

## TP2 Packaging Installation Guide

This document covers the of installation of the TP2 software on a UNIX-like environment. It mentions any software dependencies and initial configuration that might be necessary. It finally touches on how TP2 itself can be upgraded if already installed.

Version	Date	Author	Changes
1.0		Simon Edwards	Original version
1.1	October 2009	Simon Edwards	Various updates and improvements, including recent optional Sqlite support.

## Introduction

### ***Purpose of Document***

TP2 is a packaging system designed for UNIX environments. It is currently in use on various platforms including Linux, AIX, Solaris and HP-UX. Using the toolset on other platforms should not present any major issues; since it is written to use reasonably few dependencies and as much standard software as possible. This document describes how the software can be installed – typically it takes just a few minutes.

### ***Why Install TP2 in the first place?***

Of course there are many packaging toolsets already available for UNIX, so why use TP2? A summary of some of the features illustrate why:

- Easy to use – package creation, installation, removal and checking are very easy.
- Namespace Support – once the toolset is installed the administrator can set-up different “namespaces” owned by different users – allowing normal users to deploy packages into their own areas.
- Preview Installation – it is possible to run a preview-only installation which will indicate which files will be replaced by the package – it only installs files that change.
- Detailed logging – the log file includes per-file system impact assessment information, script output, installed, changed and replaced file counts.
- Atomic Package Installations – a package will install or it will not. An interrupted package installation will be completed or rolled back from the command line or on machine restart if necessary.
- Fully dependency Handling – a package can have a series of packages that must be installed for it to be used, or packages that must not be installed at the same time as itself. Package installation will draw in any necessary dependencies as required.
- DSA Signing Support - optionally packages can be signed and optional or enforced package signing can be defined on a per-namespace basis.
- Audit log support / enhanced user control in a secure manner.

Of course there are other features but hopefully the above list is enough to get you interested! The full list of available features can be found in the TP2 Administration Guide.

### ***Getting TP2***

TP2 is available from the following URL:

<http://www.linuxha.net/skulker>

As the address indicates it forms part of the packages available from the authors of Linuxha.net – a Linux high availability clustering product. The “downloads” link provides access to the latest releases – either as a native package for your particular platform or a TP2 package itself.

Descriptions for installations of native packages is not included here; administrators are expected to be able to install such packages. However since TP2 attempts to span all UNIX-compatible environments installation directly from a TP2 package on to *any* UNIX environment is now described once dependencies have been resolved.

## Software Dependencies

Although there are several dependencies for the toolset they are common ones; many may be already installed whilst those not are likely to be available via standard packages or an common install path.

The TP2 software is written in Perl - and modern version from 5.6 or above should work without problems. To check the version of Perl installed use the following command:

```
# perl -v

This is perl, v5.8.4 built for i86pc-solaris-64int
(with 32 registered patches, see perl -V for more detail)
```

Once Perl is available a series of Perl modules are required, as described by the following table:

Module	Purpose
XML::Parser	
Crypt::CBC	
Crypt::Blowfish	
Digest::SHA1	
Crypt::OpenSSL::DSA	This is used for the creation and validation of signatures on packages. This is an optional feature and so it is not necessary to install this software unless package signing support is required.
DBI	Optional - this is required if you wish to make use of Sqlite meta-data repositories. This will reduce file system usage without impacting performance significantly.
DBD::Sqlite::Amalgamation	A DBD driver for Sqlite. This is the simplest way of installation Sqlite and the DBD interface. Alternatively DBD::Sqlite is fine too.

Of course if the native package manager is missing any of the above modules the best way of getting them is via cpan, for example:

Of course the first step might be getting CPAN itself! In this case grab the latest download fomr:

<http://search.cpan.org/dist/CPAN/lib/CPAN.pm>

Copy download to a temporary directory and then as "root" enter the following commands:

```
# perl Makefile.PL
# make
# make install
```

The above commands assume you have "make" available - this definitely should be installed or available as a native package!

Note: For many of the Perl modules above a 'c' compiler will be necessary - so please ensure one of those is installed - typically "gcc".

If CPAN was not available and had to be installed the first step is to configure it using the following command:

```
# perl -MCPAN -e shell
```

Follow the instructions; in most cases the defaults to questions are sensible - but still read them.

Now you can start CPAN and install the software you are missing, for example:

```
# perl -MCPAN -e shell
cpan[2]> install Digest::SHA1
cpan[3]> install Crypt::OpenSSL::DSA
```

Once all necessary software has been installed the following three commands should work without error:

```
# perl -MDigest::SHA1 -e 1
# perl -MXML::Parser -e 1
# perl -MCrypt::OpenSSL::DSA -e 1
```

### ***Initial TP2 Installation***

Once a TP2 package has been downloaded, copy to the "/tmp" directory, for example:

```
# cp tp2+1.1.0.tp2 /tmp
```

Once the file has been copied to "/tmp" a temporary directory should be created and the contents of the "tp2" package extracted. This is straightforward:

Assuming the file in question ("tp2+0.7.6.tp2" in this case) has been copied across to "/tmp" on the target host, a temporary directory should be created and the tools extracted into this directory:

```
# mkdir /tmp/tp2code
# cd /tmp/tp2code
# gunzip -c /tmp/tp2+1.1.0.tp2 | tar xf -
```

At this point installation is simply a matter of using the tools themselves to configure everything. Firstly change into the bin directory which has just been extracted and make all programs executable:

```
# cd tp2/bin
# chmod 555 *
```

At this point a utility to create the initial configuration files and directories must be run. To do this run the following commands:

```
# ./tp2make_env
```

## TP2: Installation Guide

The above commands should be entered as above – that is running them from the “bin” directory. When run the output generated should be:

```
Created directory /var/adm/tp2
Created directory /var/adm/tp2/ns
Created directory /var/adm/tp2/security
Created directory /var/adm/tp2/ns/root
Created directory /var/adm/tp2/ns/root/meta
Created directory /var/adm/tp2/ns/root/pkg
Created directory /var/adm/tp2/ns/root/log
Created directory /var/adm/tp2/ns/root/spool
Created directory /var/adm/tp2/ns/root/repackage
Created "root" namespace configuration file.
```

Note: No output is given if the above command is not necessary.

By default if Sqlite is available it will be made use of. This can be confirmed by checking to see if the following file exists:

```
/var/adm/tp2/ns/root/meta/meta.db
```

Finding this file means that Sqlite is available and has been used. If you wish not to use it then delete the “meta.db” file before continuing:

```
# rm /var/adm/tp2/ns/root/meta/meta.db
```

## Installing TP2

The typical location for the installation of the TP2 software is:

```
/opt/tp2
```

To install the [tp2] package that has just been downloaded the following commands, again from the "/tmp/tp2code/bin" directory should be run.

```
# mkdir /opt
# ./tp2make_ns --name opt --root /opt --owner root --verbose
```

This would generate the following output:

```
Log : Created directory /var/adm/tp2/ns/tp2_ns.
Log : Created directory /var/adm/tp2/ns/tp2_ns/meta.
Log : Created directory /var/adm/tp2/ns/tp2_ns/pkg.
Log : Created directory /var/adm/tp2/ns/tp2_ns/log.
Log : Created directory /var/adm/tp2/ns/tp2_ns/spool.
Log : Created directory /var/adm/tp2/ns/opt/repackage.
Log : Created "tp2_ns" namespace configuration file.
```

At this point the "tp2" package itself can be installed using the following commands:

```
# ./tp2repos --repos /tmp --verbose
Log : Package file "tp2+1.1.0.tp2":
Log :   Extracting configuration... Done.
Log :   Estimating package size... Done.
Log :   Generating package checksum... Done.
# ./tp2install --namespace opt --pkg tp2 --repos /tmp --verbose
```

Since the "--verbose" option has not been specified in the above commands no output should appear.

## The TP2 Daemon

Although this is an optional feature it is strongly recommended that it be installed. It provides a central audit log of important operations and also provides non-root users the ability to perform certain functions if the administrator allows them.

It ensure that the daemon is started on a reboot depends on your operating system and/or distribution. A few examples are given here, though an exhaustive set of installation instructions covering every platform is not possible.

### Open Solaris (10 and above)

This OS uses a service configuration repository for storing information on services. The TP2 software comes with a value service description that can be added using the commands:

```
# cd /opt/tp2/etc
# svccfg import -V tp2daemon-svc.xml
```

This can then be seen using:

```
# svcs tp2
STATE          STIME      FMRI
disabled      22:48:49  svc:/network/tp2:default
```

And to enable it:

```
# svcadm enable tp2
# svcs tp2
STATE          STIME      FMRI
online         22:58:04  svc:/network/tp2:default
```

## Init-Based Systems

Although more UNIX-like operating systems are moving away from the traditional "init" based run-control start-up they often return those facilities for backwards compatibility and so this section may cover a majority of systems

[Describe init-based installs here](#)

## Using TP2

Once installation is complete you probably want to read the document "TP2 : Administration Guide" - which covers how TP2 is typically used to install and manage software.