

Rocket NXT 4 Build & Deployment

Rocket NXT 4 is a complete electronic publishing suite for compiling, storing, securing and distributing your reference materials online, on CD/DVD or within corporate intranets, with rapid search and intuitive navigation.



Rocket Software, Inc

275 Grove Street
Suite 3-410
Newton, MA 02466-2272
USA

www.rocketsoftware.com

Tel: (617) 614-4433
Fax: (617) 630-7112

Table of Contents

Table of Contents	2
Overview	3
Basic processes	3
Basic build process	4
Define a Library.....	4
Configure options for including content in the library.....	4
Build summary	5
Basic deployment process.....	5
Validate the build.....	6
Secure the system	6
Collect the files	6
Deploy	7
Handling updates	8
Updating and upgrading licenses	9
Variations and options	10
Multiple content owners all need control to their content builds	10
Content collections from other publishers must be integrated with our content	11
Existing build processes utilizing CCBUILD	11
Integrating corporate content and publisher content.....	11
Content network linking.....	11
Integrated sites inside the firewall.....	12
Offline customer already has NXT Solo installed	12
Providing scalability, reliability, and performance	12
Recommended minimal system architecture	12
Scaling build processes	13
Scaling online servers	13
Content scaling	13
User scaling.....	14
Scaling update servers	15
Ensuring reliability	15
Reliability and the NXT Online Server.....	15
Reliability and updates via the Library Installer	16
Recommended system specifications	16
Build systems.....	17
Online server systems (including intranet deployments)	17
Update servers.....	17
Summary	18

Overview

Rocket NXT 4 is an electronic publishing suite that gives you a single, integrated process for storing, assembling, securing and delivering reference libraries—on your corporate intranet, to your employees' hard drives, over the Web, or on CD or DVD. With powerful search and navigation tools, NXT 4 makes the information your business depends on more accessible and useful. NXT 4 includes all the pieces you need to make new information available quickly, keep your existing content current, and tightly control access to

“With NXT 4, you can publish the same content to Internet, intranet, or offline users quickly and easily”

sensitive information. With Rocket NXT 4, you can lower the costs and risks of managing and maintaining your corporate information and make up-to-date content more available and useful to everyone in your organization.

This paper describes the basic NXT 4 build and deployment model and reviews common variations of the model for both commercial and corporate publishers. Additionally, this paper reviews system configurations for performance, reliability, and scalability of the system.

Business managers should review this paper to understand how NXT 4 complements their business practices.

System administrators and **content managers** should review this paper to understand how NXT 4 fits into their current content production and delivery environment.

For specific implementation details, please refer to the NXT 4 documentation.

Basic processes

The NXT publishing process has two distinct phases: build (content compilation), and distribution (content deployment). Throughout both phases, content owners maintain responsibility for creating their content and delivering it to the build process, and business owners maintain responsibility for how the content is distributed.

We recognize that publishing efforts start with the content. Publishers use a variety of systems and processes to author, edit, review and approve content for publication. The NXT 4 publishing process fits into the publishing system once content is approved for publication.

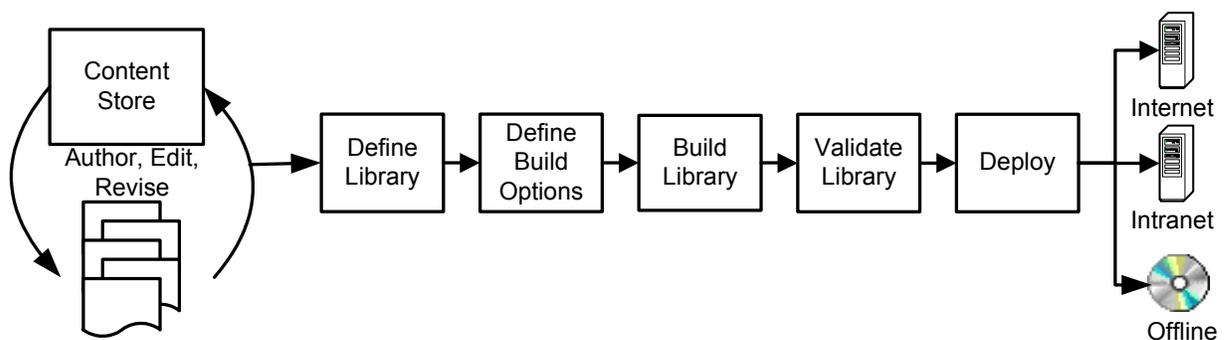


Figure 1: Basic build & deploy process

Basic build process

The basic build process using Rocket NXT 4 applications has three steps:

1. Define a library
2. Configure options for including content in the library
3. Build the library

All of these steps can be performed in the Library Manager, an application included in Rocket NXT 4 Builder.

Define a Library

A library is a representation of your content; it contains an organizational structure for your information (similar to the rows and shelves you would find in a public library). You define the structure for your library and decide where your books should appear in the library.

For example, if you were publishing legal information in the United States, you might create a library that looks like this:

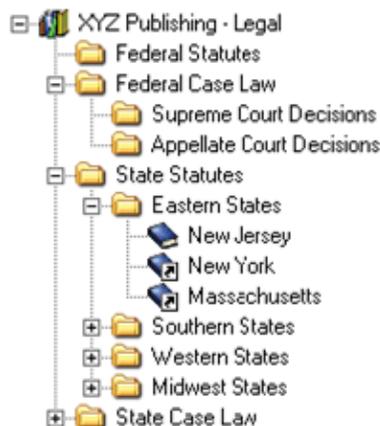


Figure 2: Sample library organization

Keep in mind that a library typically represents all of the content you would make available to a particular market, and from your library you can create and distribute several distinct publications, each of which can be aware of each other. For example, you could create an Eastern States publication and a Federal Case Law publication and distribute them separately. If a customer receives both, the user is able to access and search both publications as if they are one larger publication. Typically, you create the main body of your library once. You may periodically change the library—adding or removing items as appropriate—but you will rarely, if ever, need to rebuild the entire library.

Configure options for including content in the library

Once the structure of your library is created, you add the content. Using the Library Manager, you configure a set of rules for retrieving content from your data store—such as a relational database, a content management system, or a file system holding HTML, XML, Office documents or PDF files. These rules specify which types of documents to retrieve and how to process them.

For example, if your data store is a file system, you set up rules for traversing the file system, such as:

- o Which directories are included
- o Which file types are included or excluded
- o How the files should be ordered
- o How the files should be fielded and indexed

For simple builds, you may specify one set of rules for all folders. For more control, you can set overriding rules for individual folders. The Library Manager allows you to set the global rules and varying sort orders for individual directories; more complex overrides are done via an XML configuration file.

With database and other data stores, we provide you with rule options that apply to that data store. For example, with databases, you use SQL queries to determine which content is retrieved.

The Library Manager includes intuitive screens to walk you through the process of creating these rules. Publishers who require a greater deal of control, or who want to integrate the generation of rules through an automated process, may create the rules directly in an XML file.

Build the library

The final step in the build process is to actually build the library. When you build a library, Library Manager performs the following:

1. Connects to each data store, one at a time
2. Extracts a copy of the information from the data store, based on the rules you have defined
3. Indexes the information and stores both the index and the content in a secure data file known as an NXT content collection.
4. Creates a library install file containing all of the structural information about the library

Please note that you can build a subset of the library at any time. If you don't have all of the configuration options set up for all of your content, you can choose to build only those elements that are ready.

Build summary

NXT Builder includes the Library Manager application, which lets publishers define their libraries, set the configuration options for including their content in the library, and build a set of content collections based on the library definition. Once the initial library is created, you are ready to deploy it to your users.

Basic deployment process

With NXT 4, you can publish the same content to Internet, intranet, or offline users quickly and easily. The build process creates a set of content collections, which are secure and portable containers that can be distributed on any of these media without any changes. You don't need to go through separate processes to serve your different markets.

The deployment process has four steps:

1. Validate the build (quality control)
2. Secure the system using license management (for commercial publishers only)
3. Collect the necessary redistributable files for a publication
4. Deploy on the desired media

Validate the build

Once the library has been created, you need to verify that everything works correctly before deploying it to your users. Because all of the information is stored in content collections, you can check the collections once and then distribute them with confidence to your Internet, intranet or offline users. To validate your content collections, load them on a staging server running NXT Online Server. To check a few collections, you may load each content collection individually using the Content Network Manager, an administration tool included with the NXT Online Server. To load and validate the entire library, use the Library Installer application. Once the content is loaded on the staging server, use a browser to connect to the server and view the content. Ensure that all of the content is there, that links work correctly, and that the content is searchable.

Secure the system

For offline or intranet deployments, commercial publishers can secure their content libraries using license management, which extends the Library Manager and enables publishers to create subscriptions for their content.

If you are distributing commercial applications using the NXT Intranet Publisher or NXT Offline Publisher, your copy of NXT Builder installs the license management components. Using the Library Manager, you create one or more subscriptions, which define the access rights for your information—what content in the library is viewable and searchable, when the content expires, and how many concurrent users can access the information. You may create as many subscriptions as necessary to support your business models.

Finally, after creating the subscriptions, you create one or more licenses to distribute with your content. The subscription defines the rights, and the license enforces the rights. When customers upgrade, you send them a new license or allow them to retrieve a new license.

Collect the files

Once you are sure the content is correct, you need to gather the appropriate files for a publication. For initial deployments, you need both the content and the user interface files.

The Library Manager creates the content files that you need (content collections and a library install file). Your designers create the user interface files. The user interface is the set of HTML or XML templates used to display your content in the browser. NXT 4 provides a set of default templates that you may use or modify to meet your needs. These default templates include a table of contents, search forms, hit lists, basic navigational elements, and a viewing window for the documents. They are included with the NXT Online Server and NXT Solo products.

- o If you are distributing your entire library to your users, you need only to gather the content collections and the library install file generated by the Library Manager.
- o If you are distributing sub-sets of your library to your users (as you may for a CD or intranet deployment), you need to gather the specific content collections for that subset and the library install file generated by the Library Manager.
- o If you are distributing license managed content, you also need to gather the appropriate licenses for your publication.

Once you have identified the specific files you need (templates, collections and the library install file), you are ready to deploy.

Deploy

The specific process for deploying your content depends on the deployment media you choose to use.

For Internet or intranet deployments inside your firm, you deploy the content on the NXT Online Server.

For offline deployments to an employee's hard drive or to your customers via CD or DVD, you deploy the content using NXT Solo (commercial publishers use the NXT Offline Publisher, which is a redistributable version of NXT Solo).

For intranet deployments, where you are placing your content inside someone else's intranet, you use the NXT Intranet Publisher, which is a redistributable version of the NXT Online Publisher.

Online deployments

For traditional online deployments, for use on the Internet or within your own intranet, you deploy your library on the NXT Online Server. To do so, you:

1. Use the Content Network Manager administration tool to reference your custom user interface templates. These templates are stored in a directory on the NXT Online Server.
2. Copy the content collections and library install file to a single directory on the NXT Online Server.
3. Use the Library Installer command line interface to install and configure your library on the server. You may optionally use the Content Network Manager to add each content collection one at a time to your system.

Your site is now ready for basic use. Other options you may consider include setting up access control, creating unique views of the information for different users (including different user interface templates), and configuring content network links between two or more servers.

Offline deployments

For offline deployments (on CD or DVD), you distribute your content and templates with NXT Solo or NXT Offline Publisher. To do so, you:

1. Place the content collections and library install file in a single directory on the distribution media (CD or DVD is assumed).
2. Place the templates in a separate directory on the distribution media.
3. Include the NXT Solo or NXT Offline Publisher redistributable install on the distribution media.
4. Include the NXT Solo or NXT Offline Publisher configuration files on the distribution media.
5. If using Offline Publisher, include the appropriate license files on the distribution media.

The NXT Solo and Offline Publisher configuration files allow you to specify where the content should be accessed from (the user's hard drive or directly from the CD), where your templates are located (the install will copy them to the correct location on the user's hard drive for use with Solo), and how your brand should appear with the installation. When your users install your publication, NXT Solo installs itself on the user's system, installs the library, installs the licenses to secure the system, and installs your custom user interface templates.

Intranet deployment

For intranet deployments, you use NXT Intranet Publisher. NXT Intranet Publisher is a redistributable version of NXT Online Server, including a redistributable install application. Deployment for this option is similar to offline deployment, with the exception that you are distributing a different underlying application. To deploy with NXT Intranet Publisher, you:

1. Place the content collections and library install file in a single directory on the distribution media (CD or DVD is assumed).
2. Place the templates in a separate directory on the distribution media.
3. Include the NXT Intranet Publisher redistributable install on the distribution media.
4. Include the NXT Intranet Publisher configuration files on the distribution media.
5. Include the appropriate license files on the distribution media.

The NXT Intranet Publisher configuration files allow you to specify where the content should be accessed from (the user's hard drive or directly from the CD), where your templates are located (the install will copy them to the correct location on the user's hard drive), and how your brand should appear with the installation.

When your users install your publication, the NXT Online Server installs itself on the user's system, configures Microsoft Internet Information Server (IIS), installs the library, installs the licenses to secure the system, and installs your custom user interface templates.

Build and deployment summary

NXT 4 provides the tools you need to build and deploy your information assets to your customers and employees. Using the Library Manager, you define a library, configure the build options, create the library, secure the library, and deliver the resulting content collections to your users online, offline or to their intranets.

Handling updates

Once you have built and deployed your library, you need to determine the best way to handle updates to your content. Content is rarely static—information will be added, removed or changed in your library. You could choose to completely rebuild your library each time content changes. You could choose to rebuild the library periodically, such as quarterly or semi-annually. You could then redistribute new CDs to your offline users and update your online web site with the latest information.

NXT 4 offers an alternate option. Using the Update Build feature of the Library Manager, you can update your library every day and make the updates available to your online, offline, and intranet users without distributing new CDs. Any time content changes in any one of your content stores that would change any of your content collections, you can create an update.

The Update Build feature gathers the changed information in each data store and creates an update file for each changed content collection. These update files contain only the changed information—added, deleted or modified. The Update Build feature also updates the original content collections so they are current.

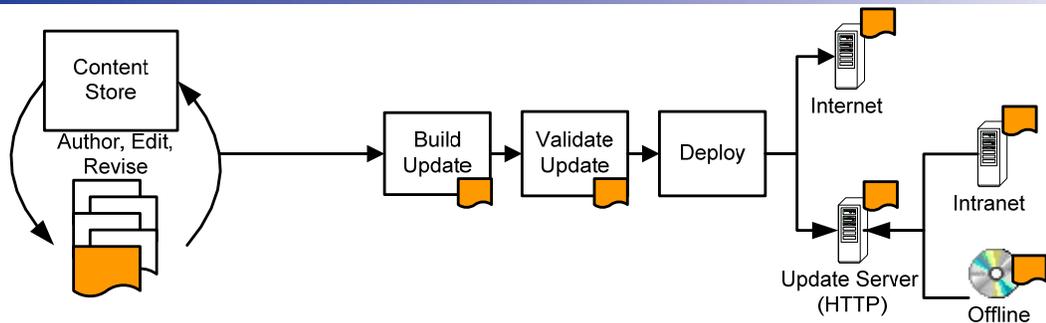


Figure 3: Update process

You deploy the update files on any HTTP or HTTPS server. Your offline and intranet users connect periodically to this server to download updates. The updates are then automatically merged into their local copies of your library and are available for immediate use. Your customers or employees get the latest information immediately, which may give you a competitive or regulatory advantage.

For your online system, you may either use the Content Network Manager to hot swap the outdated content collection with the new content collection, or use the Library Installer to merge the update files into the content collection. The hot swap is a manual process, but ensures that there is no downtime for the system. Using the Library Installer will temporarily take individual content collections offline as they are updated. As soon as the update for a collection is complete, it is brought back online and the next one is taken offline. To prevent downtime for your users when using this method, we recommend that you run two NXT Online Servers behind a load balancer. You can route all traffic to one server while you are updating the other and then switch.

Updating and upgrading licenses

Commercial publishers apply licenses to their libraries. These licenses control what a customer may see in the library, when the library expires, and how many concurrent users may access the library. Often, a publisher will distribute a demo CD or DVD, which gives the customer a free trial for a limited period of time (such as seven days) and then prompts them to purchase access to the full content. Or, they may distribute the content on a subscription basis, with a monthly, quarterly or yearly renewal.

In these cases, customers need to be able to easily upgrade or renew their licenses, and publishers need a system that won't burden their support centers. To facilitate this, NXT Intranet Publisher and NXT Offline Publisher include a Web License Management component. Using this component, customers can visit the publisher's Web site, select upgrade options, and generate a new customer-specific license without needing to speak to a support representative. When customers do call, the support representative can use the Web site to generate a new license and e-mail the license to the customer.

Figure 4 shows an update from a Solo User (Offline Publisher, which is NXT Solo with license management). This model also applies to the Intranet Publisher product.

The Subscription License Generation component provided by Rocket is a COM object. It takes a subscription definition, customer ID and expiration date as parameters and produces a secure, customer-specific license file. The user interface around this functionality is created and maintained by the publisher.

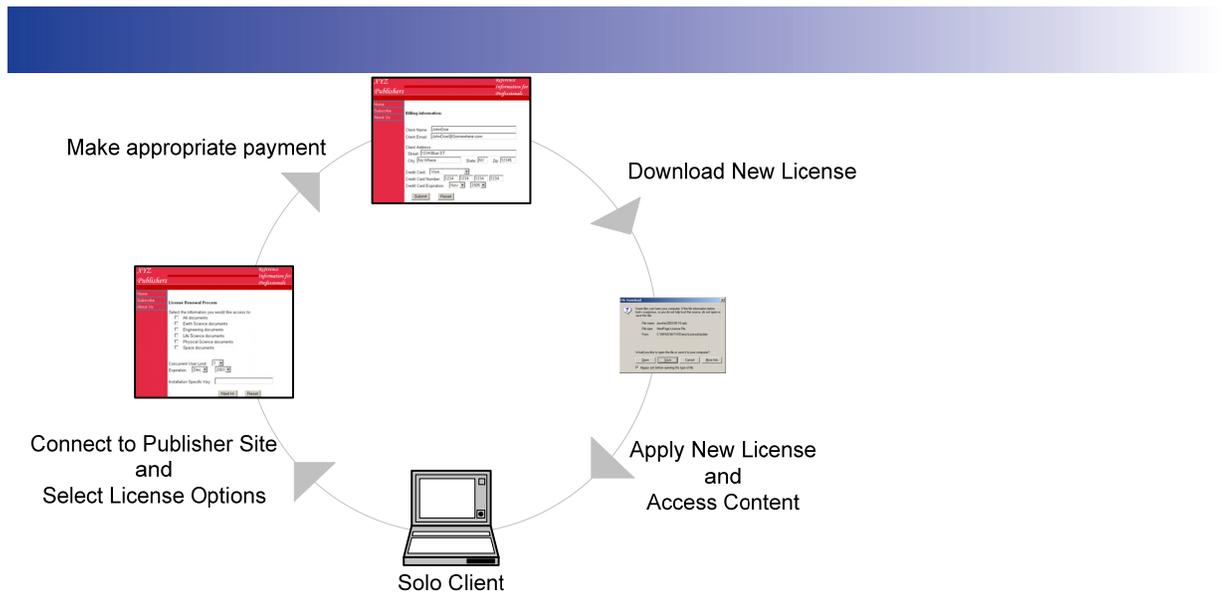


Figure 4: Web License Management

Variations and options

No two publishing systems or processes are alike. Recognizing this, Rocket has worked to make its products and applications flexible enough to adapt to your specific situation. Some variations we have considered are detailed below. Your specific situation may be different, but these examples should give you an idea of how NXT 4 fits into your system.

Multiple content owners all need control to their content builds

Publishers may have several departments all contributing content to the library. However, each department updates their information on separate schedules and do not want to coordinate the build cycles with any other department. Of course, someone has to control the final output from all the departments.

In this case, each department runs its own copy of the Library Manager. They create their content collections and updates and place them in a shared directory on the network.

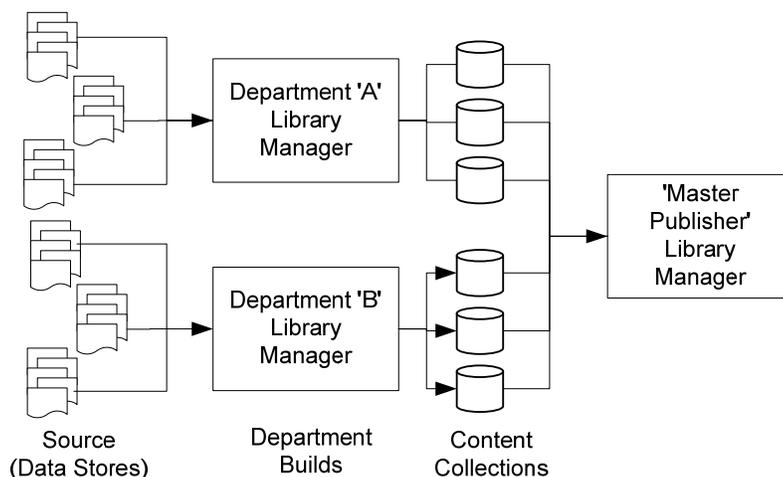


Figure 5: Distributed build process

The “master publisher,” who is responsible for the final product, uses the Library Manager to reference the content collections and updates created by the various departments. When the master publisher runs a build, the Library Manager incorporates those existing

files into its library, which is then deployed as described in the “Deploy” section of this document.

Content collections from other publishers must be integrated with our content

Some publishers allow their corporate customers to redistribute their content collections as part of their library. The publisher gives the corporate customer the content collections and the corporation creates a library that references their own information in addition to the publisher’s content collections. This model is similar to the distributed build process (shown above), except that the commercial publisher delivers the content collections to the corporation rather than just to a shared network drive. Distribution to the corporation could be handled via FTP, CD or any other distribution media.

Existing build processes utilizing CCBUILD

Many NXT customers have existing build processes in place currently utilizing CCBUILD, an NXT command line build utility for creating content collections. The Library Manager, which is new in the NXT 4 release, uses a different model for creating content collections. However, since the Library Manager can reference existing content collections, customers can continue to use their existing processes while still leveraging the other benefits of the Library Manager. We recommend that all build processes be migrated to the Library Manager, but this is not required.

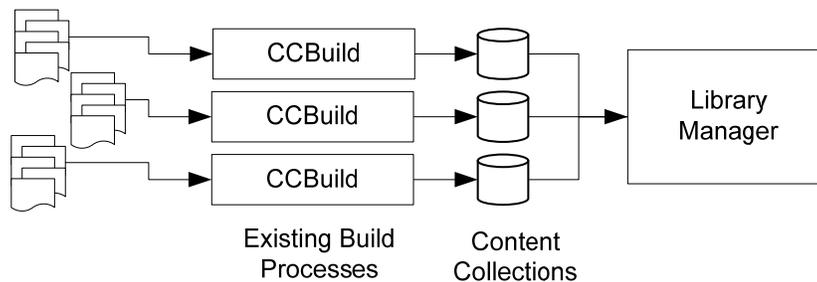


Figure 6: Existing build processes

Integrating corporate content and publisher content

Corporations that use NXT 4 for their internal content may also purchase content from external publishers for their employees to use on the corporate intranet. The method they choose for integrating the content will depend on the content distribution model from the publisher, the internal policies of the corporation, and the overall business relationship between the corporation and the external publisher.

Content network linking

The simplest model is a content network link between the corporate intranet site and the commercial publisher. Using the NXT Content Network Manager, the corporate administrator creates a link between his NXT Online Server and the publisher’s NXT Online Server. Each organization hosts and maintains their own content, but corporate users can search both sets of content. Access control at either end of the connection can prevent unauthorized corporate users from accessing the publisher content.

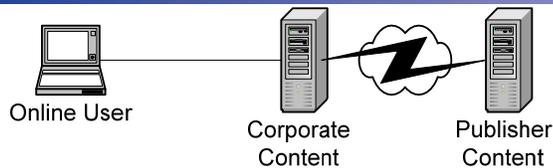


Figure 7: Content network link

Integrated sites inside the firewall

Finally, corporations could choose to integrate the content completely within the corporate firewall. If the publisher is willing to deliver the content collections and update files without license management implemented, then all content can reside on a single server (or set of servers, in order to meet your scalability needs—see Figure 8). If the publisher delivers license managed files, then the corporate content and the publisher content need to be installed on separate servers, and a content network link needs to be established between the two servers.

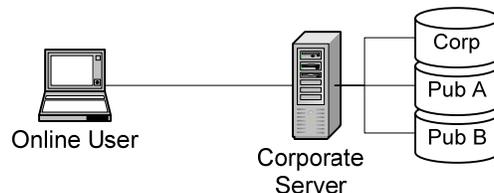


Figure 8: Single server

Offline customer already has NXT Solo installed

End users may purchase content from several publishers. Or, they may purchase several publications from the same publisher. In either case, NXT Solo may already be installed on the system when the user installs your content.

In the case where a customer installs publications from two different publishers, the user will end up with two installations of Solo. Each publication will have a different icon and user interface. In the case where a customer has purchased two publications from the same publisher, there are two possible outcomes, both of which are controlled by the publisher.

If both publications are part of the same library (which is typical), then the second publication is installed into the same location and merged into the site. Both publications are searched as if they were one.

If the publications are part of different libraries, the publisher chooses whether or not the publications are merged. If they are merged, then both libraries will use the same interface and have the same icon to launch the information. Alternately, the publisher can choose to have each publication installed separately. If so, two copies of Solo are installed and each publication will have a separate user interface and icon.

Providing scalability, reliability, and performance

While NXT 4 includes high-performance applications, you will want to take some steps to ensure that your system is reliable, that it scales to meet the needs of your users, and that it effectively delivers the information in a timely manner. The following information may assist you in planning your systems.

Recommended minimal system architecture

The following diagram shows the recommended minimum system architecture. Some of these elements may be combined onto a single server; others may be expanded upon to provide scalability, reliability and additional performance.

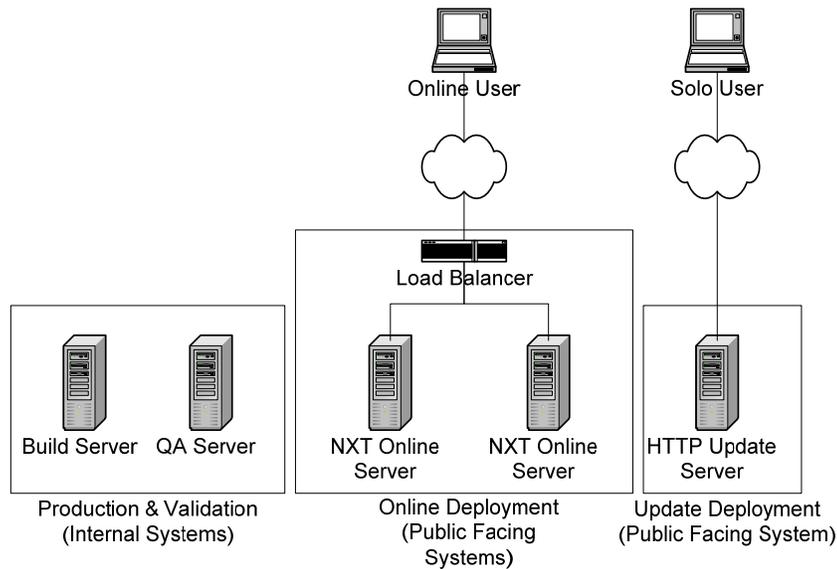


Figure 9: Minimal system configuration

Scaling build processes

Some publishers have large amounts of content to compile into content collections. Running all build processes on a single machine may not be efficient. By distributing the builds between different machines, publishers may achieve greater scalability. This process is essentially the same as the Multiple Content Owners example (see page 10). This model can be both deep (adding more build processes to the left of the diagram), and broad (adding more processes to the bottom of the diagram).

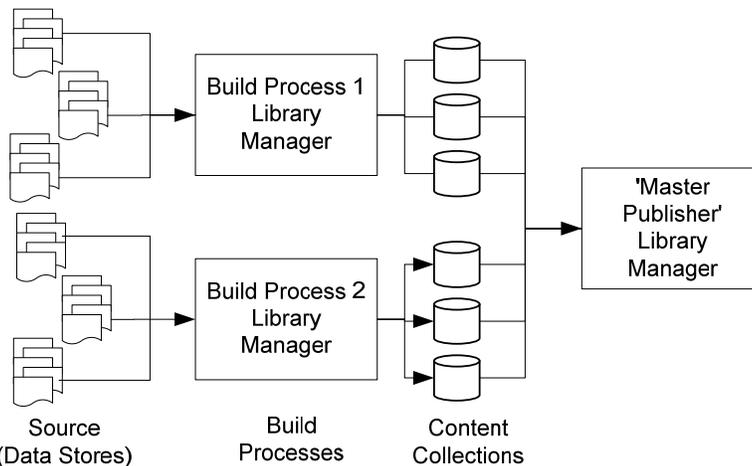


Figure 10: Scaling build processes

Scaling online servers

Content scaling

The NXT Online Server supports content networking. A content network virtually aggregates content stored on two or more NXT Online Servers. From the user's perspective, all of the content on each of the content networked machines appears as one

unified content set. They can search it, get a single set of ranked search results, and navigate through it using the table of contents. For optimal performance with content scaling, publishers should use an architecture similar to the one below:

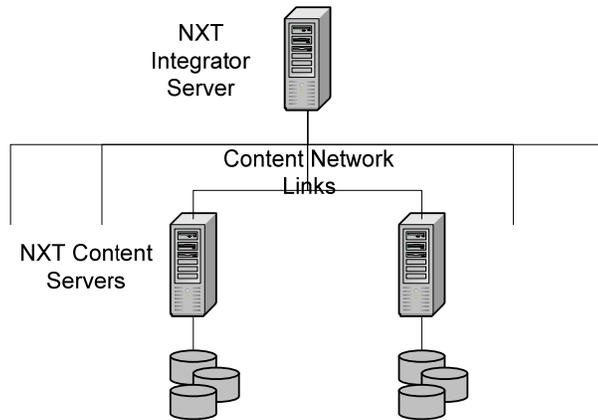


Figure 11: Content scaling with a content network

In this configuration, all of the server machines are running the NXT Online Server software. The integrator system, however, does not store any content. It services the search requests and sends them to the content servers. Publishers may add additional data servers to handle greater amounts of content.

The NXT Online Server is designed to handle up to 50 GB of content in a single search domain. A system may have more content than this, but searches should be restricted to no more than 50 GB. With the size of hard drives available today, this amount of content may be stored on a single server. However, for efficiency, Rocket recommends no more than 25 GB on a single server.

For publishers with more than 50 GB of content, publishers may consider creating separate services or publications. As users come into the system, the publisher routes them to the appropriate publication. For example, a legal publisher may have all of the statutory law for the United States, Canada and Mexico totaling 100 GB of content (40 for the US, 30 for Canada, and 30 for Mexico). When a user enters the system, he selects the area of law in which he is interested and is directed to an NXT Online Server that is delivering that content. In this way, the user gets the information he needs and the search domain does not exceed the 50 GB recommended limit.

User scaling

User scaling is handled through load balancing. If one node of an NXT system can handle 500 concurrent user sessions, then an identical node placed behind a load balancer can also handle 500 concurrent users. Load balancing provides linear scaling. The NXT Online Server works with hardware load balancers. Session state is maintained using IP stickiness for user interaction with the system. Please note that without some sort of session affinity, users may be directed to different machines between requests, losing some of the session information stored on the server.

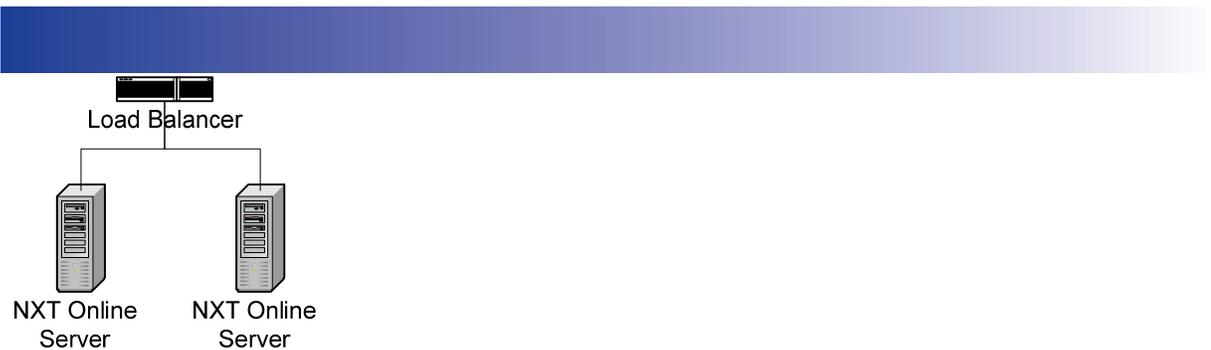


Figure 12: Load balancing for user scaling

Scaling update servers

Update servers are used to distribute updates to Solo users. Rocket requires that these servers be HTTP servers to ensure that an update server system is as scalable as possible. All of the “smarts” for the update process is in the Library Installer application used by Solo.

To scale your update servers, use a load balancer in front of two or more servers. Additionally, you can configure your libraries to point to a series of servers. If the first server is not available, Solo tries the next alternate server.

Publishers may use any HTTP server—such as IIS, Netscape or Apache—as their update server. While publishers may use any HTTP server, we recommend using IIS and enabling IIS compression. Doing so will reduce the download time for your update files. The Library Installer automatically decompresses and applies the updates.

Ensuring reliability

The NXT Online Server is designed to be reliable, but there may be times when the software goes down. The NXT Online Server has some built-in recovery mechanisms that help ensure reliability; for full reliability, Rocket recommends the use of redundant nodes and load balancers.

Reliability and the NXT Online Server

Software recovery

If the NXT Online Server encounters an irrecoverable error that causes the software to fail, the software automatically restarts. The current user state is lost, but this is usually seamless to the user. For example, if a user performs a search and is reading a document and then issues a new search, they would never know that the system restarted beneath them. However, if the user is browsing through the hit list and chooses to view the next set of search results, the context would be lost. The user would need to re-issue the search.

In short, the NXT Online Server is reliable and recoverable from most software-related errors.

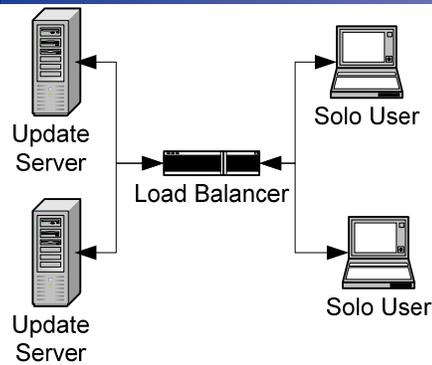


Figure 13: Scaling update servers

Hardware redundancy

There are potential errors, such as catastrophic hardware failure, from which the NXT Online Server cannot recover. Rocket recommends that any mission-critical system be replicated and reside behind a load balancer. The NXT Online Server works with hardware load balancers. Using IP stickiness, session state is maintained for user interaction with the system.

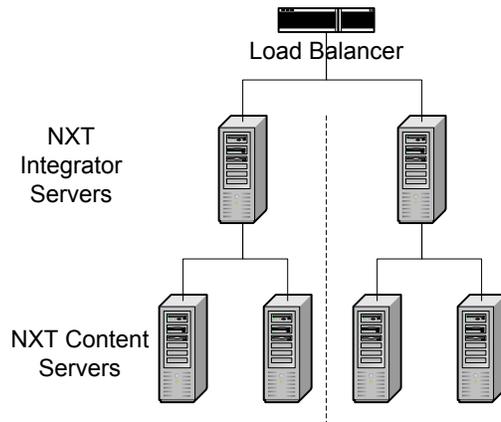


Figure 14: Extended load balanced system

Reliability and updates via the Library Installer

The Library Installer provides several failover mechanisms to ensure that updates are found and applied. To ensure that the user can access available updates when connected to the Internet, publishers may specify more than one download server. If the first server on the list is not available, then the next server is tried. Users may also select which server to download from, which gives them some additional flexibility when traveling. For example, a U.S.-based auditor who travels to Europe may choose to download updates from a server in Europe rather than the U.S. server.

Additionally, since each update location may be an entire web farm, the publisher can configure the system to be redundant and scalable for bandwidth. When a user starts downloading updates, the updates are retrieved sequentially. For example, if a user needs three updates—A, B, and C—for a single collection, Solo gets A first, then B, and finally C. If the Internet connection is interrupted during the download, Solo will resume downloading where it left off the next time a connection is available. If update A was successfully downloaded, then the next update process will start with B. Updates are applied when all necessary updates are downloaded. This ensures that the user sees the complete set of updated information.

Recommended system specifications

The recommended system specifications are those that Rocket considers reasonable minimums for building and deploying content in a full production environment. Testing or staging systems may be less robust; please refer to the product documentation for the minimum system specifications.

Build systems

Typically, Rocket recommends the following configuration for build environments:

- o Intel-compatible processor, running at 2GHz or faster
- o 4 GB RAM
- o 2 SCSI hard drives (RAID configurations provide greater performance)
- o 100 Mb NIC (for copying files to staging or deployment servers)

Efficient hard drives are critical to overall build speeds. For very large content collections, having separate hard drives for the source data and the content collections will reduce disk I/O latency. SCSI drives tend to be faster than IDE drives. Plan on 4x disk space for each collection you will build—this is necessary for the source files, temp files and final content collection.

RAM is another critical factor. The more RAM the build process can use, the more efficient it is. Disk swapping is reduced, which improves overall speed. The build tools can use up to 4GB of RAM; if you are creating large individual content collections (2GB+), consider investing in RAM.

Online server systems (including intranet deployments)

Typically, Rocket recommends the following configuration for online deployment environments:

- o Intel-compatible dual processor, running at 2GHz or faster
- o 4 GB RAM
- o RAID configured SCSI hard drives, with 128 MB cache on the RAID controller
- o 100 Mb NIC

Online server performance is affected by several factors. User load, size of content and system specifications all play a role. Similarly, content build performance is affected by the type of source content, where the source content is stored, the volume of content, and the system specifications.

In general, the larger your content sets, the greater the impact on the system. Queries across large volumes of content take longer than across small volumes of content. Of course, large and small are relative terms. The performance difference (as measured by requests per second serviced by a single server) between 5 GB and 10 GB is negligible. The performance difference between 10 GB and 50 GB is more noticeable.

This does not imply that publishers build many small content collections—from a pure performance perspective, one large content collection is more efficient than 10 smaller collections.

Update servers

Rocket does not make any minimum recommendations for update servers. The server must be capable of running an HTTP or HTTPS web server, and the web server must be capable of handling the number of download requests.

Factors that affect performance on the update server include RAM, disk drive speed and available bandwidth.

Summary

The NXT 4 product line provides several ways to build and deploy your electronic reference information. Whether serving online, intranet or offline information consumers, publishers can be assured that NXT 4 has the capability to meet their needs and fit into their existing processes.

Notices

Publication date: May 2010

Product version: NXT 4

© Rocket Software, Inc. 1998- 2010. All Rights Reserved.

The following trademarks appear in this publication:

Trademark	Trademark Owner
Rocket NXT 4	Rocket Software, Inc.

The above trademarks are property of the specified companies in the United States, other countries, or both. All other products or services mentioned in this document may be covered by the trademarks, service marks, or product names as designated by the companies who own or market them.

About Rocket Software

Rocket Software (www.rocketsoftware.com) is a global software development firm that builds and services Enterprise Infrastructure products for the world's leading OEMs, networks and software companies and enterprises. The company's products complement and extend strategic infrastructure in the areas of: application modernization, business intelligence, compliance and security, consumer productivity, database servers, database tools, file transfer, mainframe productivity, network management, publishing and search, SOA and integration, storage management, terminal emulation, and text mining. Rocket is engaged in business and technology partnerships with IBM, EMC, RSA (the security division of EMC), HP Enterprise Services, Avaya, Motorola, and many others. Rocket Software is headquartered in Newton, Massachusetts, USA.