The owd-euses Querying Tool

Placement Interview Presentation

Oliver Dixon <od641@york.ac.uk>

Department of Computer Science, University of York

Department of Mathematics, University of York

27th September 2023

Overview of the Environment: Gentoo Linux

- Gentoo Linux is a source-based distribution of the Linux operating system; such distributions typically include package managers that encourage, or mandate, the local compilation of most packages. This stands in contrast to the previously established model of downloading and installing pre-built binaries.

Overview of the Environment: Gentoo Linux

- Gentoo Linux is a source-based distribution of the Linux operating system; such distributions typically include package managers that encourage, or mandate, the local compilation of most packages. This stands in contrast to the previously established model of downloading and installing pre-built binaries.
- A source-based distribution model has a number of advantages for power users, namely the ability to enable and disable compile-time features as desired, resulting in a more compact and suitably equipped binary.

Overview of the Environment: Gentoo Linux

- Gentoo Linux is a source-based distribution of the Linux operating system; such distributions typically include package managers that encourage, or mandate, the local compilation of most packages. This stands in contrast to the previously established model of downloading and installing pre-built binaries.
- A source-based distribution model has a number of advantages for power users, namely the ability to enable and disable compile-time features as desired, resulting in a more compact and suitably equipped binary.
- With such extensive customisability, a powerful querying system is required to determine the various features that are supported by each of the available packages.

Existing Solutions

Due to the age and popularity of Gentoo Linux, a multitude of querying tools exist for this purpose:

Existing Solutions

Due to the age and popularity of Gentoo Linux, a multitude of querying tools exist for this purpose:

- *eix*: very powerful and well-established, but requires building and maintaining a centralised binary cache of every installable package.

Existing Solutions

Due to the age and popularity of Gentoo Linux, a multitude of querying tools exist for this purpose:

- *eix*: very powerful and well-established, but requires building and maintaining a centralised binary cache of every installable package.
- *euses*: one of the original cornerstones of the Gentoo toolset, but contains bugs, error-prone and non-portable code, and is incompatible with newer versions of Gentoo.

Existing Solutions

Due to the age and popularity of Gentoo Linux, a multitude of querying tools exist for this purpose:

- *eix*: very powerful and well-established, but requires building and maintaining a centralised binary cache of every installable package.
- *euses*: one of the original cornerstones of the Gentoo toolset, but contains bugs, error-prone and non-portable code, and is incompatible with newer versions of Gentoo.
- *portageq*: the standard Gentoo querying tool; fully compliant, but proves to be very non-performant on large software repositories due to its antiquated Python foundations.

Project Requirements

`owd-euses` was devised in an effort to unify the advantages of these existing solutions while eliminating the problems relating to safety, incompatibility, and performance-preserving scalability. The final tool must possess certain critical attributes:

Project Requirements

`owd-euses` was devised in an effort to unify the advantages of these existing solutions while eliminating the problems relating to safety, incompatibility, and performance-preserving scalability. The final tool must possess certain critical attributes:

1. be fully standards-compliant with the Gentoo *Package Manager Specification* (PMS) and sufficiently customisable to work on future revisions;

Project Requirements

`owd-euses` was devised in an effort to unify the advantages of these existing solutions while eliminating the problems relating to safety, incompatibility, and performance-preserving scalability. The final tool must possess certain critical attributes:

1. be fully standards-compliant with the Gentoo *Package Manager Specification* (PMS) and sufficiently customisable to work on future revisions;
2. operate on existing (plain text) files in the established directory structures, instead of generating and periodically maintaining binary caches;

Project Requirements

`owd-euses` was devised in an effort to unify the advantages of these existing solutions while eliminating the problems relating to safety, incompatibility, and performance-preserving scalability. The final tool must possess certain critical attributes:

1. be fully standards-compliant with the Gentoo *Package Manager Specification* (PMS) and sufficiently customisable to work on future revisions;
2. operate on existing (plain text) files in the established directory structures, instead of generating and periodically maintaining binary caches;
3. seamlessly scale on very large software repositories;

Project Requirements

`owd-euses` was devised in an effort to unify the advantages of these existing solutions while eliminating the problems relating to safety, incompatibility, and performance-preserving scalability. The final tool must possess certain critical attributes:

1. be fully standards-compliant with the Gentoo *Package Manager Specification* (PMS) and sufficiently customisable to work on future revisions;
2. operate on existing (plain text) files in the established directory structures, instead of generating and periodically maintaining binary caches;
3. seamlessly scale on very large software repositories;
4. be *correct*: undergoing rigorous unit-testing, stress-testing under hostile environments, and never entering an undefined or non-deterministic state.

Details of the Implementation and Source Structure

Written in C99, `owd-euses` was intended to run on POSIX-compliant Gentoo/PMS-conforming UNIX-like systems. Given the four major design principles, the codebase is very compact, currently standing at slightly over 2000 lines. The source is clearly sectioned into distinct functions and well-abstracted with regard to near-complete separation of code and data.

Details of the Implementation and Source Structure

Written in C99, `owd-euses` was intended to run on POSIX-compliant Gentoo/PMS-conforming UNIX-like systems. Given the four major design principles, the codebase is very compact, currently standing at slightly over 2000 lines. The source is clearly sectioned into distinct functions and well-abstracted with regard to near-complete separation of code and data.

Translation Unit(s)	**Purpose**
`euses.{c,h}`	Provide the core parsing and searching routines
`globbing.{c,h}`	Utilise POSIX regex support
`args.{c,h}`	Parse and manage command-line arguments
`stack.{c,h}`	Implement the standard stack ADT
`converse.{c,h}`	Manage status- and error-reporting
`colour.h`	Provide helpers for ANSI text-colouring

Source Preview: Stack Interface (1)

```c
/**
 * @file
 * @brief The stack ADT public interface, supporting typical FIFO operations.
 * @author Oliver Dixon
 * @date 20/09/2021
 */

#ifndef STACK_H
#define STACK_H

#include <stddef.h>

/**
 * @struct stack
 * @brief The base stack data-type
 */
struct stack;

/**
 * Initialises an empty stack with the given initial capacity.
 * @param capacity the initial capacity
 * @return the newly created stack, or NULL on error
 */
struct stack * stack_init ( size_t capacity );

/**
 * Destructs the stack, freeing all memory allocated by the initialisation and
 * subsequent resizes.
 * @param self the stack to destruct
 */
```

Source Preview: Stack Interface (2)

```
31  void stack_free ( struct stack * self );
32
33  /**
34   * Pushes a reference to the node at the given address to the stack.
35   * @param self the stack to which the node must be pushed
36   * @param node the node to push
37   * @return the newly pushed node address, or NULL if the stack capacity is
38   *  insufficient and can not be resized.
39   */
40  void * stack_push ( struct stack * self, void * node );
41
42  /**
43   * Peeks at the node at the top of the stack.
44   * @param self the stack to peek
45   * @return the node atop the stack, or NULL if the stack is empty
46   */
47  void * stack_peek ( struct stack * self );
48
49  /**
50   * Pops the node at the top of the stack.
51   * @param self the stack to pop
52   * @return the popped node
53   */
54  void * stack_pop ( struct stack * self );
55
56  /**
57   * Prints the stack contents to stdout.
58   * @param self the stack to print
59   */
60  void stack_print ( struct stack * self );
```

Source Preview: Stack Implementation (1)

```c
/**
 * @file
 * @brief The stack ADT implementation
 * @author Oliver Dixon
 * @date 20/09/2021
 */

#include <stdlib.h>
#include <stdio.h>

#include "stack.h"

struct stack {
    void ** data;    /**< The stack storage array */
    size_t capacity; /**< The maximum number of items storable by the stack */
    size_t size;     /**< The current number of items stored on the stack */
};

struct stack * stack_init ( size_t capacity )
{
    const size_t DEFAULT_CAPACITY = 8;
    struct stack * self = NULL;

    if ( !capacity )
        capacity = DEFAULT_CAPACITY;

    if ( ( self = malloc ( sizeof ( struct stack ) ) ) )
        if ( ( self->data = malloc ( sizeof ( void * ) * capacity ) ) ) {
            self->size = 0;
            self->capacity = capacity;
```

Source Preview: Stack Implementation (2)

```
31          } else {
32              free ( self );
33              self = NULL;
34          }
35
36      return self;
37  }
38
39  void stack_free ( struct stack * self )
40  {
41      free ( self->data );
42      free ( self );
43  }
44
45  void * stack_push ( struct stack * self, void * node )
46  {
47      void ** new_data;
48
49      if ( self->capacity <= self->size )
50          /* If there is insufficient capacity on the stack, then we attempt to
51           * double the data allowance and continue with the push. */
52          if ( ( new_data = realloc ( self->data,
53                  sizeof ( self->capacity << 1 ) ) ) == NULL ) {
54              self->data = new_data;
55              self->capacity <<= 1;
56          } else
57              /* The stack capacity was insufficient and could not be resized. */
58              return NULL;
59
60      self->data [ self->size++ ] = node;
```

Source Preview: Stack Implementation (3)

```
61      return node;
62  }
63
64  void * stack_peek ( struct stack * self )
65  {
66      return self->data [ self->size - 1 ];
67  }
68
69  void * stack_pop ( struct stack * self )
70  {
71      return ( self->size ) ? self->data [ self->size-- ] : NULL;
72  }
73
74  void stack_print ( struct stack * self )
75  {
76      const size_t size = self->size;
77
78      for ( size_t i = 0; i < size; i++ )
79          printf ( "#%zu: %p\n", i, self->data [ i ] );
80  }
81
```

Abridged Source Preview: Argument Interface (1)

```
1  /**
2   * @file
3   * @brief The public argument-processing and -management API
4   * @author Oliver Dixon
5   * @date 25/09/2021
6   */
7
8  #ifndef ARGS_H
9  #define ARGS_H
10
11 #include <stdint.h>
12 #include <stdbool.h>
13
14 /** The data-type to hold the argument-toggle bit string. This can be checked to
15  * be sufficient with an assertion of the following form, where `ARG_TAIL` is
16  * the maximum number of arguments to be considered in any single program
17  * invocation: `assert ( sizeof ( args_t ) << 3 >= ARG_TAIL - 1 )`. */
18 typedef uint16_t args_t;
19
20 /** The argument index, storing each distinct argument that is recognised. */
21 enum arg_idx {
22     /** Special argument flag: the unknown argument, used by the processors. */
23     ARG_UNKNOWN,
24
25     ARG_PRINT_REPO_NAMES,
26         /**< Print the repository responsible for each match, in the form
27          * "[repo name]::[match text]". */
28     ARG_PRINT_REPO_PATHS,
29         /**< Print the repository name, and its corresponding path, that is
30          * responsible for each match in the form
```

Abridged Source Preview: Argument Interface (2)

```
31          * "[repo path]::[repo name]::[match text]". */
32      ARG_SHOW_HELP,
33          /**< Display help information and exit. */
34      ARG_SHOW_VERSION,
35          /**< Display versioning information and exit. */
36      ARG_LIST_REPOS,
37          /**< List all searchable repositories and continue. */
38      ARG_SEARCH_STRICT,
39          /**< Search only the flag fields, identified by suffixing a hyphen, and
40           * do not consider the entire buffer as a searchable space. */
41      ARG_NO_COMPLAINING,
42          /**< Do not print warnings regarding the presence of the legacy PORTDIR
43           * directory hierarchy. */
44      ARG_SEARCH_NO_CASE,
45          /**< Perform case-insensitive searching. */
46      ARG_ATTEMPT_PORTDIR,
47          /**< Attempt to extract a PORTDIR value from a number of common
48           * locations, likely the environment variables or legacy Portage
49           * configuration files, before falling back to the newer repos.conf
50           * directory hierarchy. */
51      ARG_PRINT_NEEDLE,
52          /**< Print the search needle responsible for each match. */
53      ARG_NO_MIDBUF_WARN,
54          /**< Suppress mid-buffer warnings, and do not clutter the output with
55           * non-fatal error/warning `stderr` messages. See the QUIRKS section of
56           * the manual for details regarding potential trans-buffer issues. */
57      ARG_PKG_FILES_ONLY,
58          /**< Only search files whose names contain ".local". In practice, this
59           * restricts the search to files containing category-package pairs, and
60           * excludes global USE-flag-description documents. */
```

Abridged Source Preview: Argument Interface (3)

```
61      ARG_NO_COLOUR,
62          /**< Do not colour the match output with ANSI escape sequences. */
63      ARG_GLOBAL_ONLY,
64          /**< Do not search files containing package-local flags. */
65
66      /** Special argument flag: the tail-end of the argument vector. */
67      ARG_TAIL
68  };
69
70  /** A status vector, the enumerators of which can be used to indicate the
71   * transient state of the argument-processor. The processor implementation
72   * should restrict its public reporting interface to the confines of these
73   * codes, and should provide a human-readable message for each reportable
74   * condition. */
75  enum arg_stat {
76      ARG_STAT_OK,       /**< Stable state */
77      ARG_STAT_DOUBLE,   /**< An argument has been doubly defined */
78      ARG_STAT_UNKNOWN, /**< An unknown argument string was encountered */
79      ARG_STAT_LACK,     /**< Insufficient arguments were provided */
80      ARG_STAT_EMPTY,    /**< A given argument is meaningless or empty */
81      ARG_STAT_UNABBR,   /**< The command-abbreviation list was erroneous */
82      ARG_STAT_NOMORE,   /**< Further arguments should not be parsed as such */
83      ARG_STAT_GLBPKG,   /**< The specified search space is contradictory */
84  };
85
86  /**
87   * Checks whether an argument is enabled in the given bit string.
88   * @param args the argument vector bit string
89   * @param arg the argument to test
90   * @return is the given argument set in the given bit string?
```

Abridged Source Preview: Argument Interface (4)

```
91   */
92  inline bool arg_check ( args_t args, enum arg_idx arg );
93
94  /**
95   * Provides a human-readable message for the given status code.
96   * @param status the processor condition to decipher
97   * @return the human-readable message
98   */
99  const char * arg_strerror ( enum arg_stat status );
100
101 /**
102  * Invokes the argument-processor on the given argument vector, populating the
103  * bit string with flags corresponding to the fixed argument position index.
104  * @param [in] argc the number of arguments provided in the argument vector
105  * @param [in] argv the text argument vector, typically from the shell
106  * @param [out] bitstring the argument-processor switch output
107  * @return the status of the argument-processor
108  */
109 enum arg_stat arg_parse ( int argc, char ** argv, args_t * bitstring );
110
111 #endif /* ARGS_H */
112
```

Source Preview: Two-Way Heuristic Searching Driver

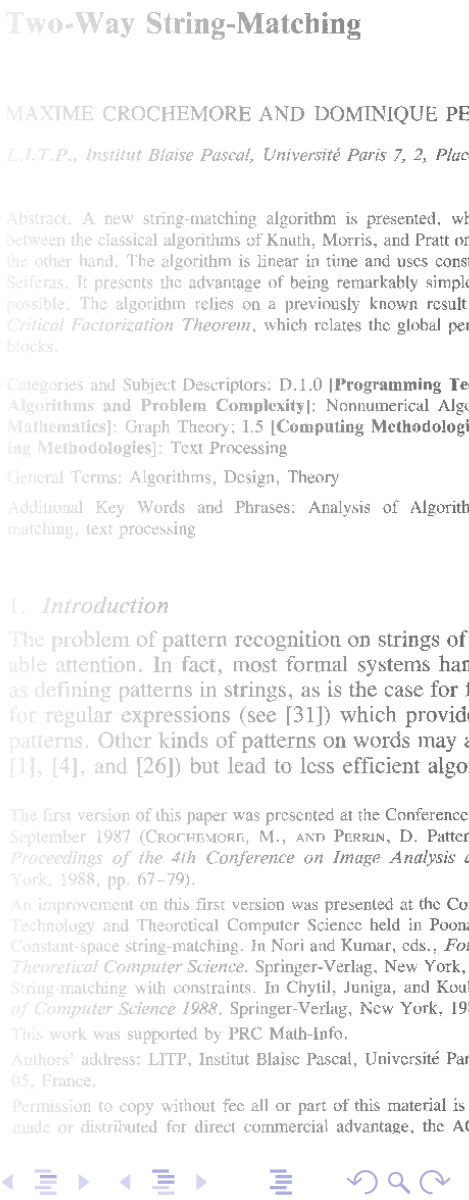
```c
/**
 * Implements a high-level driver for the Two-Way searching algorithm.
 * [... Documentation snipped for presentation]
 *
 * @param haystack the search space
 * @param haystack_len the length of the search space
 * @param needle the match string
 * @param needle_len the length of the match string
 * @return the matched text
 */

static char * twoway_search ( const char * haystack, size_t haystack_len,
        const char * needle, size_t needle_len )
{
    /* Use a reverse lexicographic search by default. */
    char * ( *searcher ) ( size_t, size_t, size_t, const char *, size_t,
        const char *, size_t, size_t ) = &rev_lexi_search;
    size_t primary, secondary, ell, period;
    char * match = NULL;

    primary = compute_suffixes ( needle, needle_len, &period, &secondary );
    ell = primary;

    if ( memcmp ( needle, needle + period, ell + 1 ) == 0 )
        /* Attempt a forward-running search where suffixes are optimal. */
        searcher = &fwd_lexi_search;

    return searcher ( primary, secondary, ell, needle, needle_len, haystack,
        haystack_len, period );
}
```


Two-Way String-Matching

MAXIME CROCHEMORE AND DOMINIQUE PE

L.I.T.P., Institut Blaise Pascal, Université Paris 7, 2, Plac

Abstract. A new string-matching algorithm is presented, wh between the classical algorithms of Knuth, Morris, and Pratt or the other hand. The algorithm is linear in time and uses const Seiferas. It presents the advantage of being remarkably simple possible. The algorithm relies on a previously known result Critical Factorization Theorem, which relates the global per blocks.

Categories and Subject Descriptors: D.1.0 [Programming Te Algorithms and Problem Complexity]: Nonnumerical Alge Mathematics]: Graph Theory; I.5 [Computing Methodologi ing Methodologies]: Text Processing

General Terms: Algorithms, Design, Theory

Additional Key Words and Phrases: Analysis of Algorith matching, text processing

1. Introduction

The problem of pattern recognition on strings of able attention. In fact, most formal systems han as defining patterns in strings, as is the case for f for regular expressions (see [31]) which provid patterns. Other kinds of patterns on words may a [1], [4], and [26]) but lead to less efficient algo

The first version of this paper was presented at the Conference September 1987 (Crochemore, M., and Perrin, D. Patter Proceedings of the 4th Conference on Image Analysis a York, 1988, pp. 67–79).

An improvement on this first version was presented at the Co Technology and Theoretical Computer Science held in Poon Constant-space string-matching. In Nori and Kumar, eds., For Theoretical Computer Science. Springer-Verlag, New York, String-matching with constraints. In Chytil, Juniga, and Koul of Computer Science 1988. Springer-Verlag, New York, 19

This work was supported by PRC Math-Info.

Authors' address: LITP, Institut Blaise Pascal, Université Par 05, France.

Permission to copy without fee all or part of this material is made or distributed for direct commercial advantage, the A

Example Usage: List Repositories and Perform a Query

Invocation: `owd-euses --repo-paths --verbose --list-repos qt5`

```
This is owd-euses, v. placement by Oliver Dixon (MMXX & MMXXIII). For support,
send e-mail to Oliver Dixon <od641@york.ac.uk>.

The source code repository and tarballs are available on-line at
https://github.com/oliverdixon/owd-euses-placement. The code is licensed under
the MIT Licence, the details of which can be found at https://mit-license.org/.

Configuration directory: /etc/portage/repos.conf/

Name: haskell           Location: /var/db/repos/haskell
Name: gentoo            Location: /var/db/repos/gentoo
Name: slonko            Location: /var/db/repos/slonko
Name: pf4public         Location: /var/lib/layman/pf4public

/var/db/repos/gentoo::gentoo::qt5 - Add support for the Qt 5 application and UI framework
/var/db/repos/gentoo::gentoo::dev-libs/quazip:qt5 - Build with Qt5 support
/var/db/repos/gentoo::gentoo::dev-python/QtPy:pyqt5 - Use dev-python/PyQt5 as Qt for Python implementation
/var/db/repos/gentoo::gentoo::dev-python/pyudev:qt5 - Install PyQt5 bindings
/var/db/repos/gentoo::gentoo::games-simulation/flightgear:qt5 - Build Qt5 launcher application
/var/db/repos/gentoo::gentoo::media-gfx/renderdoc:qt5 - Build and install the qrenderdoc GUI
/var/db/repos/gentoo::gentoo::media-libs/openimageio:qt5 - Build iv with Qt5
/var/db/repos/gentoo::gentoo::media-libs/osl:qt5 - Build the osltoy binary
/var/db/repos/gentoo::gentoo::media-video/guvcview:qt5 - Build with Qt5 interface instead Gtk+
```

Equivalent Invocation with Short Arguments: `owd-euses -pvr qt5`

Example Usage: Multiple Global Queries with Identification over a Nonstandard Directory Structure

Invocation: `PORTAGE_CONFIGROOT=/etc/portage owd-euses --global --repo-names --no-case --print-needles tls ssl vpn`

```
(tls) gentoo::gnutls - Prefer net-libs/gnutls as SSL/TLS provider (ineffective with USE=-ssl)
(ssl) gentoo::flac - Add support for FLAC: Free Lossless Audio Codec
(ssl) gentoo::gnutls - Prefer net-libs/gnutls as SSL/TLS provider (ineffective with USE=-ssl)
(tls) gentoo::ssl - Add support for SSL/TLS connections (Secure Socket Layer / Transport Layer Security)
(ssl) gentoo::ssl - Add support for SSL/TLS connections (Secure Socket Layer / Transport Layer Security)
(tls) gentoo::tls - mod_tls TLS module for Apache, intended to replace mod_ssl
(ssl) gentoo::tls - mod_tls TLS module for Apache, intended to replace mod_ssl
(vpn) gentoo::openvpn - Build the openvpn input plugin (reads the status file printed by OpenVPN)
(tls) gentoo::tls - Provide TLS using mbed TLS (formerly known as PolarSSL) - Currently disabled.
(ssl) gentoo::tls - Provide TLS using mbed TLS (formerly known as PolarSSL) - Currently disabled.
(tls) gentoo::ssl_preread - This module allows extracting information from the ClientHello message without
(ssl) gentoo::ssl_preread - This module allows extracting information from the ClientHello message without
(ssl) gentoo::logfile - Enable logging to logfiles (requires USE=ssl)
(ssl) gentoo::sslrouter - Permits routing/proxy of SSL requests
```

Equivalent Invocation with Short Arguments:
`PORTAGE_CONFIGROOT=/etc/portage owd-euses -gnce tls ssl vpn`

Example Usage: Informative Error-Handling

Invocation: `PORTAGE_CONFIGROOT=/var/db/repos/gentoo owd-euses`
Problem: The given directory structure is non-conforming.

```
1  owd-euses: warning: -2(I): No queries were provided.
2  owd-euses caught a fatal error and cannot continue.
3  Re-run with "--help --version" or "-hv" for help.
4
5    Summary: "Could not use the repository-description base directory."
6    Offending Article: "/var/db/repos/gentoo/repos.conf/"
7    Error Detail: "No such file or directory"
8    Status Code: 2 (system errno)
```

Example Usage: Informative Error-Handling

Invocation: `PORTAGE_CONFIGROOT=/var/db/repos/gentoo owd-euses`
Problem: The given directory structure is non-conforming.

```
1  owd-euses: warning: -2(I): No queries were provided.
2  owd-euses caught a fatal error and cannot continue.
3  Re-run with "--help --version" or "-hv" for help.
4
5    Summary: "Could not use the repository-description base directory."
6    Offending Article: "/var/db/repos/gentoo/repos.conf/"
7    Error Detail: "No such file or directory"
8    Status Code: 2 (system errno)
```

Invocation: `owd-euses --global --package qt5`
Problem: The search-space specification is contradictory.

```
1  owd-euses caught a fatal error and cannot continue.
2  Re-run with "--help --version" or "-hv" for help.
3
4    Summary: "Inadequate command-line arguments were provided."
5    Offending Article: "N/A"
6    Error Detail: "The global and package options cannot be set simultaneously."
7    Status Code: -8 (internal)
```

Packaging, Distribution, and Maintenance

`owd-euses` was exceptionally well-received by the community, and undoubtedly achieved its ambitions.

Packaging, Distribution, and Maintenance

`owd-euses` was exceptionally well-received by the community, and undoubtedly achieved its ambitions.

- The package was shipped with makefiles and Portage-compatible installation scripts (*ebuilds*); this allowed for easy adoption by the wider Gentoo community.

Packaging, Distribution, and Maintenance

`owd-euses` was exceptionally well-received by the community, and undoubtedly achieved its ambitions.

- The package was shipped with makefiles and Portage-compatible installation scripts (*ebuilds*); this allowed for easy adoption by the wider Gentoo community.
- After first being advertised on the Gentoo developers' mailing list, `owd-euses` was added to the official software repository, being mirrored by hundreds of servers throughout the world.

Packaging, Distribution, and Maintenance

`owd-euses` was exceptionally well-received by the community, and undoubtedly achieved its ambitions.

- The package was shipped with makefiles and Portage-compatible installation scripts (*ebuilds*); this allowed for easy adoption by the wider Gentoo community.
- After first being advertised on the Gentoo developers' mailing list, `owd-euses` was added to the official software repository, being mirrored by hundreds of servers throughout the world.
- The consistently rigorous Doxygen documentation combined with the `roff`-compatible manual page guarantees a pleasant experience for developers and end-users.

Packaging, Distribution, and Maintenance

`owd-euses` was exceptionally well-received by the community, and undoubtedly achieved its ambitions.

- The package was shipped with makefiles and Portage-compatible installation scripts (*ebuilds*); this allowed for easy adoption by the wider Gentoo community.
- After first being advertised on the Gentoo developers' mailing list, `owd-euses` was added to the official software repository, being mirrored by hundreds of servers throughout the world.
- The consistently rigorous Doxygen documentation combined with the `roff`-compatible manual page guarantees a pleasant experience for developers and end-users.
- A central toolkit for similar tasks has since been released by the central Gentoo Infrastructure Team, but `owd-euses` remains in use to a lesser extent.

Any Questions?