

Perforce Software, Inc.

510.864.7400

consulting@perforce.com

[www.perforce.com](http://www.perforce.com/)

SDP Administrator Guide
(2018.1)

(first draft)

Table of Contents

[1 UNIX Installation 3](#_Toc528663541)

[1.1 Prerequisites 3](#_Toc528663542)

[1.2 Installation 3](#_Toc528663543)

[1.3 Offline Checkpoints 4](#_Toc528663544)

[1.4 Existing SDP servers 5](#_Toc528663545)

# Overview

The SDP has four main components:

* Hardware and storage/filesystem layout recommendations for Perforce.
* Scripts to automate offline [checkpoints](http://www.perforce.com/perforce/doc.current/manuals/p4sag/02_backup.html#1045865) and other critical maintenance activities.
* Scripts to replicate the Perforce [journal](http://www.perforce.com/perforce/doc.current/manuals/p4sag/02_backup.html#1047307) to another volume or server.
* Scripts to assist with user account maintenance and other routine administration tasks.

Each of these components is covered in detail in this guide.

We expect you to check the SDP into a depot called *Perforce* as part of the installation process. The directory structure of the SDP:



## Configuring the Perforce Server

This section tells you how to configure a Perforce server machine and an instance of the Perforce Server. These topics are covered more fully in the System Administrator’s Guide and in the Knowledge Base. This chapter covers the details most relevant to the SDP.

The SDP can be installed on multiple server machines, and each server machine can host one or more Perforce server instances. (In this guide, the term server refers to a Perforce server instance unless otherwise specified.)

Each server instance is assigned a number. This guide uses instance number 1 in the example commands and procedures. Other instance numbers can be substituted as required.

This chapter also describes the general usage of SDP scripts and tools.

## Volume Layout and Hardware

To ensure maximum data integrity and performance, use three different physical volumes for each server instance. Three volumes can be used for all instances hosted on one server machine, but using three volumes per instance reduces the chance of hardware failure affecting more than one instance. It is possible, but not recommended, to put all the files onto a single physical volume. The SDP assumes (but does not require) the following three volumes:

1. Perforce metadata (database files): Use the fastest volume possible, ideally RAID 1+0 on a dedicated controller with the maximum cache available on it. This volume is normally called /metadata.
2. Journals and logs: Use a fast volume, ideally RAID 1+0 on its own controller with the standard amount of cache on it. This volume is normally called /logs. If a separate logs volume is not available, put the logs on the depotdata volume.
3. Depot data, archive files, scripts, and checkpoints: Use a large volume, with RAID 5 on its own controller with a standard amount of cache or a SAN or NAS volume. This volume is the only volume that must be backed up. The backup scripts place the metadata snapshots on this volume. This volume can be backed up to tape or another long term backup device. This volume is normally called /depotdata.



This diagram shows:

* a Perforce *application* administrator’s view of the system, including how to navigate the directory structure to find databases, log files, and versioned files in the depots.
* a Perforce *system* administrator’s view, including the physical volume where Perforce data is stored.

If three controllers are not available, put the logs and depotdata volumes on the same controller. Do not run anti-virus tools or back up tools against the metadata volume(s) or logs volume(s), because they can interfere with the operation of the Perforce server.

NOTE: Back up everything on the depotdata volume(s). Avoid backing up the metadata volume directly, because doing so can interfere with the operation of a live Perforce server, potentially corrupting data. The checkpoint and journal process archive the metadata on the depotdata volume. Backing up the logs volume is optional.

Both Unix and Windows installation of the SDP now use symlinks (on Windows this is via the mklink tool).

Logical and Physical View Mapping:



The links as <SYMLINKD> on a Windows installation:

Directory of c:\p4

20/06/2014 15:05 <DIR> .

20/06/2014 15:05 <DIR> ..

20/06/2014 15:05 <SYMLINKD> common [f:\p4\common]

20/06/2014 15:05 <SYMLINKD> config [f:\p4\config]

20/06/2014 15:05 <SYMLINKD> M1 [f:\p4\M1]

Directory of c:\p4\M1

20/06/2014 15:05 <DIR> .

20/06/2014 15:05 <DIR> ..

20/06/2014 15:05 <DIR> bin

20/06/2014 15:05 <DIR> checkpoints

20/06/2014 15:05 <DIR> depots

20/06/2014 15:05 <SYMLINKD> logs [g:\p4\M1\logs]

20/06/2014 15:05 <SYMLINKD> offline\_db [e:\p4\M1\offline\_db]

20/06/2014 15:05 <SYMLINKD> root [e:\p4\M1\root]

20/06/2014 15:05 <DIR> ssl

20/06/2014 15:05 <DIR> tmp

## Instance Names

Traditionally the SDP has used integers for instance names which show up in the paths above, for example C:\p4\***1***\root.

However it is increasingly the case that alphanumeric names are used for instances, e.g. C:\p4\***Acme***\root. Commonly organizations strive to use a single Perforce instance, one logical data set, which may be replicated around the globe. Using a single instance optimize colloboration and simplifies code access for all development activity. When there is a single instance, the name ‘***1***’ is as good as any. When there is more than one instance, e.g. if there are isolated silos of development activity, an alaphnumeric name may be more helpful than an integer for identifying the data set, such as ***Acme*** or perhaps ***LegacyApps***. Another instance is sometimes to develop and test things like Perforce trigger scripts before rolling them out to the live production instance, or to provide a standing internal training data set.

In any case, it is worth thinking and planning your naming, particularly if you have multiple instances including replicas of different types and these are located on different hosts.

If you are using instance numbers, then an example configuration where there are 2 master server instances, each with a replica, might be:

|  |  |  |
| --- | --- | --- |
| **Server hostname** | **Instance ID** | **Port** |
| p4d-sfo-01 | 1 | 1666 |
| sfo-p4d-01 | 2 | 2666 |
| sfo-p4d-02 | 1 | 1666 |
| sfo-p4d-02 | 2 | 2666 |

For consistency, instances with same ID should refer to the same logical data set, they just run on different machines.

Alternatively, alphanumeric names can be clearer and easier:

|  |  |  |  |
| --- | --- | --- | --- |
| **Server hostname** | **Instance ID** |  | **Port** |
| sfo-p4d-01 | Acme |  | 5000 |
| sfo-p4d-01 | Test |  | 5999 |
| sfo-p4d-02 | Acme |  | 5000 |
| sfo-p4d-02 | Test |  | 5999 |

Some sites apply a convention to the port number to identify whether the P4PORT value is that of a master server, a broker, replica, edge server, or proxy. In such cases the first digit is reserved to identify the instance, and the remaining 3 digits identify the target service, e.g. 666 for a broker, 999 for a master server, 668 for a proxy.

Host naming conventions vary from site to site, and often have local naming preferences or constraints. These examples the the code of the nearest major airport, sfo in this case, as a location code. Using location in the hostname is merely an example of a site preference, not necessarily a best practice.

End user P4PORT values typically do not reference the actual machine names. Instead they reference an alias, e.g. perforce or sfo-p4d (without the -01). This helps make failover operations more transparent.

## Memory and CPU

Maximum performance is obtained if the server has enough memory to keep all of the database files in memory. Make sure the server has enough memory to cache the **db.rev** database file and to prevent the server from paging during user queries.

**Guidelines for** allocating memory:

## 1.5 KB of RAM per file stored in the server.

* 32 MB of RAM per user.
* Use the fastest processors available with the fastest available bus speed. Faster processors with a lower number of cores provide better performance for Perforce. Quick bursts of computational speed are more important to Perforce's performance than the number of processors, but have a minimum of two processors so that the offline checkpoint and back up processes do not interfere with your Perforce server.

## General SDP Usage

This section presents an overview of the SDP scripts and tools. Details about the specific scripts are provided in later sections.

Most tools reside in c:\p4\common\bin. The directory c:\p4\**instance**\bin contains scripts and executables that are specific to a server instance, such as the p4.exe client. The scripts in c:\p4\**instance**\bin generally set the environment for an instance correctly, then invoke the corresponding script in c:\p4\common\bin.

Run important administrative commands using the scripts in c:\p4\**instance**\bin, when available. Then, use the p4.exe executable located in c:\p4\**instance**\bin.

Usage examples for instance 1 or instance master:

|  |  |
| --- | --- |
| c:\p4\common\bin\live-checkpoint.ps1 1 | Take a checkpoint of the live database on instance 1 |
| c:\p4\common\bin\daily-backup.ps1 master | A daily checkpoint of the master instance. |

### Monitoring SDP activities

The important SDP maintenance and backup scripts generate email notifications when they complete.

For further monitoring, you can consider options such as:

* Making the SDP log files available via a password protected HTTP server.
* Directing the SDP notification emails to an automated system that interprets the logs.

# UNIX Installation

## Prerequisites

1. Two identical database volumes – local SSD recommended.
	1. One for root - /hxdb1
	2. One for offline checkpoints and database switching - /hxdb2
	3. Volume for the depot files - /hxdepots
	4. Volume for the journal and logs - /hxlogs
2. User to run the perforce service under.
3. Perforce super user account and password for management. (Something like p4admin for example. Not an employee's account.)
4. Choose (new server) or check (existing server) the case sensitivity of the Perforce server.
5. /p4 top level folder - (Only required ahead of time if root access to the machine is not available during the install.)

## Installation

1. Download the tgz and place it on /hxdepots, then extract it with tar xvzf sdp.Unix.<version>.tgz

This will create a directory named sdp.
2. run chown -R perforce:perforce sdp\* (Or whatever your OS user’s name and group is.)
3. cd sdp/Server/Unix/setup
4. vi mkdirs.sh
5. Makes sure that the configuration section variables are set correctly for  your server.
6. Add the primary admin's email to the MAILTO configurable.
7. Save mkdirs.sh
8. Put a copy of your server's version of p4 and p4d in sdp/Server/Unix/p4/common/bin.
9. Check to make sure that a file or folder named "p4" does not exist in the top level of any of the install folders.
10. As root, run:

cd sdp/Server/Unix/setup

./mkdirs.sh <instance\_name>

ie: ./mkdirs.sh fifa

See note at the bottom of the page for servers with an existing SDP install.

1. Stop your server
2. Move the production db.\* to /p4/<instance\_name>/root
3. Move the license to /p4/<instance\_name>/root
4. Move the journal and log files to /p4/<instance\_name>/logs
5. Move any depots that do not have hard coded map fields to /p4/<instance\_name> /depots
6. Make sure that you have moved everything to the correct location.
7. Run:

cp /p4/<instance\_name>/bin/p4d\_<instance\_name>\_init /etc/init.d

cd /etc/init.d

chkconfig --add pd\_<instance\_name>\_init

chkconfig p4d\_<instance\_name>\_init on

(make sure all files are owned by the perforce user at this point)

sudo su - perforce

/p4/<instance\_name>/bin/p4d\_<instance\_name> -r /p4/<instance\_name>/root -J /p4/<instance\_name>/logs/journal " -cset server.depot.root=/p4/<instance\_name>/depots"

/etc/init.d/p4d\_<instance\_name>\_init start

1. Set up your environment to be able to connect to your server.
2. Run:

p4 configure set journalPrefix=/p4/<instance\_name>/checkpoints/p4\_<instance\_name>

1. Set any other configurables from /p4/sdp/setup/configure\_new\_server.sh that you want to use for your server.

You can edit the configure\_new\_server.sh script and run it as ./configure\_new\_server.sh <instance\_name>
2. cd /p4
3. vi p4.crontab (p4.crontab.edge for an edge server or p4.crontab.replica for a replica server)
4. Change the instance=1 to instance=<instance\_name>
5. Check the times for running the scripts and update as necessary. Comment out any that you don't want to run and save the file.
6. Run "crontab p4.crontab" to load the new crontab.
7. Run some checks to make sure everything looks like it should: "p4 info", "p4 depots", etc.

## Offline Checkpoints

1. If you have an offline set of db files, move those into /p4/<instance\_name>/offline\_db.

If you do not, then recover the most recent checkpoint into the offline\_db folder.
2. Run

"touch /p4/<instance\_name>/offline\_db/offline\_db\_usable.txt"

1. Test the offline checkpoint process:

/p4/common/bin/daily\_checkpoint.sh <instance\_name>

1. When it finishes, you should get an email containing the contents of /p4/<instance\_name>/logs/checkpoint.log

## Existing SDP servers

On a server where an older version of the SDP exists:

1. Stop the server.
2. Go to each volume where a p4 directory exists and rename it to p4.orig
3. Remove the links in /p4
4. Rename the p4.crontab\* files.
5. Update mkdirs.sh
6. Install the new sdp
7. mv the data, triggers, and any custom scripts from the p4.orig directories to the newly installed p4 directories.
8. Restart the server.
9. Test to make sure everything is okay, and if so, you can remove the p4.orig directories.
10. Update the crontab to use any newly named scripts per /p4/p4.crontab, p4.crontab.replica, or p4.crontab.edge, depending on the type of server you are upgrading.