

Software Certification Criteria

The Novell Software Test Tools test procedure is designed to help you ensure that your product meets all DeveloperNet certification requirements. However, you should familiarize yourself with the certification criteria listed here and verify that your product meets them.

General Criteria for All Applications

The application developer must supply the name and direct phone number of a technical support contact to assist Novell in the technical support of the application.

The application developer must resolve any major bugs reported to Novell Technical Support within 90 days.

The application developer must resolve all major bugs reported to Novell Technical Support before new versions of the application will be eligible for a bulletin.

All configuration notes on previous bulletins must be resolved prior to granting a bulletin for the new versions of the product.

Criteria for NLM Applications

The NLM must not threaten the security of the network in any way.

If the NLM uses any of the following NetWare resources, they must be unique among all such resources used by other NetWare applications and must be registered with Novell:

- Service Advertising Protocol (SAP) IDs
- NetWare Core Protocol (NCP) Extensions, IDs, and Names
- Application-specific Bindery Objects
- Directory Services Schema Extensions

The following elements of an NLM must be unique from those of any other Novell or DeveloperNet Labs Tested and Approved NLM:

- NLM names
- Export symbol names
- Console commands
- Server screen names

The NLM's maximum time slice must not exceed 40 ms when executed on a 33 MHz 80486 or faster server. Longer time slices may be allowed on slower servers, but no NLM may have a maximum time slice longer than 150 ms when executed on a 25 MHz 80386 server.

The product must identify itself as the source of all messages sent to the system error log and/or the file server's System Console screen.

The NLM must not leave any resources unreleased upon termination.

The product must allow any user (not just SUPERVISOR or ADMIN) with sufficient access rights or security equivalence to perform supervisor level functions and to access privileged data.

The NLM must not access memory that is not specifically allocated to it or to another NLM in the same product or product suite.

The NLM must not use any CLIB functions that require a valid CLIB context in an 'atexit()' or 'AtUnload()' routine.

The NLM must be able to detect and gracefully handle any failed dynamic memory allocation.

Where possible and reasonable, the NLM must unload gracefully upon abnormal termination.

Where possible and reasonable, the NLM must be able to detect and behave gracefully when the NetWare server is brought down while the NLM is still loaded and executing.

If the product makes use of any hardware resource of the file server, such as a serial port, it must properly register its use of that resource with the operating system.

Under no circumstances may the NLM cause the NetWare server to ABEND.

The NLM must behave gracefully in all cases when the NetWare Bindery or the eDirectory database is closed temporarily for maintenance purposes.

The product must not use unpublished NetWare APIs or interfaces.

The state of an NLM's process thread should never be halted (this will effectively stop the processor.)

The product must correctly handle and display dates up to the year 2035.

The product must be exercised in such a way as to hit 100% (or as close to 100% as possible) of the symbols/functions/procedures contained in the product's NLMs.

Criteria for NetWare MP

The NLM must meet all certification requirements for the base OS.

The NLM must operate completely and correctly in multiprocessor mode with four or more processors active.

Certification Recommendations for Developers

DeveloperNet Labs is currently considering the following recommendations for inclusion in future versions of the certification test criteria:

- Test the product against QRMs (Quarterly Release Modules) within 90 days of release.
- The product's literature includes a reference to the DeveloperNet Labs bulletin number or includes a copy of the bulletin.
- The product assumes that all memory allocations block.

Criteria for Client Applications

Manageability Criteria

The product should operate completely and correctly with the latest released version of the applicable Novell client software package (VLMs, Client32, etc.)

The product should allow any user (not just Supervisor or Admin) with sufficient access rights or security equivalence to install the application, perform network administrator level functions, and to access privileged data.

The network administrator must be able to install the product to any directory on the network without the need to “map root” a drive to the desired location.

Access to the product should be manageable by using NetWare groups.

The product must allow the user to store work files on a local or network directory where the user has sufficient rights.

The product should use some mechanism (such as file locking, read access, record locking, or synchronization) to prevent corruption caused by concurrent read/write access by multiple users.

The product should not interfere with the operation of other applications on the network.

The application must not threaten the security of the network in any way.

The application must not significantly degrade overall performance of the network.

The product must correctly handle and display all dates from 1980 to 2035.

Network Resources Criteria

If the product uses any of the following NetWare resources, they must be unique among all such resources used by other NetWare applications and must be registered with Novell:

- Static IPX/SPX Socket Numbers.
- Service Advertising Protocol (SAP) IDs.
- NetWare Core Protocol (NCP) Extensions, IDs, and Names.
- Application-specific Bindery Objects.
- Directory Services Schema Extensions

If the application provides printing capabilities, it must be able to utilize network printers.

The application should be able to fully access any relevant network peripheral devices.

The type of any eDirectory/bindery object created and/or used by the product must correspond to the actual function performed by that object within the product.

The product must be workstation-independent unless the design of the product mandates that it runs on a specific workstation.

If the product uses IP ports, it must not prevent ADMIN or equivalent user from configuring and managing them. The product also must not hard code IP ports.

Certification Recommendations for Developers

DeveloperNet Labs is currently considering the following recommendations for inclusion in future versions of the certification test criteria:

- Any eDirectory object type created or used by the product must correspond to the actual function performed by that object type.
- Where possible and reasonable, the product should take active steps to reduce data loss in the event of loss of network connection. (For example, saving data locally if the network connection is lost.)
- Anywhere network error conditions are documented, the product must generate descriptive and informative error messages.
- Errors and messages include the product's name and also include troubleshooting information such as why the message occurred and how to resolve it. (For example, Product_Name: Message [problem/resolution].)
- The product must not create a database comprised primarily of information contained in the standard eDirectory database (as initially created when eDirectory is installed).
- The application server (Service Provider) must execute completely and correctly from a workstation that does not initially have any logical connection to the server or workstation where the service resides.
- The application client (Service Utilizer) must execute completely and correctly from a workstation that does not initially have any logical connection to the server or workstation where the service resides.
- The product should treat the network as an extension of its native operating system, including treating the NetWare file/directory attributes in a manner consistent with the corresponding attributes in the native OS.
- All product executable files must be installed into directories where users are granted only RF (Read and File scan) trustee rights.
- The product should not allow file trustee rights to change upon file updates.
- If utilizing low level protocols directly, send data packets as large as possible and let the underlying protocols redefine it to smaller packets. (For example, the product should write large streams of data and let protocol break it down.)

Criteria for Java Applications

Manageability Criteria

The product should operate completely and correctly with the latest released version of the Novell implementation of the Java Virtual Machine (Novell JVM).

The product should allow any user (not just ADMIN or SUPERVISOR) with sufficient access rights or security equivalence to install the product, perform network administrator level functions, and to access privileged data.

The product should not interfere with the operation of other products on the network.

The product must not threaten the security of the network in any way.

The product must not significantly degrade overall performance of the network.

The product must correctly handle and display all dates between the years 1980 to 2035.

Network Resources Criteria

If the product provides printing capabilities, it must be able to print correctly to the default port.

The product should be able to fully access any relevant network peripheral devices.

If the product uses any of the following resources, they must be either user-definable or unique among all such resources used by other NetWare applications:

- Static IPX/SPX Socket Numbers.
- Service Advertising Protocol (SAP) IDs.
- NetWare Core Protocol (NCP) Extensions, IDs, and Names.
- Novell Directory Services (eDirectory) Schema Extensions Prefixes (NDS_PFX).
- Remote Method Invocation (RMI) Service Names.
- Common Object Request Broker (ORB) Architecture - Internet Inter-ORB Protocol (CORBA- IIOP) Service Names.

If the product uses IP ports, it must not prevent ADMIN or equivalent user from configuring and managing them. The product also must not hard code IP ports.

Certification Recommendations for Developers

DeveloperNet Labs is currently considering the following recommendations for inclusion in future versions of the certification test criteria:

- Access to the product must be manageable using ConsoleOne.
- Any eDirectory object type created or used by the product must correspond to the actual function performed by that object type.
- Where network error conditions are documented, the product must generate descriptive and informative error messages.
- Errors and messages must include the product's name. We recommend you also include troubleshooting information about why the message occurred and how to resolve it. (For example, Product_Name: Message [problem/resolution].)
- The product must not create a database comprised primarily of information contained in the standard eDirectory database (as initially created when eDirectory is installed).

Criteria for eDirectory Applications

The application must not create or use an object at the root of the tree that might cause security issues, network administration difficulties, or inconvenience to users that have insufficient rights to the root of the eDirectory tree. This must be accomplished without reducing the security level of the eDirectory root, or employing object specific security rights at the root of the tree.

You must be able to install the application in any container, not just Root.

The application's installer must require ADMIN rights only. It must not require the user to log in as ADMIN.

The application must not use bindery calls. eDirectory applications must use DS naming conventions exclusively.

The application must not perform unlimited scope searches of the eDirectory tree (i.e., NWDSSearch()). Searching of the entire tree causes unacceptable degradation of eDirectory performance.

The application must use a 'compare' and not a 'read' API whenever possible. This ensures that you are not forcing a reference to the main eDirectory tree and adversely affecting the performance of eDirectory. A local reference to a local replica is much faster.

The application must not poll DS objects. A registered event must be used to signal an attribute change. Polling has a negative effect on performance.

The eDirectory directory objects created by your application must be referenced and used by your application.

The application must not duplicate information (i.e., maintaining its own user list) already stored in eDirectory. It is better to leverage the data that already exists, improving the performance of your application and eDirectory.

The application must authenticate exclusively through eDirectory.

The application must use double-byte unicode for message tables, which provide certification with language support for users world-wide. This support is required even if your messages are in English.

The application must check the NWSLANGUAGE flag to set the code page.

Criteria for Novell Modular Authentication Services (NMAS) Applications

Installation Criteria

Hardware, software, and method setup should be detailed in the documentation distributed with the product.

If a ConsoleOne-based administration tool is provided, all product setup must be done in a sub-tab of the "Login Methods" tab.

Manageability Criteria

The method name should accurately describe your product's installed method.

The "grade" should accurately reflect the type(s) of authentication your method performs.

The method number must be the one provided to you by Novell.

The product's method properties sheet's support tab must direct the user to your company for customer support.

The user should be prompted for any actions or information required by the method while the login takes place.

The login method must properly authenticate to the eDirectory tree.

The login method should handle client timeouts in a robust manner.

Certification Recommendations for NMAS Developers

A ConsoleOne plug-in should be installed on a server volume and run from the client.

Criteria for NetWare Cluster Services

Installation

The product must either, automatically detect a clustered environment and install correctly, or provide specific instructions to users how to install their product in a NWCS environment.

Manageability

The product must be able to run and function properly on a TCP/IP only client/ server.

The product must not use Bindery API calls. NWCS servers support eDirectory only.

If the product reads or writes (error logs, data, etc.) to disk, it must be configurable to write to, Non SYS, (NWCS) volumes with high volume IDs (i.e. cluster volumes are 254, 253, etc.). This allows the multiple cluster servers to have one consistent log of what is happening.

If the product reads or writes to disk, it must connect to volume object via its IP address, eDirectory name, or DNS name.

The product must be able to migrate from one server to another server in the cluster, without data loss, when ConsoleOne moves the volume resource from one server to another.

The product must be able to migrate from one server to another server in the cluster, without data loss, when the "Cluster Leave" and "Cluster Join" commands are entered on the server's command line.

The product must be able to fail over from one server to another server in the cluster, without data loss, when the server it is running on ABENDs or is taken down.

The product must be able to automatically reconnect and recover from a cluster volume failing over (minimum of 8 seconds) to another server. This must work on both Windows 95/98 and Windows NT/2000 (TCP/IP only) workstations.

The product must not cause the cluster to failover (with its default failure detection threshold of eight seconds) because it is taking too much CPU time or staying in Real Mode too long accessing the floppy, etc.

The product must not cause the cluster to stall when leaving the cluster.

Recommendations for NWCS Developers

The product should work in both *Active/active* as well as *Active/passive* mode. *Active/Active* mode is when the product is running on more than one NWCS server so when a fail over occurs the copy of the product that is still running takes over (preferably where the other left off) for the one that failed over. *Active/passive* mode is when only one version of the product is running on the cluster servers and when fail over occurs, the product is started on another server and takes over.

The product should automatically detect a clustered environment and install correctly. Look for Clustering Services in the Product.dat file.

If the product does not automatically detect a clustered environment, a web site should be provided so updates can be released quickly, with specific instructions to users how to install and maintain their product in a NWCS environment.