

# Installation notes

This document explains how to install your copy of the Open Systems Portability Checker (**OSPC**) on your host platform.

Installation consists of **taring** the contents of the distribution tape into a work directory and running the **doinstall** script to configure the various tools.

## 1.1 Before doing the installation

Before running the installation script please make sure that the following points have been addressed.

- 1 Installation password. Make sure that you have the installation password to hand.
- 2 Disk space. Make sure that there is at least 40M byte of free disk space. The **df** command can be used to find out the space available on each disk partition.
- 3 Permissions. Check that you have sufficient permissions to copy the software to the directory in which it is to reside.

## 1.2 Copying the tape to disc

The steps are as follows:

- 1 Create a working directory <OSPC-dir>.
- 2 **cd** to <OSPC-dir>.
- 3 Extract the software from the tape using the command:

**tar xvf <tape-device>**

Common names for <tape-device> are:

<b>Sun4</b>	<b>/dev/rst8</b>
<b>RISC System/6000</b>	<b>/dev/rmt0</b>
<b>HP</b>	<b>/dev/update.src</b>
<b>Linux</b>	<b>/dev/st0</b>
<b>Motorola 88000</b>	<b>/dev/rmt8</b>
<b>DecStation</b>	<b>/dev/rmt0h</b>

While **tar** is extracting the tape contents, the names of the files being extracted will be displayed.

## 1.3 Contents of the distribution tape

Once the tape has been copied to disc the directory `checker` should exist. Within `checker` the following directories and files should exist.

### 1.3.1 machine.binary/bin

This directory contains executables of each of the tools and their associated support files. Binaries for different architectures may be found in sub directories of the appropriate name.

<b>mcc</b>	<b>- The OSPC source code checker</b>
<b>mcl</b>	<b>- The OSPC cross unit checker</b>
<b>iddb</b>	<b>- The API database compiler</b>
<b>ccc</b>	<b>- A script for use in make files</b>
<b>c89</b>	<b>- A script that implements the POSIX.2 functionality</b>
<b>arr</b>	<b>- A script for use in make files</b>
<b>profadm</b>	<b>- A script for maintaining profiles</b>
<b>errorange</b>	<b>- A script for listing the error numbers currently used</b>
<b>errormsg</b>	<b>- A script for listing the error messages associated with a given error number</b>
<b>checkinfo</b>	<b>- A directory containing platform profile and standards information</b>

### 1.3.2 document

This directory contains documentation relating to individual tools, software engineering and coding standards. The file `document/CHANGES` contains information on changes between the current release of the software and previous releases.

### 1.3.3 examples

This directory contains example programs. Each example is contained in its own subdirectory.

#### 1.3.4 include

This directory contains all of the standard C header files, plus a few of the Posix header files.

#### 1.3.5 lib

This directory contains all of the standard C library functions, in compiled, .kic, form.

#### 1.3.6 man

Manual pages for **mcc**, **mcl** and the various other tools are held here.

#### 1.3.7 README

This file contains up to date information that may not be present in the printed documentation.

### 1.4 Installing the software

#### 1.4.1 Overview

The software has been designed to be as self configuring as possible. The main installation task is deciding what setup is best suited to the development environment on the host platform.

If the man pages are required, the installation will need to be done by a user with sufficient permission to write to the appropriate system directory.

As distributed the files making up the **OSPC** are extracted from tape into a directory called `checker`. The installation is performed from this directory. The user may elect to leave the files in this directory and reference them from elsewhere, or to distribute them to other directories.

#### 1.4.2 Doing the installation

Once it has been copied to the hard disc **OSPC** is installed by running the **doinstall** script. This is stored in the `checker` directory, created when **tar** extracted the contents of the tape.

The installation script will ask a series of questions. The answers to some questions have default answers (the value of these defaults is placed in square brackets, [], at the end of the question).

First, a list of possible host processors is displayed and the user asked to select one of them. The distribution tape includes executables for all of the listed processors. The binaries for

the selected processor are copied to the appropriate working directory prior to being copied (or linked) to the final installation directory.

**Do you want to install OSPC by adding it to the PATH [y] :**

The executables for **OSPC** can be installed in one of three ways:

- 1 Add a path to the environment variable *PATH* describing how the files copied from tape can be located.
- 2 Create symbolic links in the default bin directory for each of the files in the directory <OSPC-dir>/bin.
- 3 Move the contents of the <OSPC-dir>/bin directory to the directory used to hold all applications software (for which the access path has already been specified). Note that all files in <OSPC-dir>/bin must be copied.

If symbolic links are to be added to the default executable directory you will be asked which directory they should be added to.

**Are system headers files kept in /usr/include [y] :**

By default the **OSPC** uses the header files included with the system when compiling a program. This option specifies where the system headers are located.

**Place man pages in /usr/man/man1 (in nroff format) [y]:**

The man pages may be installed in either nroff or plain text form. The directory they are being copied to should also be specified. The user id used to install **OSPC** should have sufficient permissions to write to this directory.

**Are you sure you want to install OSPC [n] :**

### 1.4.3 Locating the configuration file

All tools use the same method for locating any files associated with them. The tools assume that these files can be found along the same path as they themselves are located. That is they interrogate the system to find out the path used to located themselves.

The file checker/bin/checkinfo/host/locate (created during the installation process) contains the default paths along which **mcc** searches for include files and **mcl** searches for the library file.

## 1.5 Checking the installation

The following procedure will check that the **OSPC** has been correctly installed.

Issue the following command (at the Unix shell prompt):

```
% mcc
```

this should cause the **mcc** help screen to appear, and issuing the command:

```
% mcl
```

should cause the **mcl** help screen to appear.

In each case the help text should include half a dozen, or more options. If only a two line summary of the command line syntax appear, then go back to the installation notes.

**OSPC** is licensed on a per seat basis. Before running each tool checks to ensure that no more than the permitted number of users are already running. If the maximum number of users has been reached a message will be displayed and the tool aborted. The user will then have to wait for a slot to become free.

If running **mcc** causes the message 'Bad password' to appear check that the installation password given in response to the password query matched the one given to you by Knowledge Software, or the local agent. This password can also be found in the file `bin/checkinfo/password/password`.

### 1.5.1 Checking system resources

On some systems the maximum number of files that a user may have open at the same time is relatively low. To check that there are enough resources available to you to open all the files required by **OSPC** issue the following command (at the Unix shell prompt):

```
% mcc blah.c
```

Where `blah.c` is a small C program that `#include's` a number of files.

If a series of warnings about being unable to open various files appears then you have fallen foul of some preset limit.

On Unix systems the **ulimit** command (under the shells `sh`, `ksh`, `bash` and some `csh`'s) can be used to change various user limits. The `csh` shell has the **limit** command. Setting the number of file descriptors to 100 ought to be more than sufficient (the actual number required depends on the number of subprofiles being used).

### 1.5.2 Modifying the installation

Before modifying the installation it is recommended that the User Reference Manual be studied.

## 1.6 Uninstalling

**cd** to the directory <OSPC-dir>, and run the **uninstall** script. This will cause all of the files associated with **OSPC** to be deleted.

## 1.7 Contacts

Knowledge Software Ltd. Tel: +(44) 1252-520667

e-mail: [support@knosof.co.uk](mailto:support@knosof.co.uk)

URL <http://www.knosof.co.uk>