

globus gsi cert utils Reference Manual

5.5

Generated by Doxygen 1.3.5

Fri Jun 26 23:24:46 2009

Contents

1 Globus GSI Certificate Handling Utilities	1
2 globus gsi cert utils Module Index	1
3 globus gsi cert utils Module Documentation	1

1 Globus GSI Certificate Handling Utilities

The Globus GSI Certificate Handling Utilities library. This library contains helper functions for dealing with certificates.

- [Activation](#)
- [Cert Utils Functions](#)
- [Cert Utils Constants](#)

2 globus gsi cert utils Module Index

2.1 globus gsi cert utils Modules

Here is a list of all modules:

Activation	1
Cert Utils Functions	2
Cert Utils Constants	4

3 globus gsi cert utils Module Documentation

3.1 Activation

Globus GSI Cert Utils uses standard Globus module activation and deactivation.

Defines

- `#define GLOBUS\_GSI\_CERT\_UTILS\_MODULE`

3.1.1 Detailed Description

Globus GSI Cert Utils uses standard Globus module activation and deactivation.

Before any Globus GSI Cert Utils functions are called, the following function must be called:

```
globus_module_activate(GLOBUS_GSI_CERT_UTILS_MODULE)
```

This function returns GLOBUS_SUCCESS if Globus GSI Credential was successfully initialized, and you are therefore allowed to subsequently call Globus GSI Cert Utils functions. Otherwise, an error code is returned, and Globus GSI Cert Utils functions should not be subsequently called. This function may be called multiple times.

To deactivate Globus GSI Cert Utils, the following function must be called:

```
globus_module_deactivate(GLOBUS_GSI_CERT_UTILS_MODULE)
```

This function should be called once for each time Globus GSI Cert Utils was activated.

3.1.2 De ne Documentation

3.1.2.1 #de ne GLOBUS_GSI_CERT_UTILS_MODULE

Module descriptor.

3.2 Cert Utils Functions

A generic set of utility functions for manipulating OpenSSL objects, such as X509 certificates.

Convert ASN1_UTCTIME to time_t

- globus_result_t [globus_gsi_cert_utils_make_time](#)(ASN1_UTCTIME ctm, time_t newtime)

Get the X509 certificate type (EEC, CA, proxy type, etc.)

- globus_result_t [globus_gsi_cert_utils_get_cert_type](#)(X509 cert, globus_gsi_cert_utils_cert_type_t type)

Get the certificate name

- globus_result_t [globus_gsi_cert_utils_get_x509_name](#)(char subject_string, int length, X509_NAME* x509_name)

Get the base certificate name

- globus_result_t [globus_gsi_cert_utils_get_base_name](#)(X509_NAME subject, STACK_OF(X509) cert_chain)

Functions

- globus_result_t [globus_gsi_cert_utils_get_ecc](#)(STACK_OF(X509) cert_chain, X509_ecc)

3.2.1 Detailed Description

A generic set of utility functions for manipulating OpenSSL objects, such as X509 certificates.

3.2.2 Function Documentation

3.2.2.1 `globus_result_t globus_gsi_cert_utils_make_time (ASN1_UTCTIME ctm, time_t newtime)`

Convert a ASN1_UTCTIME structure to a time_t.

Parameters:

ctm The ASN1_UTCTIME to convert
newtime The converted time

Returns:

GLOBUS_SUCCESS or an error captured in a globus_result_t

3.2.2.2 `globus_result_t globus_gsi_cert_utils_get_cert_type (X509 cert, globus_gsi_cert_utils_cert_type_t type)`

Determine the type of the given X509 certificate For the list of possible values returned, see globus_gsi_cert_utils_cert_type_t.

Parameters:

cert The X509 certificate
type The returned X509 certificate type

Returns:

GLOBUS_SUCCESS or an error captured in a globus_result_t

3.2.2.3 `globus_result_t globus_gsi_cert_utils_get_x509_name (char subject_string int length, X509_NAME x509_name)`

Get the X509_NAME from a subject string.

OpenSSL doesn't provide this function, probably because it shouldn't be used. If you are getting an X509_NAME from just a string, it's impossible to verify its integrity.

Parameters:

subject_string The subject in the format: "/O=Grid/OU=..."
length The length of the subject string
x509_name The resulting X509_NAME object

Returns:

GLOBUS_SUCCESS unless an error occurred, in which case, a globus error object ID is returned

3.2.2.4 `globus_result_t globus_gsi_cert_utils_get_base_name (X509_NAME subject STACK_OF(X509) cert_chain)`

Get the base name of a proxy certificate.

Given an X509 name, strip off the proxy related /CN components to get the base name of the certificate's subject

Parameters:

subject Pointer to an X509_NAME object which gets stripped

`cert_chain` The certificate chain used to detect the number of CNs to strip. This is done by figuring out the number of proxies in the chain.

Returns:

`GLOBUS_SUCCESS`

3.2.2.5 `globus_result_t globus_gsi_cert_utils_get_eec (STACK_OF(X509) cert_chain, X509 * eec)`

Get the end-entity certificate associated with a certificate chain.

Parameters:

`cert_chain` Certificate chain to inspect.

`eec` Pointer to be set to the EEC value from within the cert chain. Must freed by the caller.

3.3 Cert Utils Constants

Typedefs

- typedef enum `globus_gsi_cert_utils_cert_type` `globus_gsi_cert_utils_cert_type_t`

Enumerations

- enum `globus_gsi_cert_utils_error_t`
 - `GLOBUS_GSI_CERT_UTILS_ERROR_SUCCESS` 0,
 - `GLOBUS_GSI_CERT_UTILS_ERROR_GETTING_NAME_ENTRY_OF_SUBJECT` 1,
 - `GLOBUS_GSI_CERT_UTILS_ERROR_COPYING_SUBJECT` 2,
 - `GLOBUS_GSI_CERT_UTILS_ERROR_GETTING_CN_ENTRY` 3,
 - `GLOBUS_GSI_CERT_UTILS_ERROR_ADDING_CN_TO_SUBJECT` 4,
 - `GLOBUS_GSI_CERT_UTILS_ERROR_OUT_OF_MEMORY` 5,
 - `GLOBUS_GSI_CERT_UTILS_ERROR_UNEXPECTED_FORMAT` 6,
 - `GLOBUS_GSI_CERT_UTILS_ERROR_NON_COMPLIANT_PROXY` 7,
 - `GLOBUS_GSI_CERT_UTILS_ERROR_DETERMINING_CERT_TYPE` 8,
 - `GLOBUS_GSI_CERT_UTILS_ERROR_LAST` 9 }
- enum `globus_gsi_cert_utils_cert_type_t`
 - `GLOBUS_GSI_CERT_UTILS_TYPE_DEFAULT` 0,
 - `GLOBUS_GSI_CERT_UTILS_TYPE_EEC` (1 << 0),
 - `GLOBUS_GSI_CERT_UTILS_TYPE_CA` (1 << 1),
 - `GLOBUS_GSI_CERT_UTILS_TYPE_GSI_2` (1 << 2),
 - `GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3` (1 << 3),
 - `GLOBUS_GSI_CERT_UTILS_TYPE_RFC` (1 << 4),
 - `GLOBUS_GSI_CERT_UTILS_TYPE_FORMAT_MASK`
 - `GLOBUS_GSI_CERT_UTILS_TYPE_IMPERSONATION_PROXY` (1 << 5),
 - `GLOBUS_GSI_CERT_UTILS_TYPE_LIMITED_PROXY` (1 << 6),
 - `GLOBUS_GSI_CERT_UTILS_TYPE_RESTRICTED_PROXY` (1 << 7),

```

GLOBUS_GSI_CERT_UTILS_TYPE_INDEPENDENT_PROXY(1 << 8),
GLOBUS_GSI_CERT_UTILS_TYPE_PROXY_MASK
GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_IMPERSONATION_PROXY
GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_INDEPENDENT_PROXY
GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_LIMITED_PROXY
GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_RESTRICTED_PROXY
GLOBUS_GSI_CERT_UTILS_TYPE_GSI_2_PROXY
GLOBUS_GSI_CERT_UTILS_TYPE_GSI_2_LIMITED_PROXY
GLOBUS_GSI_CERT_UTILS_TYPE_RFC_IMPERSONATION_PROXY
GLOBUS_GSI_CERT_UTILS_TYPE_RFC_INDEPENDENT_PROXY
GLOBUS_GSI_CERT_UTILS_TYPE_RFC_LIMITED_PROXY
GLOBUS_GSI_CERT_UTILS_TYPE_RFC_RESTRICTED_PROXY

```

3.3.1 Typedef Documentation

3.3.1.1 typedef enum globus_gsi_cert_utils_cert_type globus_gsi_cert_utils_cert_type_t

Certificate Types.

These certificate types are used to describe some properties of a certificate and to specify what type of proxy should be generated in the proxy core code. There are two non-proxy types of certificates understood by Globus: EEC (End-Entity Certificate) and CA (Certificate Authority Certificates), three proxy formats (GSI 2 "legacy" proxies, GSI 3 "Draft" proxies, and RFC 3820-compliant proxies), and four types of proxy (limited, impersonation "full", restricted, and independent). The latter two types are not expressible in the GSI 2 format.

In addition to enumerations for the concrete renderings of certificate format and type combined, there are default, formats-without-types and types-without-formats so that application logic which uses the proxy library can request default proxy formats which are compatible with the issuing certificate.

3.3.2 Enumeration Type Documentation

3.3.2.1 enum globus_gsi_cert_utils_error_t

Cert Utils Error Codes.

Enumeration values:

```

GLOBUS_GSI_CERT_UTILS_ERROR_SUCCESS Success - never used.
GLOBUS_GSI_CERT_UTILS_ERROR_GETTING_NAME_ENTRY_OF_SUBJECT Failed to retrieve a
    subcomponent of the subject.
GLOBUS_GSI_CERT_UTILS_ERROR_COPYING_SUBJECT A error occurred while trying to copy a X.509
    subject.
GLOBUS_GSI_CERT_UTILS_ERROR_GETTING_CN_ENTRY Failed to retrieve a CN subcomponent of
    the subject.
GLOBUS_GSI_CERT_UTILS_ERROR_ADDING_CN_TO_SUBJECT Failed to add a CN component to a
    X.509 subject name.
GLOBUS_GSI_CERT_UTILS_ERROR_OUT_OF_MEMORY Out of memory.
GLOBUS_GSI_CERT_UTILS_ERROR_UNEXPECTED_FORMATS Something unexpected happen while
    converting a string subject to a X509_NAME structure.

```

`GLOBUS_GSI_CERT_UTILS_ERROR_NON_COMPLIANT_PROXY` Proxy does not comply with the expected format.
`GLOBUS_GSI_CERT_UTILS_ERROR_DETERMINING_CERT_TYPE` Couldn't determine the certificate type.
`GLOBUS_GSI_CERT_UTILS_ERROR_LAST` Last marker - never used.

3.3.2.2 enum `globus_gsi_cert_utils_cert_type_e`

Certificate Types.

These certificate types are used to describe some properties of a certificate and to specify what type of proxy should be generated in the proxy core code. There are two non-proxy types of certificates understood by Globus: EEC (End-Entity Certificate) and CA (Certificate Authority Certificates), three proxy formats (GSI 2 "legacy" proxies, GSI 3 "Draft" proxies, and RFC 3820-compliant proxies), and four types of proxy (limited, impersonation "full", restricted, and independent). The latter two types are not expressible in the GSI 2 format.

In addition to enumerations for the concrete renderings of certificate format and type combined, there are default, formats-without-types and types-without-formats so that application logic which uses the proxy library can request default proxy formats which are compatible with the issuing certificate.

Enumeration values:

`GLOBUS_GSI_CERT_UTILS_TYPE_DEFAULT` Default proxy type.
`GLOBUS_GSI_CERT_UTILS_TYPE_EECA` end entity certificate.
`GLOBUS_GSI_CERT_UTILS_TYPE_CAA` CA certificate.
`GLOBUS_GSI_CERT_UTILS_TYPE_GSI_2` Legacy Proxy Format.
`GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_X509_PROXY_CERTIFICATE_PROLE_DRAFT` Proxy Certificate Profile (draft) Proxy Format.
`GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_X509_PROXY_CERTIFICATE_PROLE_COMPLIANT` Proxy Certificate Profile Compliant Proxy Format.
`GLOBUS_GSI_CERT_UTILS_TYPE_FORMAT_MASK` Proxy certificate formats mask.
`GLOBUS_GSI_CERT_UTILS_TYPE_IMPERSONATION_PROXY` Impersonation proxy type.
`GLOBUS_GSI_CERT_UTILS_TYPE_LIMITED_PROXY` Limited proxy type.
`GLOBUS_GSI_CERT_UTILS_TYPE_RESTRICTED_PROXY` Restricted proxy type.
`GLOBUS_GSI_CERT_UTILS_TYPE_INDEPENDENT_PROXY` Independent proxy type.
`GLOBUS_GSI_CERT_UTILS_TYPE_PROXY_MASK` Proxy types mask.
`GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_IMPERSONATION_PROXY_X509_PROXY_CERTIFICATE_PROLE_PRERFC_COMPLIANT` X.509 Proxy Certificate Profile (pre-RFC) compliant impersonation proxy.
`GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_INDEPENDENT_PROXY_X509_PROXY_CERTIFICATE_PROLE_PRERFC_COMPLIANT` X.509 Proxy Certificate Profile (pre-RFC) compliant independent proxy.
`GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_LIMITED_PROXY_X509_PROXY_CERTIFICATE_PROLE_PRERFC_COMPLIANT` X.509 Proxy Certificate Profile (pre-RFC) compliant limited proxy.
`GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_RESTRICTED_PROXY_X509_PROXY_CERTIFICATE_PROLE_PRERFC_COMPLIANT` X.509 Proxy Certificate Profile (pre-RFC) compliant restricted proxy.
`GLOBUS_GSI_CERT_UTILS_TYPE_GSI_2_PROXY_X509_PROXY_CERTIFICATE_PROLE_LEGACY_GLOBUS_IMPERSONATION_PROXY` legacy Globus impersonation proxy.
`GLOBUS_GSI_CERT_UTILS_TYPE_GSI_2_LIMITED_PROXY_X509_PROXY_CERTIFICATE_PROLE_LEGACY_GLOBUS_LIMITED_IMPERSONATION_PROXY` legacy Globus limited impersonation proxy.
`GLOBUS_GSI_CERT_UTILS_TYPE_RFC_IMPERSONATION_PROXY_X509_PROXY_CERTIFICATE_PROLE_RFC_COMPLIANT` X.509 Proxy Certificate Profile RFC compliant impersonation proxy.
`GLOBUS_GSI_CERT_UTILS_TYPE_RFC_INDEPENDENT_PROXY_X509_PROXY_CERTIFICATE_PROLE_RFC_COMPLIANT` X.509 Proxy Certificate Profile RFC compliant independent proxy.

GLOBUS_GSI_CERT_UTILS_TYPE_RFC_LIMITED_PROXY X.509 Proxy Certificate Profile RFC compliant limited proxy.

GLOBUS_GSI_CERT_UTILS_TYPE_RFC_RESTRICTED_PROXY X.509 Proxy Certificate Profile RFC compliant restricted proxy.

Index

Activation, [1](#)

Cert Utils Constants, [4](#)

Cert Utils Functions, [2](#)

globus_gsi_cert_utils

globus_gsi_cert_utils_get_base_name, [6](#)

globus_gsi_cert_utils_get_cert_type, [6](#)

globus_gsi_cert_utils_get_eoc, [6](#)

globus_gsi_cert_utils_get_x509_name, [6](#)

globus_gsi_cert_utils_make_time, [6](#)

globus_gsi_cert_utils_activation

GLOBUS_GSI_CERT_UTILS_MODULE, [2](#)

globus_gsi_cert_utils_cert_type_e

globus_gsi_cert_utils_constants, [6](#)

globus_gsi_cert_utils_cert_type_t

globus_gsi_cert_utils_constants, [5](#)

globus_gsi_cert_utils_constants

GLOBUS_GSI_CERT_UTILS_ERROR_-
ADDING_CN_TO_SUBJECT, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_-
COPYING_SUBJECT, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_-
DETERMINING_CERT_TYPE, [6](#)

GLOBUS_GSI_CERT_UTILS_ERROR_-
GETTING_CN_ENTRY, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_-
GETTING_NAME_ENTRY_OF_-
SUBJECT, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_LAST,
[6](#)

GLOBUS_GSI_CERT_UTILS_ERROR_NON_-
COMPLIANT_PROXY, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_OUT_-
OF_MEMORY, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_-
SUCCESS, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_-
UNEXPECTED_FORMAT, [5](#)

GLOBUS_GSI_CERT_UTILS_TYPE_CA, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_-
DEFAULT, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_EE, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_-
FORMAT_MASK, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_GSI_0, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_GSI_2_-
LIMITED_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_GSI_2_-
PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_GSI_6, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_-
IMPERSONATION_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_-
INDEPENDENT_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_-
LIMITED_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_-
RESTRICTED_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_-
IMPERSONATION_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_-
INDEPENDENT_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_-
LIMITED_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_-
PROXY_MASK, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_-
RESTRICTED_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_RFC, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_RFC_-
IMPERSONATION_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_RFC_-
INDEPENDENT_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_RFC_-
LIMITED_PROXY, [6](#)

GLOBUS_GSI_CERT_UTILS_TYPE_RFC_-
RESTRICTED_PROXY, [7](#)

globus_gsi_cert_utils_constants

globus_gsi_cert_utils_cert_type, [6](#)

globus_gsi_cert_utils_cert_type, [5](#)

globus_gsi_cert_utils_error, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_ADDING_-
CN_TO_SUBJECT

globus_gsi_cert_utils_constants, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_-
COPYING_SUBJECT

globus_gsi_cert_utils_constants, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_-
DETERMINING_CERT_TYPE

globus_gsi_cert_utils_constants, [6](#)

GLOBUS_GSI_CERT_UTILS_ERROR_-
GETTING_CN_ENTRY

globus_gsi_cert_utils_constants, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_-
GETTING_NAME_ENTRY_OF_-
SUBJECT

globus_gsi_cert_utils_constants, [5](#)

GLOBUS_GSI_CERT_UTILS_ERROR_LAST

globus_gsi_cert_utils_constants, [6](#)

GLOBUS_GSI_CERT_UTILS_ERROR_NON_-

COMPLIANT_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_ERROR_OUT_OF_MEMORY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_ERROR_SUCCESS
 globus_gsi_cert_utils_constants, [fs](#)
 globus_gsi_cert_utils_error_t
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_ERROR_UNEXPECTED_FORMAT
 globus_gsi_cert_utils_constants, [fs](#)
 globus_gsi_cert_utils_get_base_name
 globus_gsi_cert_utils, [s](#)
 globus_gsi_cert_utils_get_cert_type
 globus_gsi_cert_utils, [s](#)
 globus_gsi_cert_utils_get_eec
 globus_gsi_cert_utils, [s](#)
 globus_gsi_cert_utils_get_x509_name
 globus_gsi_cert_utils, [s](#)
 globus_gsi_cert_utils_make_time
 globus_gsi_cert_utils, [s](#)
 GLOBUS_GSI_CERT_UTILS_MODULE
 globus_gsi_cert_utils_activation, [a](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_CA
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_DEFAULT
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_EEC
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_FORMAT_MASK
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_GSI_2
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_GSI_2_LIMITED_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_GSI_2_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_IMPERSONATION_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_INDEPENDENT_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_LIMITED_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_GSI_3_RESTRICTED_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 RESTRICTED_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_IMPERSONATION_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_INDEPENDENT_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_LIMITED_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_PROXY_MASK
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_RESTRICTED_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_RFC
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_RFC_IMPERSONATION_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_RFC_INDEPENDENT_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_RFC_LIMITED_PROXY
 globus_gsi_cert_utils_constants, [fs](#)
 GLOBUS_GSI_CERT_UTILS_TYPE_RFC_RESTRICTED_PROXY
 globus_gsi_cert_utils_constants, [fs](#)