needed for your company's migration/upgrade:
 - The opening meeting: The CIO or CEO should open this meeting. A quick five-minute pep talk is all that is needed from the VPs, but it could be important for the team to let them know that there is executive support. The lead project manager will launch the migration. Announce the plans, hand out procedures, and review the whole process.
 - The "half-way" meeting: This occurs when 50% of users migrated have been migrated. This is a great cause of celebration, so give out some special awards!
 - The final party: At the "98% of users migrated" mark, close out the migration. Move the remainder of the migration processes into the permanent support staff.
12. Close out the project.
13. Transfer any left over migrated users into the current customer steady state support staff.

A "go/no go" decision is made before the production pilots are executed. This decision will be based on the results of the testing and pilots. If all have been successful, then the next step will be the production pilots.



Once all of the pilots had been completed, you will need to start the actual upgrade process. Use a set of "friendly" users (never use executives!) for the first pilot. The preceding diagram shows two production pilots. In reality, you will execute as many as are needed. Each pilot will provide lessons learned to be used for the next pilot.

Update and final changes: After all the pilots have been executed, and you have had an opportunity to update the processes and make any final changes to the overall upgrade process, you will be ready to roll out the upgrade to your enterprise.

Notes/Domino 8 Upgrade

So far we have discussed a generic Domino/Notes upgrade process. In this section, we discuss the specific upgrade issues for Notes/Domino 8.

Reviewing the Current Infrastructure (the Health Check)

Before you upgrade you will need to identify the components and systems that will be impacted by this upgrade. This is an opportunity for you to execute a system wide health check – this normally includes a review of the following:

- **Servers:** Identify any existing issues, such as crashes, problem servers, and slow access. Your servers should be tested before you process the upgrade. Be sure to set up similar servers in a test environment, and use `server.load` to test the performance capabilities of your servers. Also, make sure that your servers are not "sick"; you should not upgrade a server that is crashing or having hardware issues. Fix issues and problems before you upgrade.
- **Monitoring systems** (Tivoli, DDM, BMC, and so on): There are many new monitoring features with Domino 8. Be sure that your current monitoring systems work with Domino 8, and that there are no conflicts with any new features.
- **Directory architecture** (directory analysis, directory customization): This is a big step. Analyze your directory, and determine whether or not there is any customization. Determine whether or not any custom design features (views, forms, and so on) need to be moved into a new directory. In some cases, you may find these customizations are no longer needed in Domino 8.
- **Clients:** Test your clients, and make sure that your current hardware and software configuring will support Notes 8.
- **PDA and/or other wireless systems:** with each new release, new features are added. Be sure to verify that any new features don't conflict with your PDA devices. For example, we have seen in the past where a ODS change broke the connection between the local PDA and the data on the Notes Client.
- **AdminP status:** This is a great opportunity to make sure that `admin4.nsf` is replicating to all servers, and that all AdminP ACL database assignments are correct. Also, there are new features that allow you to set up several directory AdminP servers using Extended Directory access control.
- **Application analysis:** This includes any issues with applications being upgraded, custom templates, and API analysis. Be sure to test your applications with Domino 8. In general, upgrading to Domino 8 should not result in any issues relating to existing applications, but it's always a good idea to test with any upgrade. Make sure that your custom APIs still work as needed with Domino 8. In some cases, you may need to recompile some of these APIs, and in other cases, you may no longer need the APIs.

- **Custom templates:** Check for customization of system templates. Compare this customization with any new features in Domino 8, and determine whether or not you need to move this customization into the templates and applications. The use of the IVES Team Studio Delta tool will help you with your analysis.
- **Messaging architecture** (including NRPC services, SMTP services, messaging tracking, enterprise-wide communications, mass mail, corporate communication, and co-existence with other messaging systems and other tools): NRPC rarely causes problems during or after upgrades, but it's never a bad idea to test this anyway. Make sure that NRPC Notes Name Networks (a.k.a. NNN) or Domino Named Networks (DNN) work as before the upgrade. Test each SMTP Services feature that is enabled. Test each Domino message tracking feature that is enabled in your current environment. There are a wide variety of mass-mailing tools and other customized features that may be installed in your environment. Be sure to test each of these tools. Large enterprise organizations can have several varieties of mail systems and servers. Test any custom interfaces, software, and SMTP connectivity. Be sure and check out the new "out of office" configuration features for ND8 – you now have the option of using the router to launch the "out of office" messages in place of the standard agent.
- **Other services and servers:** There are a large number of Lotus/IBM products. All of these need to be tested. Examples include Quickplace, Sametime, LEI, SMTP gateways, virus scanners, backup services, and provisioning systems. Ensure that these products (and the versions you have installed) are supported with Domino/Notes 8:
- **Domino replication** (activity logging, replication topology, replication settings, connection documents, access control, replication schedules, cluster replication, if enabled): Our experience with most upgrades is there is rarely an issue with replication and upgrades, but be sure to test this.
- **Messaging topology** (server topology, named networks, domains, inbound and outbound message flow, routing requirements, routing priorities, volume metrics, client strategy, server vs. local replication, alternate client access [POP, IMAP, web, mobile users], hand-held device recommended practices [Treo vs. Blackberry]): With ND7 a new task was added: the Room and Resource manager task; be sure and test your Rooms and Resources architecture as part of your upgrade.
- **Mail-enabled applications:** Most Domino architectures will have several mail-enabled applications. These can, and will, be affected by an upgrade to a new release of Domino. Overall, Lotus Notes and Domino provides great backward compatibility, but you still need to exercise due diligence regarding any new Lotus Script elements, new "@Functions", and new design elements. At the minimum, verify that the mail-in-DB records are functioning correctly.

-
- **Architecture** (high level review, connections to internal systems [networks, unified messaging, SMTP/Internet domains]).
 - **Network** (platforms, DNS/DHCP, remote access): Overall, we see very little impact to this area with an upgrade. Again, take the time to test the new release to make sure that the 'basics' work.
 - **Calendar & scheduling** (user calendaring [delegation, manager access], enterprise scheduling [resources, shared group calendars]).
 - **Directory** (directory architecture [in particularly directory design], directory management, directory synchronization, naming [Servers, users, O and OUs]).
 - **Security** (ACL access, anonymous access, encryption and certificates, certification practice statement, organization structure, ID management, access controls [file server system, console and physical, server access/pass-thru/deny, client execution control, administration access]).
 - **Capacity**: Determine if servers can handle current user loads (mail file size, hardware sizing), load balancing/sage, and capacity planning. Our experience is that each new release of Domino provides better performance in CPU and memory, and that each new release provides more features. With each new feature or function you will find additional resources being used – in particular, system memory. Be sure to monitor, via statistical baselines, the impact of a new release on the current setup of hardware.
 - **Configuration settings**: These are a very important part of the server upgrade. Review each server configuration to determine if you need to make any changes.
 - **Environmental variables**: Check for abandoned `Notes.ini` variables and obsolete `notes.ini` settings. Check the on-line support tools for this list. The release notes may also have some information about the current set of supported `Notes.ini` variables.
 - **Management and administration** (change control, administration model, client management, remote access recommendations, staffing levels, service monitoring and reporting, systems management, backup and restore model).
 - **ESX/VmWare**: There are a number of Domino enterprises that are looking at ESX and VMware. At the time of writing, there are limited sets of data regarding the successful use of ESX for Domino 'messaging'. If you are considering using ESX for ND8 messaging, we suggest the following:
 - Review the current supportability statements (URL and release notes) from IBM on this topic.

- If possible, do not upgrade to both ND8 and ESX/WMware at the same time. This is the old rule of not making too many changes at once
- Be sure to set up a test lab to check how ESX will work with a shared CPU and memory model. Also pay close attention to the Disk I/O queues. Server.load can help you with test loads/scripts.

If you are using clustering, you should monitor the workqueue depth and seconds on queue statistics.

The Upgrade Process

After you have checked the infrastructure, it is time to start the upgrade. The following steps show the basic upgrade path. This path can vary, based on your research and the use cases that you have created:

Systemic normalization: The first step of your upgrade is to "normalize" your architecture. We have already mentioned that it is important not to make any changes, upgrades or migrations to an environment that is "sick". Take the time to review each health check category and determine if your environment is stable. If it is stable, you can then upgrade your architecture.

Upgrade the Domino Administrator clients: Upgrade all of your Domino Administrator clients. Verify that all features and functions run in the current environment before you upgrade your first server.

Upgrade the Domino directory: This step can be executed before you upgrade your first server. Remember the use case above? Use that to drive the upgrade of the directory, making any customizations and changes. Be sure to work with IBM/Lotus support to make sure that the directory is backward-compatible with your current directory. (You should have done this in the testing phase of the upgrade.)

Upgrade the administration server: This is a very important server. AdminP requires that you assign an administration server to the Domino directory (`names.nsf`).



The AdminP server task runs on all Domino servers. This task loads when the Domino server is first started, and is controlled through the `notes.ini` variable `ServerTasks`. The AdminP server task wakes up on periodic time intervals (specified in the Administration Process section of the Server document) and executes commands waiting in the Administration Request database. Each command placed in the Administration Request database has an assigned proxy action. These proxy actions are essentially the op-code that runs the administration process. Each command placed in the administration request database is represented by a document. Each document has a number of fields, including one called Proxy Action. After each action has completed on a server, a response document is created to indicate the status of that request.

There is a new option (since release 7) to use multiple servers to maintain the Domino directory. If a Domino domain is geographically dispersed, then you can use several servers to process administration requests:

Carefully evaluate your Administration server: Due to the new complexities of Domino 8 and some new proxy actions, you may need to have a dedicated administration server. AdminP can generate a large number of proxy actions as your architecture grows.

Upgrade utility servers: This step can be different with each customer. In some cases, the hub server can be upgraded first, then the utility servers. Utility servers are defined as SMTP, support, tools, and other servers. In some cases, vendors may not be ready with their updates to support a new release of Domino.

Upgrade hub servers: Upgrade each server, and then monitor the "normal" operations between each upgrade. Verify that replication is still working, that agents are still executing, and that mail is still routing.

Upgrade spoke/messaging servers: After the hub servers have been completed, upgrade your spoke and messaging servers.

Upgrade specialized servers: In some cases, these may be some of the first servers you upgrade. One example would be specialized backup software. Once again, you need to contact your vendor before you upgrade your first server or upgrade the directory. The issue is backward compatibility. Verify with each vendor that the tools and utilities will work with each release.

Upgrade the application servers: One important step is to test the applications before you upgrade. There are several tools listed in the reference section of this book that will help you.

Upgrade Notes clients: You are now at the point where you can upgrade the Notes clients. Smart-upgrade can be used if you have Notes/Domino 6/7 installed. If not, you can use a MSI/MST type installs process to roll out the code.

Implement new Domino 8 features: When all servers and clients have been upgraded, you can implement the new Domino 8 features. Each feature should be tested, and in some cases you may need to build an architecture/design for each feature. One new feature that you should consider is the mail policy. This is a new policy that can be enabled after you have upgraded both servers and clients.

Upgrade applications: When your architecture is pure Domino/Notes 8, you can start to implement new Domino 8 features in your applications. Use the testing methodology listed above.

Special Feature Upgrade Considerations

Lotus Notes and Domino 8 include a number of important new features. These features are discussed in Chapters 2 and 3 in this book. Be sure to consider these features as part of your upgrade planning:

Productivity tools: Notes 8 includes a set of office productivity tools which support the Open Document Format (ODF) standard. These include IBM Lotus Documents (create, edit, and share word processing documents), IBM Lotus Presentations (create and deliver presentations), and IBM Lotus Spreadsheets (create spreadsheets and analyze numerical data).

LOB: Notes/Domino 8 also makes it easier to integrate line-of business (LOB) solutions and data into new types of applications, called **composite applications**. Composite applications, which are manifested in the front end of a Service Oriented Architecture (SOA) (See chapter 3.)

Mail recall: This is a "planned" option of Domino 8. Work with your administrator to determine if you can use this feature and what options are available.

Improved "out of office" capabilities: This includes an option to specify special hours in addition to specific dates. Now notifications can be sent almost immediately if a person has enabled the "out of office" agent.

Central management: Domino 8 offers the option to centrally managing initial deployment and upgrades of Notes 8 client software and composite applications. Using server-managed provisioning, you can even deploy different Notes 8 client features to different users. This new capability will support the existing Notes SmartUpgrade feature. (Refer to Chapter 6.)

DB2: Domino 7 introduced an option to use IBM DB2 as an alternative to the traditional Lotus Notes storage facility (NSF) for storing Lotus Domino databases. Domino 8 will now support DB2 as part of a standard install.

Use Case Document Example

The following is an example of a use case document. You can use this example as a guideline when creating your own use case documents, which is an important step in the Domino 8 upgrade process.

Example Use case - Domino Server Upgrade.

Use case

Domino Server Upgrade

Subject area

This use case identifies the basic steps needed to upgrade the messaging servers from Notes/Domino 6.x (or 5.x) to Domino 8.

Business event

The upgrade will provide new TCO and management features to your company.

Actors

- Architecture team
- End user
- ISSL
- Administration team
- Operations

Use case overview

These use case deals with the architecture

Help and support

Be sure to checkout the following sites for help and support from IBM and Lotus.

Upgrade Central:

<http://www-306.ibm.com/software/lotus/support/upgradecentral/index.html>

Forums:

<http://www.ibm.com/developerworks/lotus>

Summary

In this chapter, we presented a high-level overview of the steps involved in upgrading your Notes/Domino environment to release 8. We began with a generic description of the Notes/Domino upgrade process. We then concluded with tasks and considerations specific to upgrading your environment to Notes/Domino 8. We also included an example use case that you can use as a template for your own use cases, as well as links to sites that can provide you with additional Notes/Domino 8 upgrade information.