

**NAME**

**gss\_inquire\_context** - Obtain information about a security context

**SYNOPSIS**

```
#include <gssapi/gssapi.h>
```

*OM\_uint32*

```
gss_inquire_context(OM_uint32 *minor_status, const gss_ctx_id_t context_handle,  

gss_name_t *src_name, gss_name_t *targ_name, OM_uint32 *lifetime_rec, gss_OID *mech_type,  

OM_uint32 *ctx_flags, int *locally_initiated, int *open);
```

**DESCRIPTION**

Obtains information about a security context. The caller must already have obtained a handle that refers to the context, although the context need not be fully established.

**PARAMETERS**

|                       |  |
|-----------------------|--|
| <i>minor_status</i>   | Mechanism specific status code.  |
| <i>context_handle</i> | A handle that refers to the security context.  |
| <i>src_name</i>       | The name of the context initiator. If the context was established using anonymous authentication, and if the application invoking <b>gss_inquire_context()</b> is the context acceptor, an anonymous name will be returned. Storage associated with this name must be freed by the application after use with a call to <b>gss_release_name()</b> . Specify NULL if not required.        |
| <i>targ_name</i>      | The name of the context acceptor. Storage associated with this name must be freed by the application after use with a call to <b>gss_release_name()</b> . If the context acceptor did not authenticate itself, and if the initiator did not specify a target name in its call to <b>gss_init_sec_context()</b> , the value GSS_C_NO_NAME will be returned. Specify NULL if not required. |
| <i>lifetime_rec</i>   | The number of seconds for which the context will remain valid. If the context has expired, this parameter will be set to zero. If the implementation does not support context expiration, the value GSS_C_INDEFINITE will be returned. Specify NULL if not required.   |
| <i>mech_type</i>      | The security mechanism providing the context. The returned OID will be a pointer to static storage that should be treated as read-only by the application; in particular the application should not attempt to free it. Specify NULL if not required.  |

`ctx_flags` Contains various independent flags, each of which indicates that the context supports (or is expected to support, if *open* is false) a specific service option. If not needed, specify NULL. Symbolic names are provided for each flag, and the symbolic names corresponding to the required flags should be logically-ANDed with the *ctx\_flags* value to test whether a given option is supported by the context. The flags are:

GSS\_C\_DELEG\_FLAG

True Credentials were delegated from the initiator to the acceptor.

False No credentials were delegated.

GSS\_C\_MUTUAL\_FLAG

True The acceptor was authenticated to the initiator.

False The acceptor did not authenticate itself.

GSS\_C\_REPLAY\_FLAG

True Replay of protected messages will be detected.

False Replayed messages will not be detected.

GSS\_C\_SEQUENCE\_FLAG

True Out-of-sequence protected messages will be detected.

False Out-of-sequence messages will not be detected.

GSS\_C\_CONF\_FLAG

True Confidentiality service may be invoked by calling **gss\_wrap()** routine.

False No confidentiality service (via **gss\_wrap()**) available. **gss\_wrap()** will provide message encapsulation, data-origin authentication and integrity services only.

GSS\_C\_INTEG\_FLAG

True Integrity service may be invoked by calling either **gss\_get\_mic()** or **gss\_wrap()** routines.

False Per-message integrity service unavailable.

#### GSS\_C\_ANON\_FLAG

True The initiator's identity will not be revealed to the acceptor. The *src\_name* parameter (if requested) contains an anonymous internal name.

False The initiator has been authenticated normally.

#### GSS\_C\_PROT\_READY\_FLAG

True Protection services (as specified by the states of the GSS\_C\_CONF\_FLAG and GSS\_C\_INTEG\_FLAG) are available for use.

False Protection services (as specified by the states of the GSS\_C\_CONF\_FLAG and GSS\_C\_INTEG\_FLAG) are available only if the context is fully established (i.e. if the *open* parameter is non-zero).

#### GSS\_C\_TRANS\_FLAG

True The security context may be transferred to other processes via a call to **gss\_export\_sec\_context()**.

False The security context is not transferable.

locally\_initiated Non-zero if the invoking application is the context initiator. Specify NULL if not required.

open Non-zero if the context is fully established; Zero if a context-establishment token is expected from the peer application. Specify NULL if not required.

### RETURN VALUES

GSS\_S\_COMPLETE Successful completion

GSS\_S\_NO\_CONTEXT The referenced context could not be accessed

### SEE ALSO

`gss_export_sec_context(3)`, `gss_get_mic(3)`, `gss_init_sec_context(3)`, `gss_release_name(3)`, `gss_wrap(3)`

## STANDARDS

RFC 2743 Generic Security Service Application Program Interface Version 2, Update 1

RFC 2744 Generic Security Service API Version 2 : C-bindings

## HISTORY

The `gss_inquire_context` function first appeared in FreeBSD 7.0.

## AUTHORS

John Wray, Iris Associates

## COPYRIGHT

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.