

# InterWorx NFS Guide

by InterWorx LLC

# Contents

<b>1</b>	<b>Preface</b>	<b>2</b>
<b>2</b>	<b>Basic NFS Usage via InterWorx</b>	<b>3</b>
2.1	NFS Overview .....	3
2.2	Exports .....	3
2.3	Mounts .....	4
<b>3</b>	<b>Advanced via Command-Line</b>	<b>6</b>

# Chapter 1

## Preface

This guide is intended to introduce and instruct a server admin on use of the NFS features of the InterWorx Control Panel. It is designed to give a quick overview without going into too much depth.

In order to gain the most benefit from this guide, it is recommended that the reader familiarize themselves with the basics of NFS. NFS allows you to expose subtrees of the filesystem on an NFS server to an NFS client, which is then able to utilize the NFS share as if it were local to the client. More information about the system can be found online on the NFS Overview, FAQ, and howto documentation on SourceForge <http://nfs.sourceforge.net/>.

## Chapter 2

# Basic NFS Usage via InterWorx

The NFS feature is accessible under System Services ▸ NFS Server in NodeWorx. It is here that you can control basic aspects of the NFS service as well as mount NFS shares from other servers on the InterWorx server.

### 2.1 NFS Overview

The overview section allows you basic control over the NFS server daemon that runs on your server. On this page you can see a statistics graph of usage generated by RRDTool, as well as the service controls for the NFS service. Starting/Stopping/Restarting the service is possible here - in addition you can set the service to start on boot which is important if your server is hosting files to an NFS client on another machine. In addition, we display the NFS version that the local NFS server is capable of supporting.

### 2.2 Exports

This section is the interface to the `/etc/exports` file which controls access to your filesystem via NFS. To create an NFS export, you simply need the following information:

**Directory To Export** This is the part of the linux directory tree which be visible to an NFS client which mounts your share. Keep in mind that permissions are important - in that if directories are not at minimum read-execute for all users and files aren't at minimum read for all users, the machine you are sharing with may have difficulty utilizing the share effectively unless you have sync'd the UID's and GID's for the user/group that owns the directories and files in the share.

**IP Allow List** This is the list of hosts or alternatively, subnets that may utilize the share. You may specify hosts in the following manner:

- Single host - simply enter the IP address of the host you want to export to.
- FQDN's - such as `host.example.com`. You can also use wildcards with FQDN's such as `*.example.com` which will allow any host with reverse-dns setup to be inside the `example.com` second-level domain zone.
- Subnets - specify with CIDR notation like so: `192.168.1.0/24` which is essentially the same as saying any host in `192.168.1.1-192.168.1.254` range is permitted to mount the share. Wikipedia's article on CIDR notation does a good job of explaining the mechanics.<sup>1</sup>

**Read/Write Access** dictates if the hosts that mount this share are permitted to make changes to the filesystem. You want to generally limit who can make changes to the filesystem to very specific hosts that you know need access.

**Root User Access** determines if the root linux user on the NFS client system is allowed to behave as root when doing things on the NFS share, or if any actions they take will be mapped to the "nobody" linux user on the local NFS server filesystem.

---

<sup>1</sup>[http://en.wikipedia.org/wiki/CIDR\\_notation](http://en.wikipedia.org/wiki/CIDR_notation)

**Sync Options** determine how much care NFS takes in storing changes made by the client locally on the NFS server. Typically when a client makes a request, the NFS client has to send the request from the server, then the server has to write the data to local disk, then it has to notify the client that the change was successfully made. There are two options here that determine when the client is notified that the changes were made:

- *Synchronous* which says that a client will not get a response that the change has been made until the server has safely written the data disk - which basically means the hard disk acknowledges the data is safe on disk.
- *Asynchronous* says that the client will receive the OK as soon as the NFS server has scheduled the data to be updated on disk. Basically if the NFS server is receiving a bunch of requests to write changes to the share, it will maintain a queue of operations it needs to perform on the disk. It will try to re-arrange the operations such that they can be completed by the disk as fast as possible. In asynchronous mode, as soon as the operation is queued, the NFS server will notify the client that the change has been completed instead of when it is written to disk. This provides better performance as there is less delay waiting on disk. On the other hand, a crash or failure on the server side may lose data that has not been committed to disk with the client believing that it was. In other words, it's riskier, but faster

When you're done filling the form out on the Exports page, clicking add will modify the `/etc/exports` file and add the export entry. You can then view/edit/delete the entry in the "Exported NFS Shares" table.

## 2.3 Mounts

Mounts are your server acting as an NFS client to another NFS server elsewhere. This is you have data stored on a server elsewhere and you want it accessible locally. In order to mount a remote partition you need the following information:

**Host** Is obviously the IP or hostname of the machine that contains the NFS share you are trying to access. As discussed above in Exports, the remote machine must be permitting your server's IP or hostname to connect and mount the share or the mount will fail.

**Remote Directory** is the directory on the remote NFS server's filesystem that you wish to mount. This is necessary because the NFS server may be exporting multiple directories and you need to specify which one you want.

**Mount Point** is the directory on your local system where you wish to have the NFS mount "appear". Much like when inserting a USB thumb drive or adding a new hard disk on any linux system, you need to choose an existing directory to make the USB thumb driver or hard disk "appear" so you can access it. When mounting to a directory, any subdirectories or files in that directory become hidden and inaccessible while there is something mounted on that directory. You typically use an empty directory for a mount point.

**Protocol** determines which IP data transmission protocol the NFS server and client will use to communicate. Typically TCP is recommended because it is able to account for lost packets as well as packets arriving out of order. TCP is generally more widely used when it comes to data integrity. UDP is a lighter, faster protocol but probably shouldn't be used unless the server and client are connected via a secure highly reliable network - such as a crossover cable between the two machines.

**Mount Type** determines how the client reacts in the event of a failure. If a program is reading or writing from the share and the connection goes down for some reason, *hard* will force the program to wait until the server comes back online. *soft* on the other hand will report an error to the program and allow it to continue execution. *hard* functions in tandem with the Interrupt Handling option.

**Remount Options** are used to determine the behavior of the system to remount the share when there is an error. *Halt* indicates that the share should be dropped from the mount table should an error occur. *Continue* indicates that the share should remain mounted as it could be a temporary failure.

**Interrupt Handling** functions in combination with the *hard* mount type. When you have the hard mount type and the connection drops, allowing interrupts gives you the ability to interrupt the operation that is waiting on the connection to come back and allow the program to continue. Without allowing interrupts, processes that are

waiting on I/O on the NFS share will be unable to execute until the NFS export comes back online or the NFS client is restarted.

After you have entered the data for your NFS mount into the form, InterWorx will do a quick verification to ensure that the data you entered made sense, and then attempt to mount the device. If the mount fails, Interworx will notify you with an error.

## Chapter 3

# Advanced via Command-Line

This part of the guide is still under development. We highly recommend looking into CentOS/RedHat's NFS guide if you wish to dive down deeper in the system and configure NFS with more granularity. [http://www.centos.org/docs/5/html/Deployment\\_Guide-en-US/ch-nfs.html](http://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-nfs.html)