



*Full credit is given to the above companies including the OS that this PDF file was generated!*

## Red Hat Enterprise Linux Release 9.2 Manual Pages on 'OSSL\_STORE\_SEARCH\_by\_issuer\_serial.3ossl' command

**\$ man OSSL\_STORE\_SEARCH\_by\_issuer\_serial.3ossl**

OSSL\_STORE\_SEARCH(3ossl)      OpenSSL      OSSL\_STORE\_SEARCH(3ossl)

### NAME

OSSL\_STORE\_SEARCH, OSSL\_STORE\_SEARCH\_by\_name,  
OSSL\_STORE\_SEARCH\_by\_issuer\_serial,  
OSSL\_STORE\_SEARCH\_by\_key\_fingerprint, OSSL\_STORE\_SEARCH\_by\_alias,  
OSSL\_STORE\_SEARCH\_free, OSSL\_STORE\_SEARCH\_get\_type,  
OSSL\_STORE\_SEARCH\_get0\_name, OSSL\_STORE\_SEARCH\_get0\_serial,  
OSSL\_STORE\_SEARCH\_get0\_bytes, OSSL\_STORE\_SEARCH\_get0\_string,  
OSSL\_STORE\_SEARCH\_get0\_digest - Type and functions to create OSSL\_STORE  
search criteria

### SYNOPSIS

```
#include <openssl/store.h>

typedef struct ossl_store_search_st OSSL_STORE_SEARCH;

OSSL_STORE_SEARCH *OSSL_STORE_SEARCH_by_name(X509_NAME *name);
OSSL_STORE_SEARCH *OSSL_STORE_SEARCH_by_issuer_serial(X509_NAME *name,
                                                       const ASN1_INTEGER
                                                       *serial);

OSSL_STORE_SEARCH *OSSL_STORE_SEARCH_by_key_fingerprint(const EVP_MD *digest,
                                                         const unsigned char
                                                         *bytes, int len);

OSSL_STORE_SEARCH *OSSL_STORE_SEARCH_by_alias(const char *alias);

void OSSL_STORE_SEARCH_free(OSSL_STORE_SEARCH *search);

int OSSL_STORE_SEARCH_get_type(const OSSL_STORE_SEARCH *criterion);
```

```

X509_NAME *OSSL_STORE_SEARCH_get0_name(OSSL_STORE_SEARCH *criterion);
const ASN1_INTEGER *OSSL_STORE_SEARCH_get0_serial(const OSSL_STORE_SEARCH
        *criterion);
const unsigned char *OSSL_STORE_SEARCH_get0_bytes(const OSSL_STORE_SEARCH
        *criterion, size_t *length);
const char *OSSL_STORE_SEARCH_get0_string(const OSSL_STORE_SEARCH *criterion);
const EVP_MD *OSSL_STORE_SEARCH_get0_digest(const OSSL_STORE_SEARCH
        *criterion);

```

## DESCRIPTION

These functions are used to specify search criteria to help search for specific objects through other names than just the URI that's given to `OSSL_STORE_open()`. For example, this can be useful for an application that has received a URI and then wants to add on search criteria in a uniform and supported manner.

## Types

`OSSL_STORE_SEARCH` is an opaque type that holds the constructed search criterion, and that can be given to an `OSSL_STORE` context with `OSSL_STORE_find()`.

The calling application owns the allocation of an `OSSL_STORE_SEARCH` at all times, and should therefore be careful not to deallocate it before `OSSL_STORE_close()` has been called for the `OSSL_STORE` context it was given to.

## Application Functions

`OSSL_STORE_SEARCH_by_name()`, `OSSL_STORE_SEARCH_by_issuer_serial()`, `OSSL_STORE_SEARCH_by_key_fingerprint()`, and `OSSL_STORE_SEARCH_by_alias()` are used to create an `OSSL_STORE_SEARCH` from a subject name, an issuer name and serial number pair, a key fingerprint, and an alias (for example a friendly name). The parameters that are provided are not copied, only referred to in a criterion, so they must have at least the same life time as the created `OSSL_STORE_SEARCH`.

`OSSL_STORE_SEARCH_free()` is used to free the `OSSL_STORE_SEARCH`.

## Loader Functions

OSSL\_STORE\_SEARCH\_get\_type() returns the criterion type for the given OSSL\_STORE\_SEARCH.

OSSL\_STORE\_SEARCH\_get0\_name(), OSSL\_STORE\_SEARCH\_get0\_serial(), OSSL\_STORE\_SEARCH\_get0\_bytes(), OSSL\_STORE\_SEARCH\_get0\_string(), and OSSL\_STORE\_SEARCH\_get0\_digest() are used to retrieve different data from a OSSL\_STORE\_SEARCH, as available for each type. For more information, see "SUPPORTED CRITERION TYPES" below.

## SUPPORTED CRITERION TYPES

Currently supported criterion types are:

### OSSL\_STORE\_SEARCH\_BY\_NAME

This criterion supports a search by exact match of subject name.

The subject name itself is a X509\_NAME pointer. A criterion of this type is created with OSSL\_STORE\_SEARCH\_by\_name(), and the actual subject name is retrieved with

OSSL\_STORE\_SEARCH\_get0\_name().

### OSSL\_STORE\_SEARCH\_BY\_ISSUER\_SERIAL

This criterion supports a search by exact match of both issuer name and serial number. The issuer name itself is a X509\_NAME pointer, and the serial number is a ASN1\_INTEGER pointer. A criterion of this type is created with OSSL\_STORE\_SEARCH\_by\_issuer\_serial() and the actual issuer name and serial number are retrieved with

OSSL\_STORE\_SEARCH\_get0\_name() and OSSL\_STORE\_SEARCH\_get0\_serial().

### OSSL\_STORE\_SEARCH\_BY\_KEY\_FINGERPRINT

This criterion supports a search by exact match of key fingerprint.

The key fingerprint in itself is a string of bytes and its length, as well as the algorithm that was used to compute the fingerprint.

The digest may be left unspecified (NULL), and in that case, the loader has to decide on a default digest and compare fingerprints accordingly. A criterion of this type is created with

OSSL\_STORE\_SEARCH\_by\_key\_fingerprint() and the actual fingerprint and its length can be retrieved with

OSSL\_STORE\_SEARCH\_get0\_bytes(). The digest can be retrieved with

OSSL\_STORE\_SEARCH\_get0\_digest().

## OSSL\_STORE\_SEARCH\_BY\_ALIAS

This criterion supports a search by match of an alias of some kind.

The alias in itself is a simple C string. A criterion of this type is created with `OSSL_STORE_SEARCH_by_alias()` and the actual alias is retrieved with `OSSL_STORE_SEARCH_get0_string()`.

## RETURN VALUES

`OSSL_STORE_SEARCH_by_name()`, `OSSL_STORE_SEARCH_by_issuer_serial()`, `OSSL_STORE_SEARCH_by_key_fingerprint()`, and `OSSL_STORE_SEARCH_by_alias()` return a `OSSL_STORE_SEARCH` pointer on success, or `NULL` on failure.

`OSSL_STORE_SEARCH_get_type()` returns the criterion type of the given `OSSL_STORE_SEARCH`. There is no error value.

`OSSL_STORE_SEARCH_get0_name()` returns a `X509_NAME` pointer on success, or `NULL` when the given `OSSL_STORE_SEARCH` was of a different type.

`OSSL_STORE_SEARCH_get0_serial()` returns a `ASN1_INTEGER` pointer on success, or `NULL` when the given `OSSL_STORE_SEARCH` was of a different type.

`OSSL_STORE_SEARCH_get0_bytes()` returns a const unsigned char pointer and sets `*length` to the strings length on success, or `NULL` when the given `OSSL_STORE_SEARCH` was of a different type.

`OSSL_STORE_SEARCH_get0_string()` returns a const char pointer on success, or `NULL` when the given `OSSL_STORE_SEARCH` was of a different type.

`OSSL_STORE_SEARCH_get0_digest()` returns a const `EVP_MD` pointer. `NULL` is a valid value and means that the store loader default will be used when applicable.

## SEE ALSO

`ossl_store(7)`, `OSSL_STORE_supports_search(3)`, `OSSL_STORE_find(3)`

## HISTORY

`OSSL_STORE_SEARCH`, `OSSL_STORE_SEARCH_by_name()`, `OSSL_STORE_SEARCH_by_issuer_serial()`, `OSSL_STORE_SEARCH_by_key_fingerprint()`, `OSSL_STORE_SEARCH_by_alias()`, `OSSL_STORE_SEARCH_free()`, `OSSL_STORE_SEARCH_get_type()`,

OSSL\_STORE\_SEARCH\_get0\_name(), OSSL\_STORE\_SEARCH\_get0\_serial(),  
OSSL\_STORE\_SEARCH\_get0\_bytes(), and OSSL\_STORE\_SEARCH\_get0\_string()  
were added in OpenSSL 1.1.1.

## COPYRIGHT

Copyright 2018-2020 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use

this file except in compliance with the License. You can obtain a copy

in the file LICENSE in the source distribution or at

<<https://www.openssl.org/source/license.html>>.

3.0.7                    2023-07-13        OSSL\_STORE\_SEARCH(3ossl)