

OptiNum Grid Installer

User and Reference Manual edition 1.0

Last updated November 14, 2013

Laurence D. Finston

Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen
(GWDG)

OptiNum Grid Installer User and Reference Manual, edition 1.0.

The author is Laurence D. Finston.

Copyright © 2010, 2011, 2012, 2013 Gesellschaft für wissenschaftliche Datenverarbeitung mbH, Göttingen, Germany

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, with the Front-Cover Texts being “GWDG Archive Interface User and Reference Manual”, and no Back-Cover Text. A copy of the license is included in the section entitled “GNU Free Documentation License.”

Short Contents

List of Figures	ix
1 Introduction	1
2 Invoking optdbcli/optdbsrv	6
3 Using the Server Program optdbsrv	14
4 Using the Client Program optdbcli	15
5 Global Variables and Constants optdbcli/optdbsrv	38
6 Global Functions optdbcli/optdbsrv	42
7 Server Connection Function Reference	43
8 Client Connection Function Reference	44
9 Web Application	45
10 Software Installation	49
11 TLS Connections	52
12 X509_Cert_Type Reference	56
13 Proxy Credential Function Reference	58
14 Scanner Server	59
15 Parser Server	60
16 Scanner Client	61
17 Parser Client	62
18 Entry_Type Reference	63
19 Distinguished_Name_Type Reference	66
20 Response_Type Reference	68
21 Scan_Parse_Parameter_Type Reference	72
22 Globus_Site_Info_Type Reference	82
23 Parser Function Reference	83
24 GPG_Key_Pair_Type Reference	84
25 Session_Data_Type Reference	85
26 Utility Function Reference	86
27 Thread Function Reference	87
28 Hidden Installation Reference (scrinstl)	89
29 Auxiliary Program and Shellsript Reference	92
30 Database Tables and Views	97
Glossary	102
Lists of Source Files	104
GNU Free Documentation License	109
GNU General Public License	117

Variable Index	128
Data Type Index	136
Function Index	137
Concept Index	141

Table of Contents

List of Figures	ix
1 Introduction.....	1
1.1 Building and Installing.....	1
1.1.1 Server and Client.....	1
1.1.1.1 Database Setup.....	2
1.1.1.2 Prerequisites	2
1.2 Communication Among the Programs	3
2 Invoking optdbcli/optdbsrv.....	6
2.1 Command Line Arguments and Options	6
2.1.1 Help Options.....	6
2.1.2 Connection Options	6
2.1.3 X.509 Authentication/Authorization Options	7
2.1.4 Input and Output Options.....	8
2.1.5 Software Installation Options.....	8
2.1.6 Debugging Options.....	10
2.1.7 Session Data Options.....	11
2.1.8 Alphabetical List of Options.....	11
2.2 Variable Reference	13
2.3 Function Reference.....	13
3 Using the Server Program optdbsrv	14
4 Using the Client Program optdbcli	15
4.1 How the Client and the Server Communicate.....	16
4.2 How To Read Parser Rules	17
4.3 Sample Dialogue	20
4.4 Software Package Information	20
4.4.1 Adding Software Package Information	20
4.4.2 Deleting Software Package Information	24
4.4.3 Showing Software Package Information	28
4.5 Setting Effective User	29
4.6 Showing X.509 Certificates	30
4.7 Environment Shellscrips.....	30
4.7.1 Sending Environment Shellscrips.....	30
4.7.2 Showing Environment Shellscrips	31
4.8 Showing Privileges	32
4.9 Retrieving Information and Installing Locally.....	33
4.10 Remote Installation	35
4.11 Transferring Proxy Credential.....	36
4.12 Globus Sites	36

4.12.1	Adding Globus Sites	36
4.12.2	Enabling and Disabling Globus Sites	37
4.12.3	Deleting and Undeleting Globus Sites.....	37
4.13	Globus Site Administrators (optdbcli).....	37
5	Global Variables and Constants	
	optdbcli/optdbsrv	38
5.1	Global Constants optdbcli/optdbsrv	38
5.2	Global Variables optdbcli/optdbsrv.....	39
6	Global Functions optdbcli/optdbsrv	42
7	Server Connection Function Reference	43
7.1	Exchange Data With Client	43
8	Client Connection Function Reference.....	44
8.1	Exchange Data With Server.....	44
9	Web Application.....	45
9.1	Accessing the Web Application.....	45
9.1.1	Data Files.....	46
9.2	Auxiliary_File_Type.....	47
9.3	Global Variables and Constants	47
9.4	Web Application Functions.....	47
9.5	Scanner.....	47
9.5.1	Variables.....	47
9.5.2	Functions	47
9.6	Parser	48
9.6.1	Functions	48
10	Software Installation	49
10.1	Normal Software Installation	50
10.2	Hidden Software Installation	50
10.3	Local Software Installation	51
10.4	Remote Software Installation.....	51
11	TLS Connections	52
11.1	Preprocessor Macros for TLS.....	52
11.2	Types for TLS	52
11.3	Variables for TLS.....	52
11.4	TLS Functions	53
12	X509_Cert_Type Reference	56
12.1	Data Members	56
12.2	Member Functions	56

13	Proxy Credential Function Reference	58
14	Scanner Server	59
14.1	Scanner Server Variables	59
14.2	Scanner Server Functions	59
15	Parser Server	60
15.1	Parser Server Symbols, Rules and States	60
15.2	Parser Server Functions	60
16	Scanner Client	61
16.1	Scanner Client Variables	61
16.2	Scanner Client Functions	61
17	Parser Client	62
17.1	Parser Client Symbols, Rules and States	62
17.2	Parser Client Functions	62
18	Entry_Type Reference	63
18.1	Data Members	64
18.2	Member Functions	64
18.2.1	Constructors	64
18.2.2	Operators	65
18.2.3	Other Functions	65
18.3	Comparison Classes for Entry_Type	65
18.3.1	Operators for Comparison Classes for Entry_Type	65
19	Distinguished_Name_Type Reference	66
19.1	Data Members	66
19.2	Member Functions	66
20	Response_Type Reference	68
20.1	Data Members	69
20.2	Member Functions	71

21	Scan_Parse_Parameter_Type Reference	72
21.1	Data Members	73
21.2	Member Functions	75
21.2.1	Constructors and Destructor	75
21.2.2	Functions for Managing Users	75
21.2.3	Functions for Entries	75
21.2.3.1	Adding Entries	75
21.2.3.2	Modifying Entries	77
21.2.3.3	Deleting Entries	77
21.2.3.4	Showing Entries	78
21.2.3.5	Fetching Entry Information	78
21.2.4	Functions for Prerequisites	78
21.2.5	Database Functions	79
21.2.6	Globus Site Functions	79
21.2.7	Functions for Privileges	80
21.2.8	Environment Shellscript Functions	80
21.2.9	Functions for GPG Keys	80
21.2.10	Session Data Functions	81
21.2.11	Other Member Functions	81
22	Globus_Site_Info_Type Reference	82
22.1	Data Members	82
22.2	Member Functions	82
23	Parser Function Reference	83
24	GPG_Key_Pair_Type Reference	84
24.1	GPG_Key_Pair_Type Data Members	84
24.2	GPG_Key_Pair_Type Member Functions	84
25	Session_Data_Type Reference	85
25.1	Data Members	85
25.2	Session_Data_Type Member Functions	85
26	Utility Function Reference	86
27	Thread Function Reference	87
27.1	Remote Installation	87
27.2	GPG Key Distribution	88
27.3	Cull Database	88

28	Hidden Installation Reference (scrinstl)	89
28.1	Global Variables	89
28.2	Global Functions	89
28.3	Distinguished_Name_Type	90
28.3.1	Distinguished_Name_Type Data Members	91
28.3.2	Distinguished_Name_Type Member Functions	91
28.4	Parameter_Type	91
28.4.1	Parameter_Type Data Members	91
29	Auxiliary Program and Shellscript Reference	92
29.1	Start Remote Installation	92
29.2	Kill Process	93
29.3	Install GPG Key Pair	93
29.4	Change Owner for Installed Files	93
29.5	Generate Passwords or Passphrases	93
29.5.1	Options	93
29.5.2	Global Variables	95
29.5.3	Functions	96
29.6	Purge Installed Files	96
30	Database Tables and Views	97
30.1	Users	97
30.2	User_Info	97
30.3	Certificates	97
30.4	Privileges	98
30.5	Delegates	99
30.6	Entries	99
30.7	Prerequisites	100
30.8	Globus_Sites	100
30.9	Key_Pairs	100
30.10	Public_Keys	101
30.11	Session_Data	101
30.12	Environment_Shellscripts	101
30.13	Globus_Sites_Admins	101
	Glossary	102

Lists of Source Files	104
Client/Server Source File List	104
Client/Server CWEB Files	104
Client/Server Shellscripts	106
Hidden Installation Source File List	107
Hidden Installation CWEB Files	107
Hidden Installation Shellscripts.....	107
Web Application Source File List.....	107
Application CWEB Files.....	108
Application Shellscripts	108
 GNU Free Documentation License	 109
 GNU General Public License	 117
 Variable Index.....	 128
 Data Type Index.....	 136
 Function Index	 137
 Concept Index.....	 141

List of Figures

1. Create/Edit/Display Entries with client program <code>dbsrvcli</code> directly	3
2. Create/Edit/Display Entries with web application <code>optwbsrv</code>	4
3. Web Application	5
4. Sample Dialogue	20
5. Entries Table Row Without Encrypted Data	22
6. Entries Table Row With Encrypted Data	23

1 Introduction

The OptiNum Grid Installer consists of the following packages:

- A client/server application ‘`optdbcli/optdbsrv`’. See Chapter 7 [Server Connection Function Reference], page 43, and Chapter 8 [Client Connection Function Reference], page 44.
- A web application ‘`optwbsrv`’. See Chapter 9 [Web Application], page 45.
- A program for “hidden installation” ‘`scrinstl`’. See Section 10.2 [Hidden Software Installation], page 50.

Miscellaneous auxiliary programs, shellscripts, etc., are included in these packages.

The three individual packages are distributed in the form of compressed archive files (*zipped tar files* or “tarballs”):

Client/Server	<code>optdbcli/optdbsrv</code>	<code>dbsrvcli-1.0.tar.gz</code>
Web Application	<code>optwbsrv</code>	<code>optwbsrv-1.0.tar.gz</code>
Hidden Installation	<code>scrinstl</code>	<code>scrinstl-1.0.tar.gz</code>

1.1 Building and Installing

1.1.1 Server and Client

Please note that a functioning version of the server ‘`optdbsrv`’ will not be built if the MySQL-client library ‘`libmysqlclient`’ is not available. In this case, only the client ‘`optdbcli`’ will be built. (A non-functioning version of ‘`optdbsrv`’ that issues an error message and exits will be built.)

In the future, the client may also require a MySQL database for handling *session data*. See Chapter 25 [Session_Data_Type Reference], page 85. If ‘`libmysqlclient`’ is not available, then the client should still be built, but without support for session data.

Unpack the distribution ‘`dbsrvcli-1.0.tar.gz`’:

```
tar xvfz dbsrvcli-1.0.tar.gz
```

Change the working directory to `./dbsrvcli-1.0`:

```
cd dbsrvcli-1.0
```

Run the `configure` script:

```
./configure --prefix=<installation directory>
```

For example, to install the package in the working directory, i.e., the ‘`dbsrvcli`’ directory itself, `configure` can be called like this:

```
./configure --prefix='pwd'
```

or like this:

```
./configure --prefix='pwd'
```

The executable files will be installed in a subdirectory ‘`bin`’ of the directory specified as the argument to the ‘`--prefix`’ option. If the latter is not used, they will instead be installed

in `/usr/bin`, which requires `root` privileges. An *absolute path* must be specified, that is, not a *relative path* containing `.` or `..`.

Build the package:

```
make
```

Install the package:

```
make install
```

1.1.1.1 Database Setup

This is only needed for the server, since the client doesn't access the MySQL database.

The shellscript `'crtsrvdb.sh'` in `'dbsrvcli-1.0/src/DATABASE'` can be used to create the database `'dbsrvcli'`. It uses the file `'crtsrvdb.sql'`, which contains SQL code.

`'crtsrvdb.sh'` can be called with or without arguments. If it is called with no arguments or with `'0'`, `"0"` or `'0'` as the first argument, `'dbsrvcli'` will be the name of the database created, which is what the installer package requires. For testing purposes, another name may be passed as the first argument.

Additional arguments are "users" that will be inserted in order into the `'Users'` table. See Section 30.1 [Users Database Table], page 97. Please note that in this case, `'0'` (or `'dbsrvcli'`) must be passed to `'crtsrvdb.sh'` as its first argument, if the database created is to have the default name `'dbsrvcli'`.

If the database to be created already exists, `'crtsrvdb.sh'` exits with an error.

`'crtsrvdb.sh'` also creates a user `'OptiNum'`, unless this user already exists. If it does, a message is issued and the shellscript continues execution.

1.1.1.2 Prerequisites

The client program `'optdbcli'` has fewer prerequisites than the server program `'optdbsrv'`. If the client doesn't require a package, this is indicated in the following list.

g++ The GNU C++ Compiler.

GNU Bison

 The GNU version of the compiler generator `yacc`

Flex A free version of the scanner generator `lex`

GNUTLS A C library that implements the TLS protocol (formerly SSL)

libtasn1 A C library for processing data

 in "Abstract Syntax Notation One" (ASN.1) format. Used by GNUTLS.

libgcrypt11 The GNU Crypto Library.

 Used by GNUTLS

CWEB A *literate programming* tool for the source code. CWEB is included in both versions of the Installer package and built when `'make'` is invoked.

MySQL Database software. Server only.

For the documentation, the following packages are also required.

Texinfo The documentation tool used for creating this handbook. It is currently not required for the client-only version. However, it may be in the future.

T_EX A package for computer typesetting. Used by **Texinfo**.

On a GNU/Linux system where **yast2** or **zypper** is available, it should be possible to install the packages listed here with no difficulty.

1.2 Communication Among the Programs

**Create/Edit/Delete/Display Entries
with client program `dbsrvcli` directly**

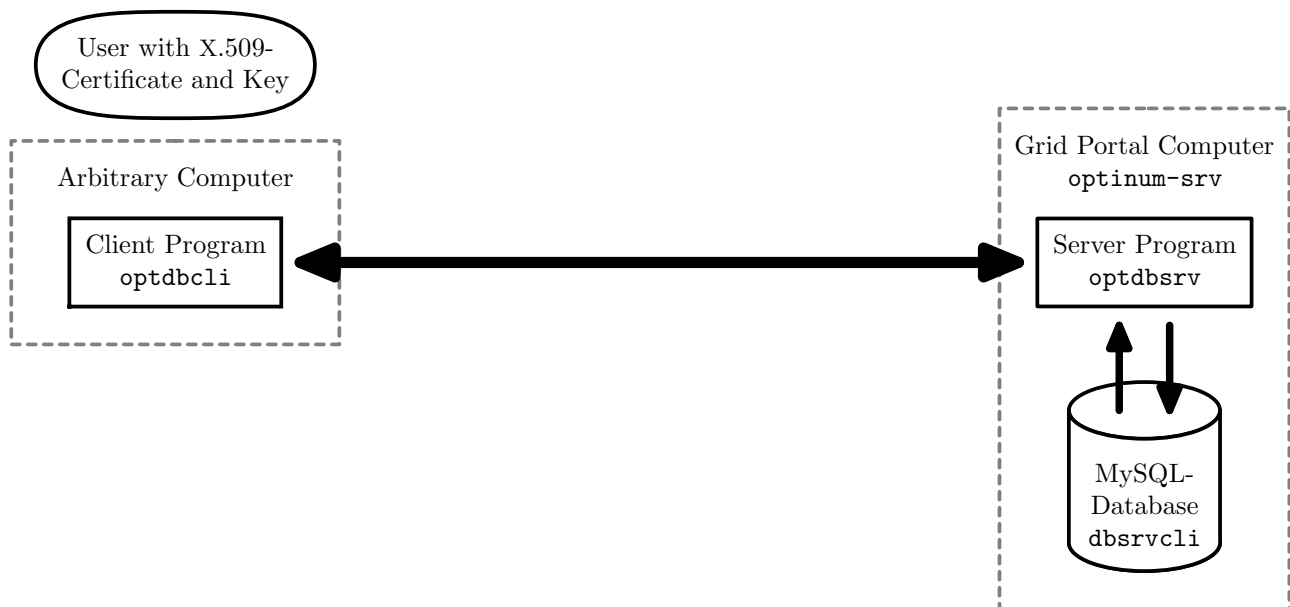


Fig. 1

Create/Edit/Delete/Display Entries with web application optwbsrv

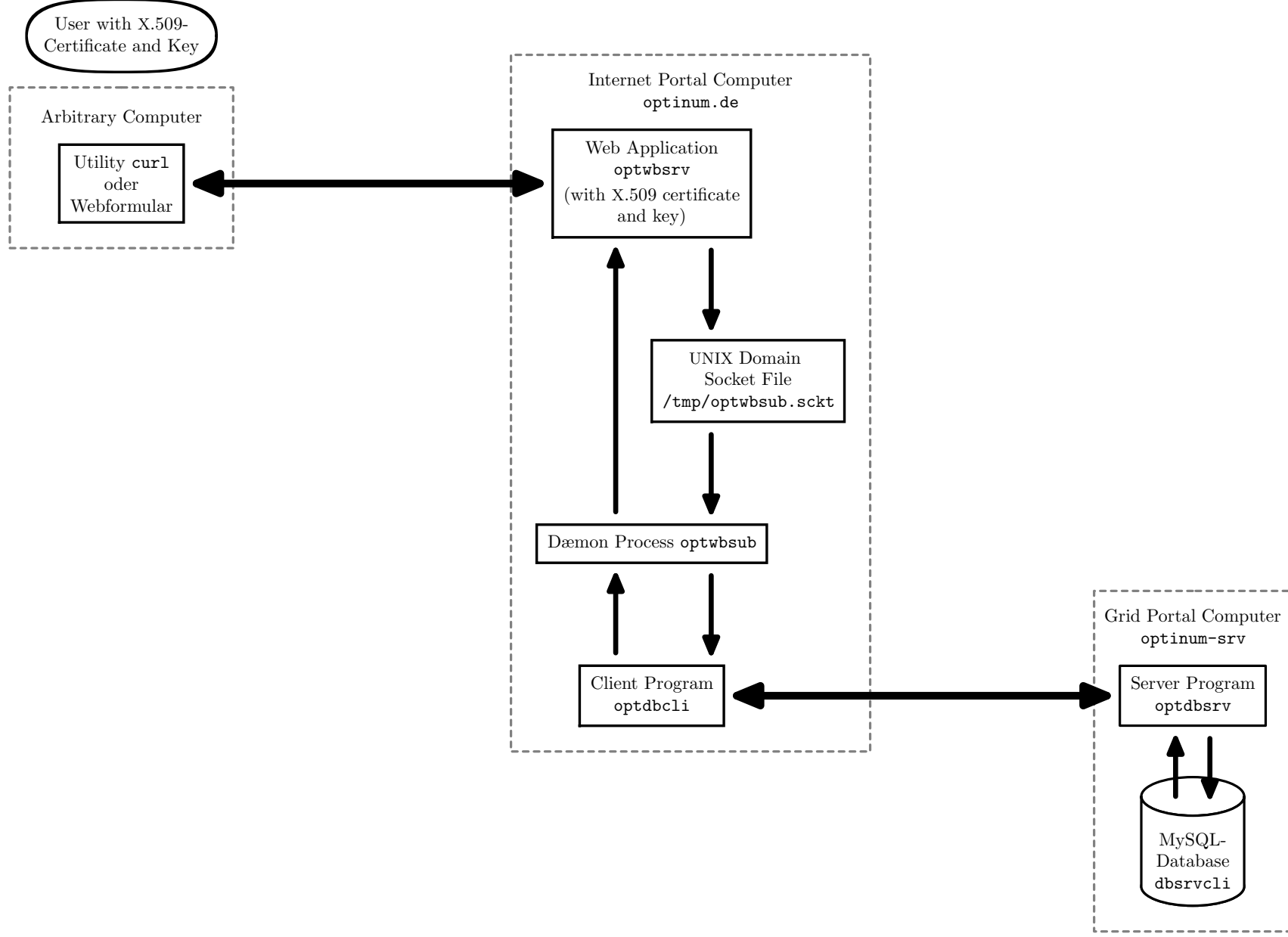


Fig. 2

Web Application

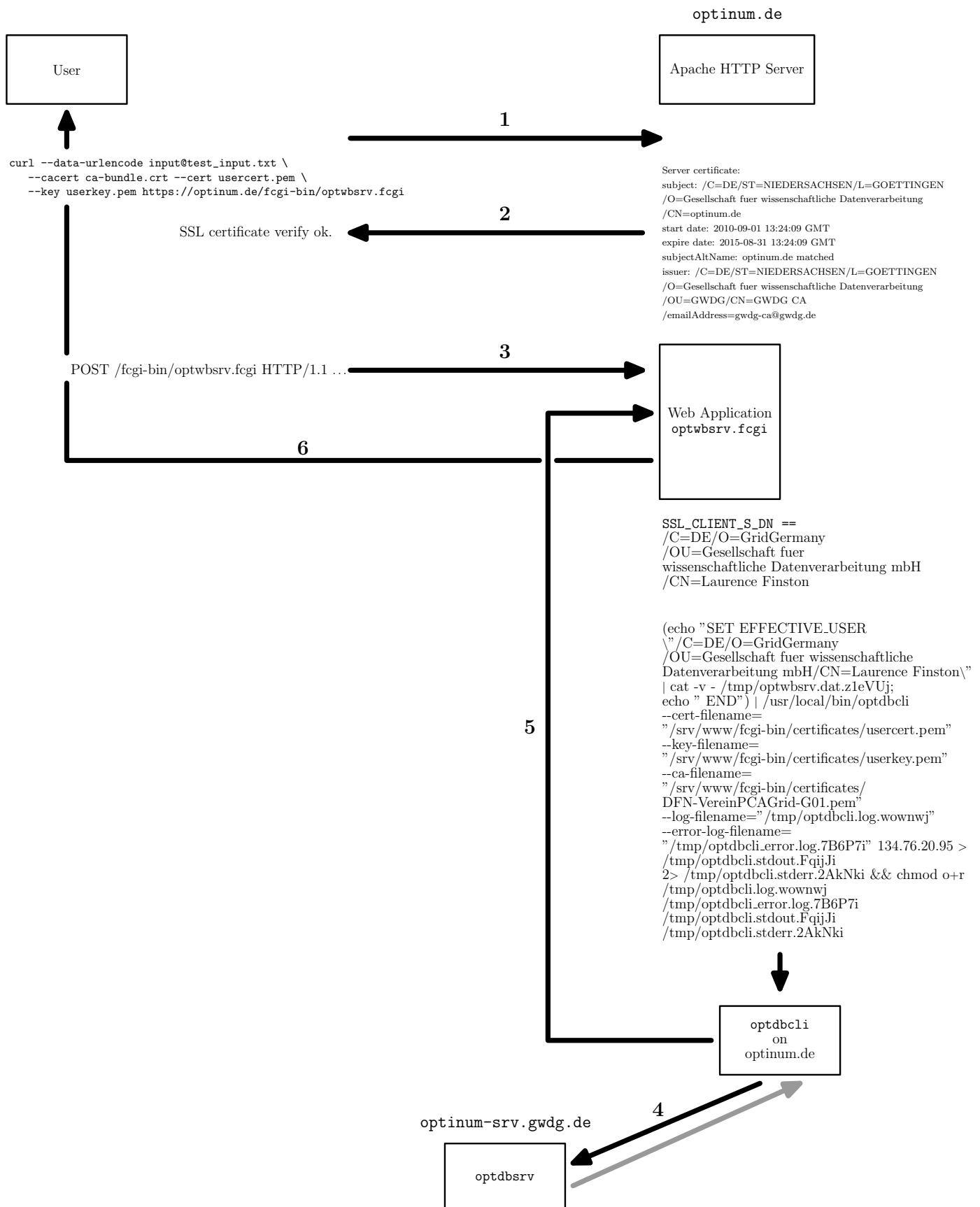


Fig. 3

2 Invoking optdbcli/optdbsrv

In order to maintain consistency, the server and the client share the program code for their command-line options. The code is located in the file `'cmdlnopt.web'`.

Some options are only used by the server or the client, while others are interpreted differently by the two programs.

In the following synopses and list of options, `'ARG'` without square brackets indicates a required argument, `'[ARG]'` an optional one.

```
optdbsrv [options] [port number authenticated] \
        [port number non-authenticated]
```

```
optdbcli [options] IP-address_of_server [input filename] \
        [server port number]
```

`'optdbsrv'` has no required arguments. The defaults for `'[port number authenticated]'` and `'[port number non-authenticated]'` are 5556 and 5557, respectively. See Chapter 5 [Global Variables and Constants optdbcli/optdbsrv], page 38.

`'optdbcli'` has a single required argument, namely the IP address of the server to which it should connect. The IP address may be specified using digits and periods, e.g., `'134.76.20.95'` or as the name, e.g., `'optinum-srv.gwdg.de'`. The default for the `'[server port number]'` argument is 5556, if the connection uses X.509 certificate authentication and 5557 otherwise.

2.1 Command Line Arguments and Options

The command-line options are processed using the C library function `'getopt_long_only'` and are handled as described in its GNU/Linux manual page (*man page*). Single hyphens can be used instead of double hyphens and unambiguous abbreviations of options are permitted, e.g., `'--bi'` or `'-bi'` for `'--bison-trace'`.

Required arguments may appear following the option with intervening whitespace or they may be specified like this: `'--option=ARG'`. An optional argument must be specified in the second way, otherwise, it will be interpreted as a non-option argument.

Most of the command-line options are associated with *global variables*. See Section 5.2 [Global Variables optdbcli/optdbsrv], page 39, for more information.

2.1.1 Help Options

```
--version      Server and client. Outputs program name with version number and copyright
                information and exits

--help         Server and client. Outputs a help message and a list of options and exits.
```

2.1.2 Connection Options

```
--port-auth ARG
--port-non-auth ARG
                Server and client. The ports on which 'optdbsrv' listens for connections from
                the client. '--port-auth' is for connections where the user provides an X.509
```

certificate and key pair and ‘--port-non-auth’ is for connections where the user provides a *proxy credential*. The values used by the server and client must match. Normally, there should be no need to specify values explicitly, unless the default ports (5556 for ‘--port-auth’ and 5557 for ‘--port-non-auth’) are being used for something else on the server-side.

2.1.3 X.509 Authentication/Authorization Options

--cert-filename ARG

--key-filename ARG

Server and client. Filenames of the X.509 certificate and key files. Client: Authentication and authorization are performed using either an X.509 certificate and a private key or a *proxy credential* (see option ‘--proxy-credential’, below). Server: The server identifies itself to the client using an X.509 certificate/key pair. The default files are ‘hostcert.pem’ and ‘hostkey.pem’, respectively, in the directory from which ‘optdbsrv’ is called.

--ca-filename ARG

Server and client. Filename of the *CA certificate*, i.e., the X.509 certificate of the *certification authority* that issued the user certificate. In the future, it should be possible to use this option multiple times so that the server or client can verify certificates against more than one CA certificate. However, this has not yet been implemented.

--crl-filename ARG

Server and client. Revocation list for the CA certificate (see ‘--ca-filename’ above). **Please note:** This option currently has no effect, since the use of revocation lists has not yet been implemented.

--cert-format

Certificate format, i.e., X.509 or some other format. Currently, this option isn’t used and only X.509 certificates are supported.

--key-encrypt

Currently not used. The key files for X.509 certificates can be encrypted. At present, the package doesn’t support the use of encrypted keys.

--proxy-credential ARG

Client only. A proxy credential containing an X.509 certificate/key pair issued by a user’s certificate, i.e., not a certification authority, and usually with a short period of validity. A proxy credential can be generated using the shellscript ‘gen_proxy.sh’, which is included in the distribution. See [Client/Server (optdbcli/optdbsrv) Shellscripts], page 106.

--grid-mapfile ARG

Normally, server only. The grid-mapfile to be used for determining the validity of the *Distinguished Name* extracted from the proxy credential, when one is used by the client for authentication/authorization, and for associating a Distinguished Name with a user name. The default value is ‘/etc/grid-security/grid-mapfile’.

For testing, this option can be useful for the client as well. However, since there is a risk of misuse, it is disabled on the the hosts ‘optinum-srv.gwdg.de’, ‘rocks-goegrid.gwdg.de’ ‘optinum.de’ and ‘fau136a.informatik.uni-erlangen.de’. On all other hosts, it is permitted.

2.1.4 Input and Output Options

--input-filename ARG

Client only. The input file contains commands that are sent verbatim to the server without processing by the client.

--output-filename ARG

Currently not used.

--log-filename ARG

--error-log-filename ARG

Client only. Filenames for log and error log files.

2.1.5 Software Installation Options

These options are for the client only.

--package-name ARG

Specifies the name of a package to be installed, if used in combination with the ‘--install’ option. If used in combination with the ‘--fetch’ option, the name of a package, for which download and installation information should be extracted from the database. See below for information on the ‘--install’ and ‘--fetch’ options.

--package-version [ARG]

If the option ‘--package-name’ is specified more than once, and a version number is specified for at least one of the packages, then ‘--package-version’ must be specified exactly as many times as ‘--package-name’, in order to maintain the correspondence between package name and package version. In this case, ‘--package-version’ can be specified without an argument as a placeholder.

Please note: If an argument is specified, it must be specified like this: ‘--package-version=ARG’. If it were specified as ‘--package-version ARG’, the argument would be interpreted as a non-option argument.

--fetch Causes the server to extract information for the packages specified with ‘--package-name’ and possibly ‘--package-version’ from the database and send it to the client.

This information consists of the package name, the package version, if any, the download URL and/or command and an installation script, if present. However, the package is not downloaded or installed, unless the ‘--install’ is also specified. ‘--install’ implies ‘--fetch’, so it is redundant to use ‘--fetch’ together with ‘--install’. However, it does no harm.

--install

Causes the packages specified with ‘--package-name’ and possibly ‘--package-version’ to be installed. If the ‘--no-resolve-dependencies’

is *not* used, the software prerequisites of these packages will also be installed. ‘--resolve-dependencies’ can be explicitly specified, but resolving dependencies is currently the default behavior.

--no-install

Suppresses installation of packages specified with ‘--package-name’ and possibly ‘--package-version’. This option will normally not be needed, since ‘--install’ must be specified explicitly for packages to be installed. However, it may be useful for testing, since this option will suppress installation if it appears in an invocation of ‘optdbcli’ following ‘--install’. (The converse is also true, i.e., installation will be performed if ‘--install’ follows ‘--no-install’.)

--reinstall

Forces reinstallation of the package or packages specified using the ‘--package-name’ option. If one or more of these packages has prerequisites, and neither ‘--no-prerequisites-reinstall’ nor ‘--no-resolve-dependencies’ has been specified, reinstallation will also be forced for the prerequisites.

--resolve-dependencies

--no-resolve-dependencies

If ‘--resolve-dependencies’ is specified, the software prerequisites for the packages specified with ‘--package-name’ are also fetched or installed. If ‘--no-resolve-dependencies’ is specified, this is suppressed. The default is to resolve dependencies.

--prerequisites-reinstall

--no-prerequisites-reinstall

Specify that prerequisites should be reinstalled or not. These options can be used if the default behavior is not desired. This is for prerequisites to be handled like the packages specified using the ‘--package-name’ option directly, i.e., they are reinstalled or not, according to whether the ‘--install’ or ‘--reinstall’ option has been used. However, it may often be undesirable for prerequisites to be reinstalled, even when reinstalling packages that depend on them.

Currently, it is only possible to specify how all prerequisites in a given run should be handled; it is not possible to make finer distinctions, i.e., that the prerequisites of package A should be reinstalled, but not the prerequisites of package B.

--install-directory ARG

The directory in which software packages are installed (in their own subdirectories). Default is ‘\$HOME/Installer/’.

--download-fail-continue

--install-fail-continue

If used, ‘optdbcli’ will continue to execute if the command for downloading and/or installing a package fails. The default behavior is for the ‘optdbcli’ to exit with a non-zero return value. Continuing may be useful when installing multiple packages where at least package that’s installed later doesn’t depend on all previous packages having been downloaded and installed successfully.

--create-links

--no-links

Create or suppress the creation of *symbolic links* after installing one or more software packages.

As of 2012.02.27, links are created by default. The author plans to change this. By default, links are created in '\$HOME/bin/', '\$HOME/include/', '\$HOME/lib/' and/or '\$HOME/share/', as appropriate. A directory other than '\$HOME/' may be specified using the '--link-directory' option (see below).

--link-directory ARG

The directory where links are created in subdirectories 'bin/', 'include/', 'lib/' and 'share/', as appropriate. The default is '\$HOME'. See '--create-links' and '--suppress-links', above.

2.1.6 Debugging Options

--trace [ARG]

Server and client. Sets the debugging value for all functions that have a `DEBUG` variable and which call the function `set_debug_level`, which is most of them. See Chapter 26 [Utility Function Reference], page 86. It also sets the debugging level for the parsing functions `yyparse` and `zzparse`. See Chapter 15 [Parser Server], page 60, and Chapter 17 [Parser Client], page 62. Therefore, the use of the '--bison-trace' option (see below) is redundant if '--trace' is used with `ARG > 0`.

`ARG > 0` Debugging is turned on.

`ARG < 0` Debugging is turned off.

`ARG = 0` Debugging level is left unchanged.

--bison-trace [ARG]

--parser-trace [ARG]

Server and client. These options are synonymous. They enable debugging output for the parsing functions `yyparse` and `zzparse`. See Chapter 15 [Parser Server], page 60, and Chapter 17 [Parser Client], page 62.

No argument

Debugging output automatically generated by Bison

`ARG ≤ 0` No debugging output

`ARG = 1` Programmed debugging output from actions only

`ARG > 1` Automatically generated and programmed debugging output

Invalid (non-integer) argument

No debugging output

--flex-trace

--scanner-trace

Server and client. These options are synonymous. They enable debugging output for the scanning functions `yylex` and `zzlex`. See Chapter 14 [Scanner Server], page 59, and Chapter 16 [Scanner Client], page 61.

--save-temp-files

Server and client. For debugging. Disables deletion of temporary files created by the programs.

--sleep ARG

Server and client. For debugging and testing threads in the server.

2.1.7 Session Data Options

--session-id ARG

Client only. This option primarily intended for use with the web application in order to maintain a “state” encompassing multiple invocations of the client program ‘optdbcli’. It must be used in combination with the ‘**session_id**’ command in the input sent to the server; otherwise it has no effect. If the session ID ‘**ARG**’ doesn’t match the session ID, which the server sends back to the client, the client exits with an error. See Chapter 25 [Session_Data_Type Reference], page 85.

2.1.8 Alphabetical List of Options

--bison-trace

Section 2.1.6 [Debugging Options], page 10.

--ca-filename

Section 2.1.3 [X.509 Authentication/Authorization Options], page 7.

--cert-filename

Section 2.1.3 [X.509 Authentication/Authorization Options], page 7.

--cert-format

Section 2.1.3 [X.509 Authentication/Authorization Options], page 7.

--create-links

Section 2.1.5 [Software Installation Options], page 8.

--crl-filename

Section 2.1.3 [X.509 Authentication/Authorization Options], page 7.

--download-fail-continue

Section 2.1.5 [Software Installation Options], page 8.

--error-log-filename

Section 2.1.4 [Input and Output Options], page 8.

--fetch Section 2.1.5 [Software Installation Options], page 8.

--flex-trace

Section 2.1.6 [Debugging Options], page 10.

--grid-mapfile

Section 2.1.3 [X.509 Authentication/Authorization Options], page 7.

--help Section 2.1.1 [Help Options], page 6.

--input-filename

Section 2.1.4 [Input and Output Options], page 8.

- `--install`
Section 2.1.5 [Software Installation Options], page 8.
- `--install-directory`
Section 2.1.5 [Software Installation Options], page 8.
- `--install-fail-continue`
Section 2.1.5 [Software Installation Options], page 8.
- `--key-encrypt`
Section 2.1.3 [X.509 Authentication/Authorization Options], page 7.
- `--key-filename`
Section 2.1.3 [X.509 Authentication/Authorization Options], page 7.
- `--link-directory`
Section 2.1.5 [Software Installation Options], page 8.
- `--log-filename`
Section 2.1.4 [Input and Output Options], page 8.
- `--no-install`
Section 2.1.5 [Software Installation Options], page 8.
- `--no-links`
Section 2.1.5 [Software Installation Options], page 8.
- `--no-prerequisites-reinstall`
Section 2.1.5 [Software Installation Options], page 8.
- `--no-resolve-dependencies`
Section 2.1.5 [Software Installation Options], page 8.
- `--output-filename`
Section 2.1.4 [Input and Output Options], page 8.
- `--package-name`
Section 2.1.5 [Software Installation Options], page 8.
- `--package-version`
Section 2.1.5 [Software Installation Options], page 8.
- `--parser-trace`
Section 2.1.6 [Debugging Options], page 10.
- `--port-auth`
Section 2.1.2 [Connection Options], page 6.
- `--port-non-auth`
Section 2.1.2 [Connection Options], page 6.
- `--prerequisites-reinstall`
Section 2.1.5 [Software Installation Options], page 8.
- `--proxy-credential`
Section 2.1.3 [X.509 Authentication/Authorization Options], page 7.

`--reinstall`
 Section 2.1.5 [Software Installation Options], page 8.

`--resolve-dependencies`
 Section 2.1.5 [Software Installation Options], page 8.

`--save-temp-files`
 Section 2.1.6 [Debugging Options], page 10.

`--scanner-trace`
 Section 2.1.6 [Debugging Options], page 10.

`--session-id`
 Section 2.1.7 [Session Data Options], page 11.

`--sleep` Section 2.1.6 [Debugging Options], page 10.

`--trace` Section 2.1.6 [Debugging Options], page 10.

`--version`
 Section 2.1.1 [Help Options], page 6.

2.2 Variable Reference

The following variables and functions are defined in ‘`cmdlnopt.web`’.

`vector<string> cert_filenames` [Global variables]
`vector<string> key_filenames`

`vector<string> ca_filenames` [Global variables]
`vector<string> crl_filenames`
 These variables are used by the server only.

2.3 Function Reference

`int process_command_line_options (int argc, char* argv[])` [Function]
`int show_filename_arguments (void)` [Function]
 Show filename arguments for certificates, keys, CAs (*Certification Authorities*), and CRLs (*Certificate Revocation Lists*). For debugging.

3 Using the Server Program `optdbsrv`

Starting the server program ‘`optdbsrv`’ is simpler than starting the client. Normally, it should suffice to start it without any arguments. For testing purposes, it can be called like this:

```
optdbsrv
```

In this case, it will listen on port 5556 for connections from clients using authentication with X.509 certificates and on port 5557 for clients that use an *X.509 proxy credential* to identify themselves. (Other ports can be specified with the ‘`--port-auth`’ and ‘`port-non-auth`’ options, if desired. See Section 2.1.2 [Connection Options], page 6.) The defaults for the certificates used are ‘`DFN-VereinPCAGrid-G01.pem`’ for the CA certificate and ‘`hostcert.pem`’ and ‘`hostkey.pem`’ for the certificate and key used by the server to identify itself to the client.

In production use, the server will normally run continuously in the background as a *daemon* process and should therefore be invoked as follows:

```
nohup ./optdbsrv 2>&1 &
```

The final ampersand places the process in the background while ‘`nohup`’ ensures that it will continue to run after the user that starts it logs out. It would be a good idea to create a user *without* root permissions for the purpose of running ‘`optdbsrv`’, and perhaps for other tasks connected with the Installer package.

Any output written by ‘`optdbsrv`’ to standard output or standard error will be redirected to ‘`nohup.out`’. (‘`2>&1`’ redirects standard error to standard output while the latter is redirected to `nohup.out`.) Whoever is responsible for administering ‘`optdbsrv`’ should ensure that this file is truncated once in a while, so it doesn’t grow too large.

For testing purposes, ‘`optdbsrv`’ can simply be invoked from the command line (i.e., as a foreground process):

```
./optdbsrv
```

The options ‘`--bison-trace`’, ‘`--flex-trace`’ and/or ‘`--trace`’ can be specified for more debugging output. (In addition, ‘`--parser-trace`’ and ‘`--scanner-trace`’ are synonyms for ‘`--bison-trace`’ and ‘`--flex-trace`’, respectively.)

The option ‘`--trace`’, with no argument or a numerical argument > 0 , causes debugging information for most if not all of the functions that are called while ‘`optdbsrv`’ is running. See Chapter 2 [Invoking `optdbcli/optdbsrv`], page 6, for more information about these options and others that can be used when invoking ‘`optdbsrv`’. See Section 2.1.6 [Debugging Options], page 10, for more information.

4 Using the Client Program `optdbcli`

Most users will only require the client program `'optdbcli'` and won't need to build, install or run the server program `'optdbsrv'`, as long as there's an instance of the latter running somewhere to which the client can connect.

`'optdbcli'` is used for several different purposes:

1. Storing information in the remote software database administered by `'optdbsrv'`
2. Initiating a remote software installation on the cluster front-end computer associated with a Globus site
3. Fetching information about software packages from the remote database and installing the software on the local machine where `'optdbcli'` is running
4. Fetching and displaying information from various tables of the remote database
5. For users who are responsible for maintaining information about Globus sites, certain administrative tasks can be performed using `'optdbcli'` instead of accessing the database using a MySQL client (such as `mysql`).

In the first two cases, commands, either from a file or passed to `'optdbcli'` via standard input, are sent by `'optdbcli'` to the server program `'optdbsrv'`. They are not processed in any way by `'optdbcli'`.

In the third case, no input file or input from a pipe is used, because in the “normal” case of a remote installation, `'optdbcli'` is called as part of a *Globus job* and there is no way of passing it input in either of these ways, at least not at present. Instead, the command line options `'--install'` or `'--reinstall'` together with the options `'--package-name'` and optionally `'--package-version'` are used to control the operation of `'optdbcli'`.

Let's assume that the server `'optdbsrv'` is running on `'optinum-srv.gwdg.de'` and that the file `'input.txt'` contains commands for telling the server to store information about software packages and how to install them in the database for the user who's invoking the client. The contents of `'input.txt'` can be sent to the server in the following, equivalent ways:

```
optdbcli --input-file input.txt optinum-srv.gwdg.de
optdbcli --input-file=input.txt optinum-srv.gwdg.de
optdbcli --input-file "input.txt" optinum-srv.gwdg.de
optdbcli optinum-srv.gwdg.de input.txt 5556
cat input.txt | optdbcli optinum-srv.gwdg.de
```

Arguments may be quoted to prevent expansion by the shell (see Section “Shell Expansion” in *The GNU Bash Manual*). See Chapter 2 [Invoking `optdbcli/optdbsrv`], page 6, for details about options and their arguments.

These invocations assume that the user's certificate and private key file and are in the default location, namely `'$HOME/.globus/usercert.pem'` and `'$HOME/.globus/userkey.pem'`, respectively. In addition, it assumes that the *CA certificate* (Certificate Authority certificate) is also in the default location, which is determined by means of conditionally compiled code. If these files are not in the default locations, they must be specified using the options `'--cert-filename'`, `'--key-filename'` and/or `'--ca-filename'`. In addition, the default filenames may be changed in the source code before building the package, if desired.

Please note that the default value for the path of the CA certificate is the empty string for platforms that are not among the ones referred to specifically in the conditionally compiled code! It will therefore normally be necessary to specify the CA certificate using the ‘`--ca-filename`’ option.

4.1 How the Client and the Server Communicate

When ‘`optdbcli`’ is invoked, input can be passed to it:

1. as the argument to the ‘`--input-filename`’ option
2. as a non-option filename argument following the name or IP address of the machine where the server program is running (e.g., ‘`optinum-srv.gwdg.de test_input.txt`’)
3. through a pipe to its standard input. In this case, the input need not be stored in a file, it could be a string argument to the `echo` command or the output of another command or program.

See Chapter 2 [Invoking `optdbcli`/`optdbsrv`], page 6.

‘`optdbcli`’ does not process these commands in any way, but passes them unexamined to ‘`optdbsrv`’, following the *handshake procedure*, i.e., authentication/authorization using an X.509 certificate/private key pair or an X.509 proxy credential, and the establishment of secure communication using the GNUTLS library, which implements the TLS 1.0 protocol.

‘`optdbcli`’ and ‘`optdbsrv`’ each contains its own *parser function*, `yyparse` for ‘`optdbsrv`’ and `zzparse` for ‘`optdbcli`’. See Section 17.2 [Parser Client Functions], page 62, and Section 17.2 [Parser Client Functions], page 62. A parser function in effect implements a “language” that a computer program “understands”, so to speak.

The parser functions `yyparse` and `zzparse` are generated by the program GNU Bison from input in a special format that generates *rules of grammar* and specifies actions to be taken when input to the program “matches” the rules. For more information about GNU Bison, parsing, etc., see Section “Bison” in *The GNU Bison Manual*.

`optdbcli` and `optdbsrv` have different parser functions and therefore each has its own “language”. They communicate by sending each other messages using each other’s language. That is, `optdbcli` sends messages to `optdbsrv` in `optdbcli`’s language and `optdbsrv` sends messages back to `optdbcli` in `optdbcli`’s. This implies that the programs send messages that they themselves wouldn’t be able to interpret, but there is no need for them to be able to do this.

The commands in the input that can be passed to `optdbcli` are in the “`optdbsrv` language”. Certain command-line options to `optdbcli` also cause it to send commands to `optdbsrv`. The latter then takes action on these commands, where appropriate, and sends commands back to `optdbcli`, which may react by sending more commands back to `optdbsrv`. This continues until both partners are finished sending commands to the peer. At this point, the connection is ended (normally by `optdbcli`, unless there’s an error). See Section 7.1 [Exchange Data With Client], page 43, and Section 8.1 [Exchange Data With Server], page 44.

4.2 How To Read Parser Rules

The following sections provide examples of commands that can appear in the input file. For complete information about the symbols, grammar rules and states of `yyparse`, i.e., the parser function for `optdbsrv`, see Chapter 15 [Parser Server], page 60.

This is an example of a parser rule:

```
⟨program⟩ → ⟨statement list⟩ END
```

Parser rules have a *left-hand* and a *right-hand side*, separated by the symbol ‘ \rightarrow ’. The items that appear in a parser rule are either *terminal* or *non-terminal symbols*. In this manual, terminal symbols, also known as *tokens*, are printed in uppercase in a fixed-width, typewriter-style font, like ‘END’ in this example. Non-terminal symbols are printed in lowercase in a roman font, and surrounded by angle braces, like ‘⟨program⟩’ and ‘⟨statement list⟩’ in this example.

Only a single, non-terminal symbol can ever appear on the left-hand side of a rule, like ‘⟨program⟩’ in this example. Non-terminal symbols can also appear on the right-hand sides of rules, like ‘⟨statement list⟩’ in this example, while terminal symbols can only appear on the right-hand side of parser rules.

In the reference sections Chapter 15 [Parser Server], page 60 and Chapter 17 [Parser Client], page 62 you will see that the terminal symbols all end in ‘_YY’ or ‘_ZZ’, respectively. This is because terminal symbols are implemented in the C code as preprocessor macros, the definitions appear in the header file generated by `bison` and the latter is included by various other source files. The suffixes ‘_YY’ and ‘_ZZ’ help to ensure that these names won’t interfere with the names of other entities in the source code, e.g., variables, functions, other preprocessor macros, etc. *In this section and other other sections containing examples of parser rules, the suffixes are omitted for the sake of better readability.*

The non-terminal symbol on the left-hand side is the “target” of the rule. It *maps to* the sequence of items on the right-hand side of the rule. When the ‘`optdbsrv`’ or ‘`optdbcli`’ receives input, the scanner function `yyscan` or `zzscan`, respectively, converts the stream of characters into a sequence of tokens (i.e., terminal symbols) and passes the latter to the parser function `yyparse` or `zzparse`, respectively.

The parser function matches the sequence of tokens to its set of rules *reducing* as it proceeds. If the sequence of tokens fails to match the grammar rules defined by the Bison code, this causes an error.

A single left-hand side can map to more than one right-hand side:

```
⟨download info⟩ → DOWNLOAD_URL | DOWNLOAD_COMMAND | ⟨download info encrypted⟩
```

The ‘|’ symbol stands for “or”. The previous example is thus equivalent to:

```
⟨download info⟩ → DOWNLOAD_URL
⟨download info⟩ → DOWNLOAD_COMMAND
⟨download info⟩ → ⟨download info encrypted⟩
```

That is, `DOWNLOAD_URL`, `DOWNLOAD_COMMAND`, and ‘⟨download info encrypted⟩’ all *reduce to* ‘⟨download info⟩’.

Many rules contain the special “symbol” ‘(Empty)’, which stands for “nothing” or “no symbol at all”, e.g.,

$$\langle \text{statement list} \rangle \longrightarrow (\text{Empty}) \mid \langle \text{statement list} \rangle \langle \text{statement} \rangle$$

This means that a $\langle \text{statement list} \rangle$ may be empty, i.e., map to no symbols at all, or it may map to a $\langle \text{statement list} \rangle$ followed by a $\langle \text{statement} \rangle$. This is an example of a *recursive rule*, where the non-terminal on the left-hand side also appears on the right-hand side of the rule. This example means that a $\langle \text{statement list} \rangle$ can be nothing or a $\langle \text{statement list} \rangle$ followed by a $\langle \text{statement} \rangle$. In this way, a $\langle \text{statement list} \rangle$ can be constructed from an unlimited number of statements, including no statements at all!

For example, input containing only `END` is valid for ‘`optdbsrv`’:

```
echo "END" | optdbcli localhost
```

‘`optdbcli`’ sends the input to ‘`optdbsrv`’. The scanner function `yyscan` converts the character ‘E’, ‘N’ and ‘D’ to the token ‘`END`’ (really ‘`END_YY`’, as explained above) and passes the latter to the parser function `yyparse`. From the point of view of `yyparse`, a $\langle \text{program} \rangle$ is a $\langle \text{statement list} \rangle$ followed by the terminal symbol `END`. The “nothing” preceding `END` matches the rule ‘ $\langle \text{statement list} \rangle \longrightarrow (\text{Empty})$ ’, reducing to $\langle \text{statement list} \rangle$, which, followed by the `END` token, in turn matches the rule ‘ $\langle \text{program} \rangle \longrightarrow \langle \text{statement list} \rangle \text{END}$ ’, ending execution successfully.

‘(Empty)’ can also be used for *options*, i.e., elements of commands which may appear or be left out, e.g.,

$$\langle \text{statement} \rangle \longrightarrow \text{SHOW CERTIFICATE USER } \langle \text{html optional} \rangle \\ \mid \text{SHOW CERTIFICATES ALL } \langle \text{html optional} \rangle$$

$$\langle \text{html optional} \rangle \longrightarrow (\text{Empty}) \mid \text{HTML}$$

If the option ‘`HTML`’ is used, the output is formatted as for a web page, using `HTML` commands. If left out, it is formatted as plain text.

Implementing options by means of rules with empty right-hand sides in this way usually works well, but occasionally it causes reduce/reduce conflicts in the grammar, whereby there is no obvious reason why it should occur in one case but not in another. In such cases, the grammar must be reformulated in order to remove the conflicts. See Section “Reduce/Reduce Conflicts” in *The GNU Bison Manual*, and Section “Bison Grammar Files” in *The GNU Bison Manual*, for more information about Bison grammars in general.

There is nearly a one-to-one correspondence between the commands a user writes and the parser rules for `yyparse`, except for the ones that process messages returned from ‘`optdbcli`’. Most of the terminal symbols correspond to keywords of the “`optdbsrv`” language, such as ‘`SHOW`’, ‘`HTML`’, ‘`FOR`’, ‘`TO`’, etc. It is not necessary to type them in uppercase, as in the descriptions of the parser rules; they can be typed either in uppercase or lowercase, but the cases cannot be mixed within a single word, e.g.,

```
show entries user
SHOW ENTRIES USER
SHOW entries USER
show ENTRIES user
etc.,
```

are all valid, but not `'sHow eNtrIES uSer'`.

The following terminal symbols, however, do not refer to keywords:

`'INTEGER'` An optional sign `'+'` or `'-'`, followed by one or more decimal digits 0–9.

`'STRING'` Strings may contain the alphanumerical characters and a selection of special characters. This selection may need to be extended depending on what characters are used in URLs, filenames, or other items that can appear in commands for `'optdbsrv'`. The selection is determined by the corresponding rules in the file `'scanner.web'`, from which the input file for Flex is generated.

Strings may be undelimited or delimited by double quotation marks (`'"`) or angle braces (`'<>'`).

`'SEMI_COLON'`

`'SLASH'` Punctuation characters in the input are converted to appropriately-named tokens. Currently, these are the only punctuation characters used in the input for `'optdbsrv'`.

Some other terminal or non-terminal symbols may appear in rules that do not correspond to commands which would ever be written by a user, but which are generated by `'optdbcli'` as responses and sent back to `'optdbsrv'`.

4.3 Sample Dialogue

The following is a sample dialogue between the client program ‘optdbcli’ and the server program ‘optdbsrv’, leaving out a couple of less interesting steps.

Storing Software Package Data

Client invocation: `./optdbcli --input-filename=filename optinum-srv.gwdg.de`

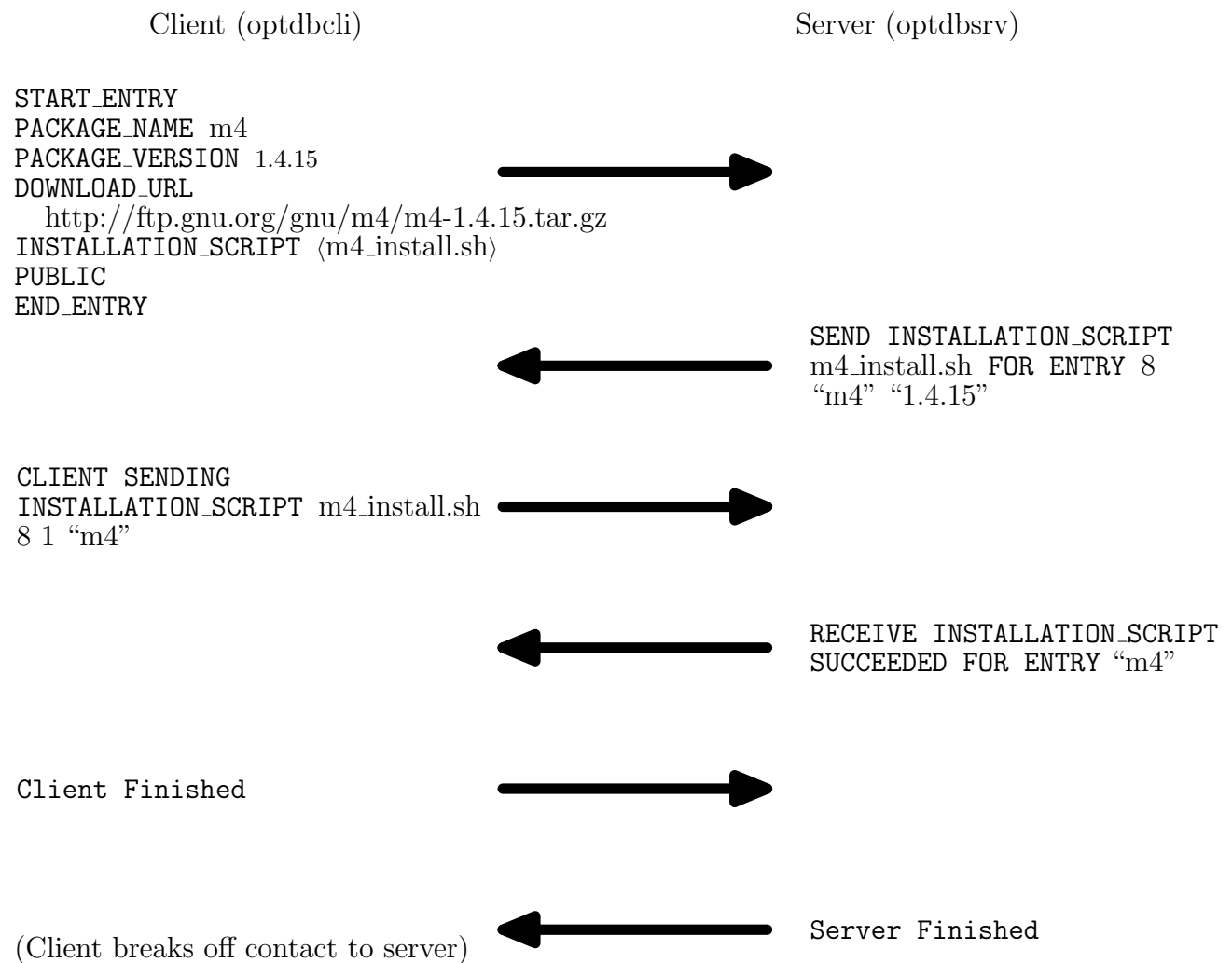


Fig. 4

4.4 Software Package Information

4.4.1 Adding Software Package Information

`<statement> → START_ENTRY <entry statement list> END_ENTRY`

```

⟨entry statement list⟩ → (Empty)
                        | ⟨entry statement list⟩ ⟨keyword value pair⟩
                        | ⟨entry statement list⟩ ⟨prerequisite statement⟩
                        | ⟨entry statement list⟩ PRIVATE
                        | ⟨entry statement list⟩ PUBLIC

```

```

⟨keyword value pair⟩ → PACKAGE_NAME STRING
                     | PACKAGE_VERSION STRING
                     | PACKAGE_VERSION INTEGER
                     | DOWNLOAD_URL STRING
                     | DOWNLOAD_URL_ENCRYPTED STRING
                     | DOWNLOAD_COMMAND STRING
                     | DOWNLOAD_COMMAND_ENCRYPTED STRING
                     | INSTALLATION_SCRIPT STRING
                     | INSTALLATION_SCRIPT_ENCRYPTED STRING
                     | OWNER STRING
                     | AUTHORIZATION STRING

```

```

⟨prerequisite statement⟩ → PREREQUISITE STRING
                        | PREREQUISITE STRING STRING

```

In the sample dialogue above, the user passes the following input to ‘`optdbcli`’, which sends it on to to ‘`optdbsrv`’:

```

START_ENTRY
PACKAGE_NAME      m4
PACKAGE_VERSION   1.4.15
DOWNLOAD_URL      http://ftp.gnu.org/gnu/m4/m4-1.4.15.tar.gz
INSTALLATION_SCRIPT "m4_install.sh"
END_ENTRY

```

```

END

```

Here, the ‘`DOWNLOAD_URL`’ keyword is used. When ‘`optdbcli`’ is invoked at a later time to initiate the installation of ‘`m4 1.4.15`’, `wget` is used to download the compressed archive file (“gzipped tar file” or *tarball*) ‘`m4-1.4.15.tar.gz`’ located at `http://ftp.gnu.org/gnu/m4/m4-1.4.15.tar.gz`.

This approach will not work for packages that cannot simply be download in this way. For example,

```

DOWNLOAD_COMMAND git clone git://git.savannah.gnu.org/m4.git

```

In this case, the download URL and the filename for the installation script don’t require quoting, since they don’t contain any whitespace or characters that might be interpreted as *wildcards* for expansion in the course of processing by the server. Nonetheless, it is safest to quote download URLs just in case. Double quotation marks, as above, or angle braces (`<>`) may be used.

Entries Table Row Without Encrypted Data

user_id	1
entry_id	9
package_name	m4
package_version	1.4.15
version_ctr	NULL
download_url	ftp://ftp.gwdg.de/pub/gnu/ftp/gnu/m4/m4-latest.tar.gz
download_command	NULL
maintainer_name	NULL
maintainer_email_address	NULL
private	0
checked_by_admin	0
installation_script	#!/bin/bash echo "m4 installation script" PACKAGE_NAME=m4 PACKAGE_VERSION=1.4.15 [...]
created	2011-09-09 07:30:56 UTC
last_modified	2011-09-09 07:30:56 UTC
encrypted	0
owner	NULL
download_url_encrypted	NULL
download_command_encrypted	NULL
installation_script_encrypted	NULL
authorization	NULL

Fig. 5

Entries Table Row With Encrypted Data

user_id	1
entry_id	11
package_name	hello
package_version	2.7
version_ctr	NULL
download_url	NULL
download_command	NULL
maintainer_name	NULL
maintainer_email_address	NULL
private	1
checked_by_admin	0
installation_script	NONE
created	2011-09-09 07:30:56 UTC
last_modified	2011-09-09 07:30:56 UTC
encrypted	1
owner	C=DE/O=GridGermany/OU=Gesellschaft fuer wissenschaftliche Datenverarbeitung mbH/CN=Laurence Finston
download_url_encrypted	-----BEGIN PGP MESSAGE----- Version: GnuPG v2.0.9 (GNU/Linux) hQIOA9va92BAyk8NEAf8C0SjONiWZnp5l9J3y76ETzlUduHP0+SMdhby Qgs09ppkWMS3f8pAzmbKONGr80V98bhX/ryJdiPZVilCrzmidogxX2AI [...] -----END PGP MESSAGE-----
download_command_encrypted	NULL
installation_script_encrypted	-----BEGIN PGP MESSAGE----- Version: GnuPG v2.0.9 (GNU/Linux) hQIOA9va92BAyk8NEAgAgXA5CM9GDE3Vt5jnhthmF4Dei9taUCShQrrc rlvAT+GI49bjYVsm5FaH3lJHccOyXVE8n/1JhD47BOPGXNwm6inAbMps [...] -----END PGP MESSAGE-----
authorization	-----BEGIN PGP MESSAGE----- Version: GnuPG v2.0.9 (GNU/Linux) hQIOA9va92BAyk8NEAf9GTwSyTxCZFVKH1jJByMDqVG0b0hAjP9 SyPr1CqK3dFtqreNIxz6WU2UT8HRIMvLu3HA8bqfV44H+7Ipf8Y [...] -----END PGP MESSAGE-----

Fig. 6

4.4.2 Deleting Software Package Information

```

<statement> → MARK AS DELETED ENTRY <entry string list>
              | MARK AS DELETED ALL ENTRIES FOR USER
              | UNDELETE ENTRY <entry string list>
              | UNDELETE ALL DELETED ENTRIES FOR USER
              | DELETE ENTRY <entry string list> FINALLY
              | DELETE ALL ENTRIES FOR USER FINALLY
              | DELETE PREREQUISITE <prerequisite string list> FOR ENTRY <entry string list>
              | DELETE PREREQUISITES FOR ENTRY <entry string list>

```

```

<entry string list> → (Empty) | <entry string list> STRING

```

```

<prerequisite string list> → (Empty) | <prerequisite string list> STRING

```

A software package entry can be marked for deletion, for example:

```
mark as deleted entry "hello" "2.7"
```

```
⇒
```

```
Marked entry 'hello 2.7' as deleted successfully.
```

```
show entry hello 2.7
```

```
⇒
```

```
*** User Entries ***
```

```

user_id:                1
commonName:             Laurence Finston
organizationalUnitName: Gesellschaft fuer wissenschaftliche \
                        Datenverarbeitung mbH
countryName:            DE
entry_id:               1
package_name:           hello
package_version:        2.7
version_ctr:            NULL
download_url: \
    ftp://ftp.gwdg.de/pub/gnu/ftp/gnu/hello/hello-2.7.tar.gz
download_command:       NULL
maintainer_name:        NULL
maintainer_email_address: NULL
private:                0
checked_by_admin:       0
installation_script:    PRESENT
created:                2012-05-21 12:16:25 UTC
last_modified:          2012-05-21 14:19:12 UTC
encrypted:              0
owner:                  NULL
download_url_encrypted: NULL

```

```

download_command_encrypted:    NULL
installation_script_encrypted: NULL

authorization:                NULL

deleted:                      1

```

The ‘deleted’ field is set to 1 (‘true’) and the timestamp in the ‘last_modified’ field is updated. The entry will now remain in the database for approximately 24 hours before being deleted by the function `cull_database` (see Section 27.3 [Cull Database], page 88). This function runs in a thread of its own and “wakes up” every two hours to delete expired items from the database. A software package entry that has been marked for deletion should therefore never remain in the ‘Entries’ table for more than 26 hours plus a few seconds.

Between the time that an entry has been marked for deletion and the time it’s actually deleted, it will still be shown by a ‘show’ command and *and can still be installed*. It can also be “undeleted”, if the user decides that it’s needed after all:

```

undelete entry hello 2.7
⇒
Unmarked entry ‘hello 2.7’ as deleted successfully.

```

The ‘delete ... finally’ commands delete one or more entries “irrevocably”¹:

```

delete entry hello 2.7 finally
⇒
Deleted entry ‘hello 2.7’.

```

Marking an entry for deletion that’s already been marked or undeleting one that hasn’t been marked for deletion has no effect. Nor does marking or undeleting a non-existent entry. However, in all of these cases the server sends a warning to the client:

```

mark as deleted entry bison 2.4.3
⇒
Marked entry ‘bison 2.4.3’ as deleted successfully.

mark as deleted entry bison 2.4.3
⇒
WARNING! Failed to mark entry ‘bison 2.4.3’ as deleted:
Entry already marked for deletion.
Continuing.
WARNING! In ‘main’: ‘warnings_occurred’ == 1
(No errors)
See error logfile ‘/tmp/optdbcli_error.log.JV95cR’ and/or \
    logfile ‘/tmp/optdbcli.log.pXk2pp’ for more information.
Exiting function with return value 4.

delete entry bison 2.4.3 finally
⇒
Deleted entry ‘bison 2.4.3’.

```

¹ Of course, a new entry containing the same data can be created again at any time.

```

delete entry bison 2.4.3 finally
⇒
WARNING! Failed to delete entry 'bison 2.4.3'. Continuing.
WARNING! In 'main': 'warnings_occurred' == 1
(No errors)
See error logfile '/tmp/optdbcli_error.log.GT0Rhp' and/or \
logfile '/tmp/optdbcli.log.uM6HOM' for more information.

```

When software package entries are deleted “finally”, or “culled” from the ‘Entries’ database table by `cull_database`, rows in the ‘Prerequisites’ table that refer to these packages are deleted, too. Please note that they are *not* deleted when a package has merely been marked for deletion. Otherwise, the information in the ‘Prerequisites’ table would be lost, if the package were undeleted before being deleted finally.

In the following example, ‘m4 1.4.15’ is a prerequisite of both ‘bison 2.4.3’ and ‘hello 2.7’. After the entry for ‘m4 1.4.15’ has been deleted, it is no longer a prerequisite of the other packages:

```

show entries user with_dependents with_prerequisites
⇒
*** User Entries ***

user_id:                1
commonName:             Laurence Finston
organizationalUnitName:  Gesellschaft fuer wissenschaftliche \
                        Datenverarbeitung mbH
countryName:            DE
entry_id:               1
package_name:           bison
package_version:        2.4.3
[...]
deleted:                0

Prerequisites:          m4 1.4.15

No dependents

user_id:                1
commonName:             Laurence Finston
organizationalUnitName:  Gesellschaft fuer wissenschaftliche \
                        Datenverarbeitung mbH
countryName:            DE
entry_id:               2
package_name:           hello
package_version:        2.7
[...]
deleted:                0

```

Prerequisites: m4 1.4.15

No dependents

```

user_id: 1
commonName: Laurence Finston
organizationalUnitName: Gesellschaft fuer wissenschaftliche \
                        Datenverarbeitung mbH
countryName: DE
entry_id: 3
package_name: m4
package_version: 1.4.15
[...]
deleted: 0

```

No prerequisites

```

Dependents: bison 2.4.3
            hello 2.7

```

delete entry m4 1.4.15 finally

⇒

Deleted entry 'm4 1.4.15'.

show entries user with_dependents with_prerequisites

⇒

*** User Entries ***

```

user_id: 1
commonName: Laurence Finston
organizationalUnitName: Gesellschaft fuer wissenschaftliche \
                        Datenverarbeitung mbH
countryName: DE
entry_id: 1
package_name: bison
package_version: 2.4.3
[...]

deleted: 0

```

No prerequisites

No dependents

```

user_id:                1
commonName:             Laurence Finston
organizationalUnitName:  Gesellschaft fuer wissenschaftliche \
                        Datenverarbeitung mbH
countryName:            DE
entry_id:               2
package_name:           hello
package_version:        2.7
[...]
deleted:                0

```

No prerequisites

No dependents

See also Section 21.2.3.3 [Functions for Deleting Entries], page 77.

4.4.3 Showing Software Package Information

The ‘show entries’ command is used for fetching information about software packages from the database and displaying them on the user’s computer terminal.

The complete syntax of the ‘show entries’ command is as follows:

```

⟨statement⟩ → SHOW ENTRY ⟨entry string list⟩ ⟨show option list⟩
              | SHOW ENTRIES ⟨show option list⟩

⟨show option list⟩ → (Empty) | ⟨show option list⟩ ⟨show option⟩

⟨show option⟩ → ALL | USER | PUBLIC | WITH_PREREQUISITES WITH_DEPENDENTS
              | WITH_INSTALLATION_SCRIPT | HTML | JAVASCRIPT | LEVELS ⟨level list⟩

⟨level list⟩ → (Empty) | ⟨level list⟩ INTEGER INTEGER
              | ⟨level list⟩ INTEGER INTEGER SEMI_COLON
              | ⟨level list⟩ INTEGER INTEGER SLASH

```

Example:

```

show entries user
⇒
*** User Entries ***

user_id:                1
commonName:             Laurence Finston
organizationalUnitName:  Gesellschaft fuer wissenschaftliche \
                        Datenverarbeitung mbH
countryName:            DE
entry_id:               1
package_name:           hello
package_version:        2.7
version_ctr:            NULL

```

```

download_url:                ftp://ftp.gwdg.de/pub/gnu/ftp/gnu/\
                             hello/hello-2.7.tar.gz
download_command:            NULL
maintainer_name:             NULL
maintainer_email_address:    NULL
private:                     0
checked_by_admin:            0
installation_script:         PRESENT
created:                     2012-05-14 13:00:45 UTC
last_modified:               2012-05-14 13:00:45 UTC
encrypted:                   0
owner:                       NULL
download_url_encrypted:      NULL
download_command_encrypted:  NULL
installation_script_encrypted: NULL

authorization:               NULL

deleted:                     0

```

See also Section 21.2.3.4 [Showing Entries], page 78.

4.5 Setting Effective User

`<statement> → SET EFFECTIVE_USER STRING`

‘STRING’ is the Distinguished Name of another user.

Example:

```

SET EFFECTIVE_USER "/C=DE/O=GridGermany/OU=Gesellschaft fuer \
wissenschaftliche Datenverarbeitung mbH/CN=Laurence Finston"
⇒
Set effective user succeeded.

```

This command can only be executed by a user who is permitted to act as a *delegate* for the user referred to by the Distinguished Name (‘STRING’) argument. That is, there must be a corresponding row in the ‘Delegates’ database table (see Section 30.5 [Delegates Database Table], page 99) or the user who executes the command must have the ‘superuser’ privilege. See Section 30.4 [Privileges Database Table], page 98. Otherwise, an error message is issued and ‘optdbcli’ exits:

```

SET EFFECTIVE_USER "/C=DE/O=GridGermany/OU=Gesellschaft fuer \
wissenschaftliche Datenverarbeitung mbH/CN=Laurence Finston"
⇒
ERROR: In ‘zzparse’:
Set effective user failed.
Exiting ‘zzparse’ unsuccessfully With return value 1.
ERROR! In ‘exchange_data_with_server’: ‘zzparse’ returned 1

```



```

Exiting function unsuccessfully with return value 1.
ERROR! (3) In 'client_connect_auth': 'exchange_data_with_server' \
    failed, returning 1:
Exiting function unsuccessfully with return value 1.
ERROR! In 'main': 'client_connect_auth' failed, returning 1
'errors_occurred' == 2
'warnings_occurred' == 0
See error logfile '/tmp/optdbcli_error.log.AL2MbF' and/or \
    logfile '/tmp/optdbcli.log.dkHaAP' for more information.
Exiting 'main' unsuccessfully with exit value 1.

```

4.6 Showing X.509 Certificates

The complete syntax of the 'show certificates' command is as follows:

```

<statement> → SHOW CERTIFICATE USER <html optional>
              | SHOW CERTIFICATES ALL <html optional>

```

```

<html optional> → (Empty) | HTML

```

Any user can show the data stored in the remote database for his or her own X.509 certificate, i.e., the row for that user in the 'Certificates' database table. See Section 30.3 [Certificates Database Table], page 97. There should be row for all users; if there isn't, 'optdbcli' should have exited.

Only users with at least one of the privileges 'superuser', 'show_certificates', 'add_globus_site_admin', or 'delete_globus_site_admin' may show certificate information for other users.

Users who can add or delete administrators of the data pertaining to Globus sites in the 'Globus_Sites' database table must be able to view the certificate information in order to find the user IDs for the `add admin globus_site` and `delete admin globus_site` commands. See Section 30.8 [Globus_Sites Database Table], page 100, and Section 30.13 [Globus_Sites_Admins Database Table], page 101.

Normally, the 'show_certificates' should be set for these users, but the function `Scan_Parse_Parameter_Type::edit_globus_site_admin` will check 'add_globus_site_admin' or 'delete_globus_site_admin', as appropriate, just in case it isn't. See Section 21.2.6 [Globus Site Functions], page 79.

If a user without at least one of these privileges sends the command 'show certificates all' to the server, his or her own certificate data will be shown and a warning will be issued that the user may not view the certificate data for other users.

4.7 Environment Shellscripts

4.7.1 Sending Environment Shellscripts

```

<statement> → CLIENT TO SEND ENVIRONMENT SHELLSCRIPT STRING GLOBUS_SITE
              STRING <admin optional>

```

$\langle \text{admin optional} \rangle \longrightarrow (\text{Empty}) \mid \text{ADMIN}$

Environment shellscripts are used during *remote installation*. See Section 27.1 [Remote Installation], page 87. They are executed in the shell in which the installation takes place. An environment shellscript will therefore typically contain assignments to variables such as ‘PATH’, ‘LD_LIBRARY_PATH’, ‘CFLAGS’, ‘LDFLAGS’ etc., in order to ensure that programs, libraries, header files, and other things needed for installation are found.

An environment shellscript is specific to a particular Globus site. An administrator of a Globus site entry in the database (who may or may not be the administrator of the Globus site itself) may store one or more environment shellscripts for that Globus site in the ‘Environment_Shellscripts’ database table. See Section 30.12 [Environment_Shellscripts Database Table], page 101. (There may be more than one such administrator for a given Globus site.) When a user (i.e., any user) starts a remote installation, passing that Globus site as an argument, this shellscript or these shellscripts will be fetched from the database and executed in the shell before the installation is started. They are fetched and executed in the order in which they appear in the database, i.e., according to the value stored in the ‘environment_shellscript_id’ field for the corresponding row in the table. In general, this means that they will be executed in the order in which they were stored in the database.

In addition, any user can store one or more environment shellscripts in the database. When *this user* (i.e., not other users) starts a remote installation, these shellscripts will be fetched from the database and executed in the shell before the installation is started, but after the ones supplied by the administrators. Like these, they are executed in the order in which they were stored in the database.

See also Section 21.2.8 [Environment Shellscript Functions], page 80, and Section 30.12 [Environment_Shellscripts], page 101.

4.7.2 Showing Environment Shellscripts

The complete syntax of the ‘show environment shellscripts’ command is as follows:

$\langle \text{statement} \rangle \longrightarrow \text{SHOW ENVIRONMENT SHELLSCRIPTS } \langle \text{show environment shellscript option list} \rangle$

$\langle \text{show environment shellscript option list} \rangle \longrightarrow (\text{Empty})$
 $\mid \langle \text{show environment shellscript option list} \rangle \langle \text{show environment shellscript option} \rangle$

$\langle \text{show environment shellscript option} \rangle \longrightarrow \text{USER} \mid \text{GLOBUS_SITE STRING} \mid \text{GLOBUS_SITES ALL}$

A user can show his or her own environment shellscripts and/or the “admin” shellscripts for a particular Globus sites or all Globus sites.

Example:

```
show environment shellscripts globus_sites all
⇒
*** Environment shellscripts ***

environment_shellscript_id: 1
user_id: 12
user_name: dgon0015
commonName: Laurence Finston
```

```

organizationalUnitName: Gesellschaft fuer wissenschaftliche \
    Datenverarbeitung mbH
globus_site_id: 1
hostname: optinum-srv.gwdg.de
admin: 1
disabled: 0
shellscript: #### Environment variables for optinum-srv.gwdg.de

export LIBS="$LIBS -lmysqlclient"
export LDFlags="$LDFlags -L/opt/optinum/dbsrvcli/lib"
export CFlags="$CFlags -I/opt/optinum/dbsrvcli/include"
export CXXFlags="$CFlags -I/opt/optinum/dbsrvcli/include"

```

See also Section 21.2.8 [Environment Shellscript Functions], page 80.

4.8 Showing Privileges

The complete syntax of the ‘show privileges’ command is as follows:

```

<statement> → SHOW PRIVILEGES USER <html optional> | SHOW PRIVILEGES ALL <html optional>

<html optional> → (Empty) | HTML

```

Any user can show his or her own privileges, i.e., the row (if any) for that user in the ‘Privileges’ database table. See Section 30.4 [Privileges Database Table], page 98. If a user has no special privileges, then there will probably be no data in the table for that user. However, it is possible that there would be a row with a value of 0 for each of the fields corresponding to a privilege.

Only users with the privilege ‘superuser’ or ‘show_privileges’ can show data for other users from the ‘Privileges’ table. If a user without at least one of these privileges sends the command ‘show privileges all’ to the server, his or her own privileges will be shown (if they exist) and a warning will be issued that the user may not view privileges information about other users:

```

show privileges all
⇒
*** Privileges info for user 1 (lfinsto) ***

user_id:                1
user_name:               lfinsto
commonName:             Laurence Finston
organizationalUnitName:  Gesellschaft fuer wissenschaftliche \
                        Datenverarbeitung mbH
superuser:              0
show_certificates:      0
show_privileges:        0
add_globus_site:        0
delete_globus_site:     0
enable_globus_site:     0

```

```

edit_globus_site:          0
add_globus_site_admin:     0
delete_globus_site_admin:  0
delegate:                  0

```

```

WARNING! In 'zzparse': Rule 'statement: SERVER_ZZ SHOW_ZZ \
                                PRIVILEGES_ZZ FAILED_ZZ INTEGER_ZZ':

```

```

User not permitted to view privileges for other users.
Continuing.

```

```

WARNING! In 'main': 'warnings_occurred' == 1
(No errors)

```

```

See error logfile '/tmp/optdbcli_error.log.I5pwAH' \
and/or logfile '/tmp/optdbcli.log.EoN9ru' for more information.
Exiting function with return value 4.

```

See also Section 21.2.7 [Functions for Privileges], page 80.

4.9 Retrieving Information and Installing Locally

Let us now assume that the user is working on the *front-end node* of a grid computer cluster and wishes to fetch information about downloading and installing the package 'GNU hello', version 2.7, which he or she has previously had stored in the database using the following commands:

```

start_entry
package_name <hello>
package_version <2.7>
download_url <ftp://ftp.gwdg.de/pub/gnu/ftp/gnu/hello/hello-2.7.tar.gz>
installation_script <inst_script_hello.sh>
end_entry

optdbcli --fetch --package-name hello --package-version="2.7" \
optinum-srv.gwdg.de
⇒
ftp://ftp.gwdg.de/pub/gnu/ftp/gnu/hello/hello-2.7.tar.gz # \
# hello # 2.7

```

When the option '--fetch' is used, the download command, download URL and/or installation script for the packages specified with the '--package-name' and (optionally) '--package-version' options are "fetched" from the database by the server and sent to the client. In this case, no further action is taken.

```

optdbcli --install --package-name hello --package-version="2.7" \
optinum-srv.gwdg.de
⇒
ftp://ftp.gwdg.de/pub/gnu/ftp/gnu/hello/hello-2.7.tar.gz # # \
hello # 2.7
Directory '/home/lfinsto/Installer/hello_2.7' doesn't exist.
iter->download_command ==

```

```

iter->download_url == \
ftp://ftp.gwdg.de/pub/gnu/ftp/gnu/hello/hello-2.7.tar.gz
...
'optdbcli' not creating symbolic links.
...
make install succeeded
In hello_2.7_install.sh: Installed hello 2.7 successfully in \
    /home/lfinsto/Installer/hello_2.7
bin subdirectory exists
no include subdirectory
no lib subdirectory

Execute the following commands in a shell to set its environment \
    (if desired):

export PATH=$PATH:/home/lfinsto/bin

In 'install_packages': Installed hello 2.7 successfully in \
    '/home/lfinsto/Installer/hello_2.7'.

```

This example differs from the last only in that the `--fetch` option has been replaced by `install`. In this case, the client will attempt to install the packages specified with the `--package-name` and (optionally) `--package-version` options. (`--install` implies `--fetch`.)

The directories `$HOME/Installer/` and `$HOME/installer_work/` are created, if they don't already exist. In addition, the subdirectory `$HOME/installer_work/hello_2.7/` is created, overwriting any existing subdirectory of this name.

If the entry for `hello 2.7` in the database contains an installation script, it is copied into `$HOME/installer_work/hello_2.7/`. Otherwise, a default shellsript is created in this directory.

If the entry contains a "download command", the latter is executed with `$HOME/installer_work/hello_2.7/` as the working directory. Otherwise, if it contains a "download URL", `wget` is used to fetch its contents, which in most cases will be a compressed `tar` file containing the distribution of the software package (here, `hello 2.7`).

Next, the installation script is executed, either the one stored in the database or the default script, if there was none. Normally, the script will unpack the compressed `tar` file, enter the top-level directory of the distribution and perform any actions needed to build the package. On Unix-like systems, this will typically consist of calls to `configure`, `make` and `make install`. However, any given may require special handling.

Multiple packages may be specified using `--package-name`. However, if `--package-version` is used, it must be used for all packages. In addition, the order in which the package versions are specified in the command must correspond to the order in which the packages are specified. For example, in this call:

```

optdbcli --package-name hello --package-name m4 \
    --package-version "2.7" --package-version="1.4.15" ...

```

the version of ‘hello’ to be fetched or installed is 2.7 while the version of ‘m4’ is 1.4.15.

Since some packages may not have version numbers or the user might not know or care about them, the argument to ‘`--package-version`’ is optional. However, if multiple packages are specified and at least one of them has a “non-empty” ‘`--package-version`’, instances of ‘`--package-version`’ without an argument must be specified as placeholders for each of the other packages, e.g.,

```
optdbcli --package-name hello --package-name m4 \
  --package-name bison --package-version --package-version="1.4.15" \
  --package-version ...
```

Here, only ‘m4’ has a non-empty package version, but empty instances of ‘`--package-version`’ must be used for ‘hello’ and ‘bison’.

The previous examples demonstrate the rules, but it’s better practice to keep corresponding instances of ‘`--package-name`’ and ‘`--package-version`’ together, e.g.,

```
optdbcli --package-name hello --package-version \
  --package-name m4 --package-version="1.4.15" \
  --package-name bison --package-version
...
```

See Chapter 2 [Invoking `optdbcli`/`optdbsrv`], page 6, for more details about arguments and options used when invoking the client program ‘`optdbcli`’.

4.10 Remote Installation

```
<statement> → install_or_reinstall PACKAGE_NAME STRING
               package_version_optional GLOBUS_SITE STRING
               stdout_filename_optional stderr_filename_optional timeout_optional
               reinstall_prerequisites_optional link_expression_list
```

```
install_or_reinstall → INSTALL | REINSTALL
```

```
<package version optional> (Empty) | PACKAGE_VERSION STRING
```

```
<stdout filename optional> (Empty) | STDOUT_FILENAME STRING
```

```
<stderr filename optional> (Empty) | STDERR_FILENAME STRING
```

```
<timeout optional> (Empty) | TIMEOUT INTEGER
```

```
<reinstall prerequisites optional> → (Empty) | REINSTALL PREREQUISITES
                                     | NO REINSTALL PREREQUISITES
```

```
<link expression list> → (Empty) | link_expression_list link_expression
```

`<link expression>` \longrightarrow `NO_LINKS` | `WITH_LINKS` | `LINK_DIRECTORY STRING`

See also Section 27.1 [Remote Installation], page 87, and Section 4.11 [Transferring Proxy Credential (`optdbcli`)], page 36.

4.11 Transferring Proxy Credential

Remote installation (see Section 27.1 [Remote Installation], page 87, and Section 4.10 [Remote Installation (`optdbcli`)], page 35, above) requires that a *proxy credential* be available on the computer where ‘`optdbsrv`’ is running. The following command is used to transfer the proxy credential:

`<statement>` \longrightarrow `CLIENT TO TRANSFER PROXY_CREDENTIAL STRING`

Example:

```
CLIENT TO TRANSFER PROXY_CREDENTIAL <x509up_u1000>
```

The proxy credential (‘`x509up_u1000`’ in this example) must already exist on the computer where the client is running. The shellscript ‘`gen_proxy.sh`’, which is included in the distribution, can be used to generate a proxy credential. See [Client/Server (`optdbcli`/`optdbsrv`) Shellscripts], page 106. It, in turn, uses the program ‘`grid-proxy-init`’, which is part of the Globus Toolkit distribution and should be located in ‘`$GLOBUS_LOCATION/bin/`’, e.g., ‘`/usr/local/globus-4.0.8/bin/grid-proxy-init`’.

‘`optdbsrv`’ processes this command immediately, so that a proxy credential can be transferred and one or more remote installation commands can be executed with a single invocation of the client. Please note that a proxy credential only needs to be transferred if there is no valid proxy available on the computer where ‘`optdbsrv`’ is running; it is not necessary to transfer proxy credentials for each remote installation or each invocation of ‘`optdbcli`’.

The proxy credential is stored in the ‘`/tmp/`’ directory on the computer where the server is running. The owner is the user on that system corresponding to the entry in the ‘`grid-mapfile`’ for the Distinguished Name of the certificate or proxy credential passed as an argument to ‘`optdbcli`’ or the Distinguished Name in a ‘`SET EFFECTIVE_USER`’ command. See Section 4.5 [Setting Effective User (`optdbcli`)], page 29. Please note that the latter case is only possible for users who are authorized to act as a *delegate* for one or more other users. See Section 4.5 [Setting Effective User (`optdbcli`)], page 29, and Section 30.5 [Delegates Database Table], page 99.

4.12 Globus Sites

The commands for adding, deleting, etc., Globus site information are only available to users with the appropriate privileges (‘`add_globus_site`’, ‘`delete_globus_site`’, etc.). See Section 30.4 [Privileges], page 98.

4.12.1 Adding Globus Sites

`<statement>` \longrightarrow `ADD GLOBUS_SITE STRING` `<globus site option list>`

`<globus site option list>` \longrightarrow (`Empty`) | `<globus site option list>` `<globus site option>`

$\langle \text{globus site option} \rangle \longrightarrow$ INTERACTIVE_NODE STRING | APPEND_LD_LIBRARY_PATH STRING
 | APPEND_PATH STRING | OPTDBCLI_PATH STRING | IP_ADDRESS STRING
 | INTERACTIVE_NODE_IP_ADDRESS STRING | ENABLED | DISABLED | REPLACE

4.12.2 Enabling and Disabling Globus Sites

$\langle \text{statement} \rangle \longrightarrow$ ENABLE GLOBUS_SITE STRING | DISABLE GLOBUS_SITE STRING

4.12.3 Deleting and Undeleting Globus Sites

$\langle \text{statement} \rangle \longrightarrow$ MARK AS DELETED GLOBUS_SITE STRING
 | UNDELETE GLOBUS_SITE STRING enable_optional
 | DELETE GLOBUS_SITE STRING FINALLY

enable_optional \longrightarrow (Empty) | ENABLE | DISABLE

4.13 Globus Site Administrators (optdbcli)

$\langle \text{statement} \rangle \longrightarrow$ add_or_delete ADMIN GLOBUS_SITE STRING USER INTEGER
 | EDIT GLOBUS_SITE STRING SET globus_site_option_list
 | EDIT GLOBUS_SITE STRING UNSET globus_site_option_list_1

$\langle \text{add or delete} \rangle \longrightarrow$ ADD | DELETE

$\langle \text{globus site option list 1} \rangle \longrightarrow$ (Empty) | globus_site_option_list_1 globus_site_option_1

$\langle \text{globus site option 1} \rangle \longrightarrow$ INTERACTIVE_NODE | APPEND_LD_LIBRARY_PATH
 | APPEND_PATH | OPTDBCLI_PATH | IP_ADDRESS
 | INTERACTIVE_NODE_IP_ADDRESS

5 Global Variables and Constants optdbcli/optdbsrv

5.1 Global Constants optdbcli/optdbsrv

`string DEFAULT_PORT_STR_AUTH = "5556"` [Global Constant]
The port to which users will be connecting per default with authentication as a string.

`int DEFAULT_PORT_NUM_AUTH = 5556` [Global Constant]
The port to which users will be connecting per default with authentication as an integer.

`string DEFAULT_PORT_STR_NON_AUTH = "5557"` [Global Constant]
The port to which users will be connecting per default without authentication as a string.

`int DEFAULT_PORT_NUM_NON_AUTH = 5557` [Global Constant]
The port to which users will be connecting per default without authentication as an integer.

`int MAX_PROTOCOL = 4` [Global Constants]
`int NULL_PROTOCOL = 0`
`int FTP_PROTOCOL = 1`
`int HTTP_PROTOCOL = 2`
`int HTTPS_PROTOCOL = 3`
 MAX_PROTOCOL must be changed whenever protocols are added.

`string DEFAULT_CA_FILENAME`
The default value is conditionally compiled, depending on the platform. It is only non-empty for a selection of machines known to the author. If non-empty, the filename is always 'DFN-VereinPCAGrid-G01.pem', but the path differs.

`string DEFAULT_CRL_FILENAME = ""`

`int MAX_FILENAME = 128` [Global Constant]

`int MAX_USER_ID = 128` [Global Constant]

`unsigned int SHOW_USER_ENTRIES = 1` [Global Constant]

`unsigned int SHOW_PUBLIC_ENTRIES = 2` [Global Constant]

`string DEFAULT_LOG_FILENAME = "/tmp/optdbcli.log"` [Global Constant]

`string DEFAULT_ERR_LOG_FILENAME =` [Global Constant]
`"/tmp/optdbcli_error.log"`

5.2 Global Variables optdbcli/optdbsrv

<code>unsigned int thread_ctr</code> Initialized to 0.	[Global Variable]
<code>pthread_mutex_t thread_ctr_mutex</code> <code>pthread_mutex_t cerr_mutex</code> <code>pthread_mutex_t global_variable_mutex</code> <code>pthread_mutex_t session_data_mutex</code> <code>pthread_mutex_t log_strm_mutex</code> <code>pthread_mutex_t err_log_strm_mutex</code> Mutexes.	[Global Variables]
<code>ofstream output_file_strm</code> <code>ofstream log_strm</code> <code>ofstream err_log_strm</code> Output file streams.	[Global Variables]
<code>string server_ip_address</code> The IP address of the server. Used by the client.	[Global Variables]
<code>string port_str_auth</code> <code>string port_str_non_auth</code> <code>int port_num_auth</code> <code>int port_num_non_auth</code> Ports. Initialized to "" or 0, as appropriate.	[Global Variables]
<code>string input_filename</code> <code>string output_filename</code> Filenames.	[Global Variables]
<code>bool save_temp_files = false</code>	[Global Variable]
<code>string owner_certificate = ""</code>	[Global Variable]
<code>int sleep_value = 0</code> For debugging.	[Global Variable]
<code>map<string, string> dn_fields</code>	[Global Variable]
<code>map<string, string> dn_username_map</code>	[Global Variable]
<code>bool is_client</code> <code>bool is_server</code>	[Global Variables]
<code>string distinguished_name_str</code> <code>Distinguished_Name_Type distinguished_name</code> <code>vector<Distinguished_Name_Type> distinguished_name_vector</code>	[Global Variables]
<code>string grid_mapfile_filename</code> Initialized to a value depending on conditionally compiled code.	[Global Variable]

<code>vector<string> package_names</code>	[Global Variables]
<code>vector<string> package_versions</code>	
<code>bool resolve_dependencies = true</code>	[Global Variable]
<code>bool install = false</code>	[Global Variable]
<code>bool fetch = false</code>	[Global Variable]
<code>bool authenticated_connection = false</code>	[Global Variable]
<code>int errors_occurred = 0</code>	[Global Variable]
<code>int warnings_occurred = 0</code>	[Global Variable]
<code>string DEFAULT_CERT_FILENAME</code>	[Global Variables]
<code>string DEFAULT_KEY_FILENAME</code>	
<code>string log_filename</code>	[Global Variables]
<code>string err_log_filename</code>	
<code>bool download_fail_continue</code>	[Global Variables]
<code>bool install_fail_continue</code>	
<code>string link_directory</code>	[Global variable]
<code>int create_links_value = 0</code>	[Global variable]
<code>string globus_site</code>	[Global variable]
Set by means of the '--globus-site' option.	
<code>map<string, bool> globus_site_map</code>	[Global variable]
<code>string optimum_installer_gpg_pubkey_fingerprint</code>	[Global variable]
<code>string gpg_homedir</code>	[Global variable]
<code>int trace_value = 0</code>	[Global variable]
<code>bool scanner_trace = false</code>	[Global variable]
<code>bool reinstall_flag = false</code>	[Global variable]
<code>string install_directory</code>	[Global variable]
<code>int prerequisites_reinstall_value = 0</code>	[Global variable]
<code>vector<string> temp_file_vector</code>	[Global variable]
<code>vector<string> fifo_vector</code>	[Global variable]
<code>string proxy_cred_filename</code>	[Global variable]
<code>bool transfer_proxy_cred = false</code>	[Global variable]
<code>string transfer_proxy_cred_filename</code>	[Global variable]
<code>bool proxy_cred_verified = false</code>	[Global variable]
<code>vector<string> cert_filenames</code>	[Global variables]
<code>vector<string> key_filenames</code>	
<code>vector<string> ca_filenames</code>	[Global variables]
For server only, unless the client needs to verify a proxy credential. In this case, only the first one is used.	

`vector<string> crl_filenames`

For server only.

[Global variables]

`string session_id`

For client only.

[Global variable]

6 Global Functions optdbcli/optdbsrv

These functions are defined in ‘`glblfncls.web`’. They don’t require many declarations of data types, functions and/or variables declared in the OptiNum Installer package, so the corresponding header file can be included in files where other header files from the package are not included, or before these header files are included.

```
int set_cert_files (string &curr_key_filename, string [Function]
                  &curr_cert_filename, string &curr_ca_filename, string
                  &curr_crl_filename)
```

```
int init_maps (void) [Function]
```

```
void lock_cerr_mutex (void) [Function]
```

```
void unlock_cerr_mutex (void) [Function]
```

```
string get_datestamp (void) [Function]
```

7 Server Connection Function Reference

int main (*int argc, char** argv*) [Function]

Main function for server. Defined in ‘*optdbsrv.web*’.

Sets values of global variables. Some of these are set to other values in ‘*optdbcli*’, so they can’t be initialized where they are declared in ‘*glblvrbl.web*’.

Processes command-line options.

Performs initializations, i.e., of the mutexes that are used. In addition, some of the libraries with which ‘*optdbsrv*’ is linked require initialization, i.e., the MySQL, libgcrypt and GNUTLS libraries.

It then starts a thread with the function ‘*listen_auth*’ and another with the function ‘*listen_non_auth*’. These functions continue to execute until the server is stopped. In normal use, the server is meant to run in the background “forever”. In practice, it will be stopped when developing or debugging the program, whenever the machine on which it is running is rebooted, or for any number of other reasons.

While it doesn’t do any harm to just end the program by sending it a signal to “kill” it, which is what one normally does, the ‘*main*’ function contains code to “join with” the two threads, after which it destroys the mutexes and calls ‘*mysql_library_end*’ and ‘*gnutls_global_deinit*’. (In addition, it closes an output file, which as of 2010.07.23 is, however, not open.) This ensures that program ends cleanly if the threads exit for some reason, whereby both threads must exit; if only one exits, ‘*main*’ will block forever, waiting for the other one to exit.

The following functions are defined in ‘*connect.web*’.

void* connect_func (*void* v*) [Function]

For “authorized” connections using X.509 certification.

void* connect_func_non_auth (*void* v*) [Function]

7.1 Exchange Data With Client

int exchange_data_with_client (*Scan_Parse_Parameter_Type * param*) [Function]

exchange_data_with_client is one of the most important functions in the server program ‘*optdbsrv*’. It controls the communication with the client program ‘*optdbcli*’. See also Section 8.1 [Exchange Data With Server], page 44.

8 Client Connection Function Reference

`int main (int argc, char** argv)` [Function]
Defined in ‘`optdbcli.web`’.

`void finish (void)` [Function]
Exit handler. Passed as argument to ‘`atexit`’.

The following functions are defined in ‘`cnnectcli.web`’.

These functions are defined in ‘`cnnectcli.web`’.

`int client_connect_auth (void)` [Function]
Connects with the server using authorization/authentication by means of X.509 certificates.

`int client_connect_non_auth (void)` [Function]
Connects with the server using authorization/authentication by means of a *proxy credential* passed to the client as the argument to the command line option `--proxy-certificate`.

8.1 Exchange Data With Server

`int exchange_data_with_server (Scan_Parse_Parameter_Type & param)` [Function]
`exchange_data_with_server` is one of the most important functions in the client program ‘`optdbcli`’. It controls the communication with the server program ‘`optdbsrv`’. See also Section 7.1 [Exchange Data With Client], page 43.

9 Web Application

The web application ‘`optwbsrv.fcgi`’ makes it possible for users without access to the client application ‘`optdbcli`’ to create entries in the package database. It may not be possible or desirable to install ‘`optdbcli`’ on a particular computer for some reason or other. In particular, it may not be possible to install it on a system running under one of the Windows operating systems.

The web application ‘`optwbsrv.fcgi`’ is a *FastCGI* program that is executed under the control of a *webserver*, such as Apache `httpd`. (CGI stands for “Common Gateway Interface”.) Normally, users will not want to install it themselves, but will use ‘`curl`’ or a similar package to send commands to an instance of ‘`optwbsrv.fcgi`’ installed at a remote location.

However, it is, of course, possible to install ‘`optwbsrv.fcgi`’ locally. It must be installed in (i.e., copied to) a directory, where the webserver looks for such programs or “scripts”. A typical location would be ‘`/srv/www/fcgi-bin/`’. The owner of ‘`optwbsrv.fcgi`’ must either be root or a user with appropriate permissions. For example, a user created especially for the purpose of executing CGI and/or FastCGI scripts.

The C++ source code for the web application ‘`optwbsrv`’ is contained in a single file, namely ‘`[...]/optwbsrv/optwbsrv.web`’.

9.1 Accessing the Web Application

The web application can be accessed as follows:

```
curl --data-urlencode input@<data file> \
    --data-urlencode \
    installation_scripts@<installation script data file> \
    --cacert ca-bundle.crt \
    --cert usercert.pem --key userkey.pem \
    https://some-machine.somewhere.de/fcgi-bin/optwbsrv.fcgi
```

‘`curl`’ is a command-line program that sends data to a webserver. Here, the latter is installed on the computer ‘`some-machine.somewhere.de`’. The webserver resolves the path ‘`/fcgi-bin/`’ appended to the name of the server to some directory on its system. This is determined by the configuration of the webserver. The server calls the web application ‘`optwbsrv.fcgi`’ to process the *request*, passing to it any data it contains.

The arguments:

‘`input@<data file>`’ ‘`curl`’ sends the contents of the ‘`<data file>`’ to the server.

‘`installation_scripts@<installation script data file>`’ ‘`curl`’ sends the contents of the ‘`<installation script data file>`’ to the server.

‘`--data-urlencode`’ Ensures that special characters in ‘`test_data.txt`’ are *URL encoded*.

‘`--cacert ca-bundle.crt`’ A *CA certificate bundle*, i.e., a file containing one or more CA (Certification Authority) certificates

‘`--cert usercert.pem`’

‘`--key userkey.pem`’

‘`https://some-machine.somewhere.de/fcgi-bin/optwbsrv.fcgi`’

9.1.1 Data Files

When creating entries using ‘`optwbsrv`’ via ‘`curl`’ or in some other way, the command ‘`INSTALLATION_SCRIPT <filename>`’ cannot be used. The reason for this is that the contents of ‘`<data file>`’ are passed by ‘`optdbcli`’ (the client) directly to ‘`optdbsrv`’ (the server); they are processed neither by ‘`optdbcli`’ nor by ‘`optwbsrv`’. The filename must refer to a file on the computer where ‘`optdbcli`’ is running, which in this case is the one where ‘`optwbsrv`’ is running, since the latter calls ‘`optdbcli`’.

In order to solve this problem, commands must be sent separately to the web application which it *will* process, the contents of the files containing the installation scripts must be transferred to the computer where ‘`optwbsrv`’ is running, and they must be associated with the correct entries.

The file ‘`<installation script data file>`’ contains commands and the contents of the installation scripts in a special format, e.g.,

```
START_INSTALLATION_SCRIPT_OPTW
PACKAGE_NAME_OPTW <m4>
START_TEXT_OPTW
echo "m4 installation script"
...
exit 0;
END_TEXT_OPTW
END_INSTALLATION_SCRIPT_OPTW

START_INSTALLATION_SCRIPT_OPTW
PACKAGE_NAME_OPTW <bison>
START_TEXT_OPTW
echo "bison installation script"
...
exit 0;
END_TEXT_OPTW
END_INSTALLATION_SCRIPT_OPTW
```

It is not possible to determine beforehand what a shellscript might contain, so the syntax used for these files is deliberately verbose in order to reduce the chances that something might appear in a shellscript that ‘`optwbsrv`’ would interpret as a command. For example, the command names all use the suffix ‘`_OPTW`’, since it is unlikely that that a variable name used in a shellscript would also have this suffix. However, this is not, of course, complete protection.¹

When ‘`optwbsrv`’ processes this file, it stores the shellscript code between each instance ‘`START_TEXT_OPTW`’ and ‘`END_TEXT_OPTW`’ in its own temporary file and appends a corresponding ‘`ADD INSTALLATION_SCRIPT`’ command with this filename to the code passed to ‘`optdbcli`’ and from it to ‘`optdbsrv`’. That is, if new entries are to be created, the commands ‘`START_ENTRY ... END_ENTRY`’ will be in ‘`<data file>`’

¹ A more reliable method would be to count the bytes in the shellscript file and send exactly this many bytes to ‘`optdbsrv`’. However, the author does not believe that this extra degree of safety is needed at this time.

and `'START_INSTALLATION_SCRIPT_OPTW ... END_INSTALLATION_SCRIPT_OPTW'` in `'<installation script data file>'`.

9.2 Auxiliary_File_Type

`'struct Auxiliary_File_Type'` is declared in `'instsctp.web'`.

9.3 Global Variables and Constants

`extern char ** environ` [Global Variable]

`const int IN_BUFFER_SIZE = 1048576` [Global Constant]

9.4 Web Application Functions

`int main (int argc, char** argv)` [Function]

The main function for the web application.

`int init (void)` [Function]

Initialization function. Currently this function simply returns 0.

`int parse_args (string& buffer, vector<pair<string, string> >& parameters)` [Function]

9.5 Scanner

The Flex input code for generating the scanner is in `'scnwbsrv.web'`.

Apart from the `'INITIAL'` start condition, which is automatically defined, the scanner code defines the exclusive start condition `'SAVING'`, which is used when reading the text of the installation script.

Since shellscripts may contain many different characters and sequences of characters, and it would not be possible for the programs in the Optinum-Grid package (or their authors) to enforce restrictions beyond the ones set by the shells themselves, the scanner must be able to handle anything that might reasonably appear in a shellscript. In particular, it should not misinterpret text in a shellscript for commands meant for the scanner itself.

It is, of course, not possible to protect against this entirely: If a shellscript contains the string `'END_TEXT_OPTW'`, this will cause the scanner to “think” that it’s finished reading the text of the shellscript and leave the `'SAVING'` start condition. It will then expect commands, which will probably not be forthcoming, which will, in turn, cause an error to occur.

The name `'END_TEXT_OPTW'` and the names of the other commands were deliberately chosen so that it would be unlikely that anyone would use them in a shellscript, unless it was with the intention of causing an error.

9.5.1 Variables

`bool DEBUG_SCANNER` [Variable]

9.5.2 Functions

`int yylex (YYSTYPE* lvalp, yyscan_t parameter)` [Function]

9.6 Parser

The GNU Bison input code for generating the parser is in ‘`prswbsrv.web`’.

9.6.1 Functions

`int yyparse` (*yyscan_t parameter*) [Function]

`int yywrap` (*void*) [Function]

`int yyerror` (*void* v, char const* s*) [Function]

10 Software Installation

`int install_packages (Scan_Parse_Parameter_Type ¶m)` [Function]
 Defined in `'instllfn.web'`.

This function is called when `'optdbcli'` is called using the `'--install'` option and at least one package name is specified using the `'--package-name'` option. Versions may also be specified using the `'--package-version'` option, whereby the number of instances of `'--package-version'` must either be zero or it must match the number of instances of `'--package-name'`. Otherwise, `'optdbcli'` would not be able to associate the versions to the package names. `'--package-version'` may be used without an argument as a placeholder for packages where the version is unknown or the user doesn't wish to specify one for some reason. See See Chapter 2 [Invoking `optdbcli/optdbsrv`], page 6, for more information.

Before this function is called, the server program `'optdbsrv'` will have retrieved information from the database for each of the packages specified. *Unless* `'optdbcli'` has been invoked using the `'--no-resolve-dependencies'` option, `'optdbsrv'` will also have retrieved information from the database for all of the packages upon which the packages specified with `'--package-name'` depend, i.e., their *prerequisites*. `'optdbcli'` may also be invoked using the option `'--resolve-dependencies'`, but resolving dependencies is the default behavior. (`'--resolve-dependencies'` may be used to cancel the effect of `'--no-resolve-dependencies'`, if the latter appears first on the command-line in an invocation of `'optdbcli'`.) See See Chapter 2 [Invoking `optdbcli/optdbsrv`], page 6.

If information for multiple packages is retrieved, and some packages are prerequisites of others, then the information will be sorted in order to ensure that packages are 1.~installed in the correct order and 2.~only installed once.

The information retrieved from the server consists of:

1. A number used for sorting, as described above.
2. A download URL, if present.
3. A download command, if present.
4. An installation script. If none is specified, a default script is created. (This occurs in the parser rule `'statement: NO_ZZ INSTALLATION_SCRIPT_ZZ FOR_ZZ ENTRY_ZZ entry_string_list ORDER_ZZ INTEGER_ZZ'`, which is defined in the file `'prsrcInt.web'`, see Chapter 17 [Parser Client], page 62.)
5. The package name.
6. The package version, if present. Please note that a package version may have been retrieved from the database even if `'optdbcli'` was invoked without an instance of the `'--package-version'` option for this package!

The directories `'$HOME/bin'`, `'$HOME/lib'`, `'$HOME/include'`, `'$HOME/installer_work'` and `'$HOME/Installer'` are created, if they don't already exist. In addition, a subdirectory for each package is created in `'$HOME/Installer/'`, where the names of these subdirectories consist either of the package name alone (if no package version was specified using the `'--package-version'` option) (e.g.,

“`“$HOME/Installer/ngspice”`”) or of the package name and the package version, separated by an underscore character (e.g., “`“$HOME/Installer/m4_1.4.14”`”).

A *download command*, if present, takes precedence over *download URL*. A download command is simply one or more shell commands, which are executed using ‘system’. If there is no download command, but a download URL is present, ‘wget’ is used to download the contents of the URL, which should contain the distribution of the package in question.

The package is downloaded into the directory ‘`$HOME/installer_work/`’. Then, the *installation script*, either one supplied by the user or the default installation script, is executed using ‘system’. The package is installed in the corresponding subdirectory of ‘`$HOME/Installer/`’, whose name is derived as described above.

Following installation, if the directories ‘`$HOME/Installer/<package>/bin/`’, ‘`$HOME/Installer/<package>/lib/`’, and/or ‘`$HOME/Installer/<package>/include/`’ exist, *symbolic links* are created in ‘`$HOME/bin/`’, ‘`$HOME/lib/`’, and/or ‘`$HOME/include/`’, that point to any files in these directories.

int create_links (string source_dir) [Function]

This function is currently only called for packages that are installed based on encrypted database entries (“secure installation”). If the installed package contains directories named ‘bin’, ‘include’ and/or ‘lib’, links are created in ‘`\$HOME/bin`’, ‘`\$HOME/include`’ and/or ‘`\$HOME/lib`’, respectively. Please note: These directories in the installed package can be at any level! For example, links to files in a directory ‘hello-2.7/gnulib/lib’ will also be made in ‘`\$HOME/lib`’; not ‘`\$HOME/gnulib/lib`’.

It would be possible to create the directory ‘`\$HOME/gnulib/lib`’, but within `optdbcli`, it is not known what directories will be created when the package is installed. That is, the directory into which `scrinstl` copies files is known. This is the `|source_dir|` argument to `|create_links|`. However, `optdbcli` has now further information about the directory structure below this directory, which will differ from package to package. \par This function therefore accounts for the most common cases in a simple way. If special handling is required, the installation script must account for it. However, the latter runs under the user id of the owner of the package (“company”), so it will probably be necessary to have it pass information back to `scrinstl` and possibly have `scrinstl` pass it back to `optdbcli`, in order to have one of these programs create the links under the user id of the user (“customer”).

10.1 Normal Software Installation

10.2 Hidden Software Installation

For many software packages, the source code is freely available and there is no need for secrecy regarding the source code itself, the location (URL) where the package is available for downloading or the installation procedure. In such cases, a user can store the URL and optionally an installation script in a database entry, as described above. An entry can be marked as ‘private’, so that other users of the OptiNum-Grid Installer system will not be able to view the database entry, but all of the information is, of course, available to the user who created it.

However, there are situations where this model doesn't apply, particularly where the software package itself or its use is sold commercially. For such cases, the OptiNum-Grid Installer provides a means of performing a *hidden software installation*.

The software database entry for a package that uses this facility differs from an ordinary database entry in that certain fields contain encrypted information that can only be decrypted by the owner of the package (*software provider*). The software provider provides its customers with encrypted files and the latter create database entries using them. When the package should be installed on a grid resource, the encrypted files are decrypted, the software is downloaded and the installation procedure is carried out *all under the user account of the software provider*. Upon successful installation, whatever the software provider intends to make available to the customer, e.g., executable, library, header and/or other files, are copied to the latter's account. In this way, it is ensured that the customer receive only as much material or information as the software provider wishes to provide.

```
int hidden_install (Scan_Parse_Parameter_Type &param, [Function]
                  vector<string> &arg_vector)
```

10.3 Local Software Installation

The OptiNum-Grid Installer is intended for installation software packages on grid resources. Therefore, from the user's point of view, the normal case is the "Remote Software Installation" (see below). However, it is also possible to use it for installing packages directly on the user's computer. This is especially useful for testing, but may be for other purposes, as well.

To install a package locally, invoke `optdbcli` using the options '`--install`' or '`--reinstall`' together with the option '`--package-name`' and optionally '`--package-version`', as described in Section 2.1.5 [Software Installation Options], page 8.

```
optdbcli --install --package-name="hello" --package-version="2.7"
```

10.4 Remote Software Installation

A *remote software installation* is performed on a grid resource. It is initiated by means of the command

```
[install | reinstall] package_name STRING [package_version STRING]
globus_site STRING [options]
```

in the input sent by `optdbcli` to `optdbsrv`.

For example,

```
reinstall package_name <hello> <2.7> globus_site <optinum-srv.gwdg.de>
```

causes 'hello 2.7' to be installed on 'optinum-srv.gwdg.de'. If it has already been installed by the users, it will be reinstalled.

Please note that no distinction is made between a normal and a hidden installation at this point. When `optdbsrv` retrieves the data for the software package from the database, it is determined whether the entry contains encrypted data or not. If it does, a hidden installation is performed. Otherwise, a normal installation is performed.

11 TLS Connections

TLS Connections: Communication using TLS.

11.1 Preprocessor Macros for TLS

```
TLS_SESSION_CACHE = 50                                [Preprocessor Macros]
MAX_SESSION_ID_SIZE = 32
MAX_SESSION_DATA_SIZE = 512
DH_BITS = 1024
SA = struct sockaddr
SOCKET_ERR (err,s) if(err==-1) perror(s);pthread_exit(0);
```

11.2 Types for TLS

CACHE is struct declared by means of a typedef in ‘gntlsfnc.web’.

```
char session_id[MAX_SESSION_ID_SIZE]                    [Data members]
char session_id[MAX_SESSION_ID_SIZE]
int session_id_size
char session_data[MAX_SESSION_DATA_SIZE]
int session_data_size
    Declared in ‘gntlsfnc.web’.
```

11.3 Variables for TLS

```
gnutls_x509_crl_t * crl_list                             [Variable]
    Global within ‘ex_verify.web’. All the available CRLs (Certificate Revocation
    Lists).
```

```
int crl_list_size                                         [Variable]
    Global within ‘ex_verify.web’.
```

```
static char srp_dh_group2048[]                           [Variable]
    Global within ‘gntlsfnc.web’.
```

```
static CACHE * cache_db                                  [Variable]
    Global within ‘gntlsfnc.web’.
```

```
static int cache_db_ptr                                   [Variable]
    Global within ‘gntlsfnc.web’.
```

```
const string mysql_password_encrypted                    [File-local constant]
    If used, mysql_password_encrypted should be a password for the MySQL database
    engine, encrypted using GPG and ASCII-encoded. It is used in listen_auth (see Sec-
    tion 11.4 [TLS Functions], page 53, below) for connecting with the MySQL database
    on platforms where a password is required. This should be the case for “production”
    installations of ‘optdbsrv’, but may not be for local installations used for testing.
```

11.4 TLS Functions

`int verify_certificate (gnutls_session_t session, const char *hostname = 0)` [Function]
 Defined in 'ex_rfc2818.web'.

`int verify_certificate_chain (gnutls_session_t session, const char *hostname, const gnutls_datum_t *cert_chain, int cert_chain_length, gnutls_x509_crt_t *ca_list, int ca_list_size, X509_Cert_Type* user_cert = 0, X509_Cert_Type* issuer_cert = 0, Scan_Parse_Parameter_Type* param = 0)` [Function]
 Defined in 'ex_verify.web'.

`void verify_cert2 (gnutls_x509_crt_t crt, gnutls_x509_crt_t issuer, gnutls_x509_crl_t *crl_list, int crl_list_size, Scan_Parse_Parameter_Type *param = 0)` [Static Function]
 Defined in 'ex_verify.web'. Verifies a certificate against an other certificate which is supposed to be it's issuer. Also checks the *crl_list*, if any, for whether the certificate has been revoked.

`void verify_last_cert (gnutls_x509_crt_t crt, gnutls_x509_crt_t *ca_list, int ca_list_size, gnutls_x509_crl_t *crl_list, int crl_list_size, Scan_Parse_Parameter_Type* param = 0)` [Function]
 Defined in 'ex_verify.web'.

`gnutls_session_t initialize_tls_session (gnutls_certificate_credentials_t& cert_cred)` [Function]
 Defined in 'gntlsfnc.web'.

`int generate_dh_params (gnutls_dh_params_t& dh_params);` [Function]
 Defined in 'gntlsfnc.web'.

`int generate_dh_params_non_auth (gnutls_dh_params_t& dh_params)` [Function]
 Defined in 'gntlsfnc.web'.

`gnutls_session_t initialize_tls_session_non_auth (gnutls_anon_server_credentials_t& anoncred)` [Function]
 Defined in 'gntlsfnc.web'.

<code>int generate_rsa_params (gnutls_rsa_params_t& rsa_params)</code> Defined in 'gntlsfnc.web'.	[Function]
<code>void wrap_db_init (void)</code> Defined in 'gntlsfnc.web'.	[Function]
<code>void wrap_db_deinit (void)</code> Defined in 'gntlsfnc.web'.	[Function]
<code>int wrap_db_store (void *dbf, gnutls_datum_t key, gnutls_datum_t data)</code> Defined in 'gntlsfnc.web'.	[Function]
<code>gnutls_datum_t wrap_db_fetch (void *dbf, gnutls_datum_t key)</code> Defined in 'gntlsfnc.web'.	[Function]
<code>int wrap_db_delete (void *dbf, gnutls_datum_t key)</code> Defined in 'gntlsfnc.web'.	[Function]
<code>int print_info (gnutls_session_t session)</code> Defined in 'gntlsfnc.web'.	[Function]
<code>const char * bin2hex (const void *bin, size_t bin_size)</code> Defined in 'gntlsfnc.web'.	[Function]
<code>void print_x509_certificate_info (gnutls_session_t session)</code> Defined in 'gntlsfnc.web'.	[Function]
<code>int extract_dn_fields (gnutls_x509 crt_t& cert, X509_Cert_Type* x509_cert = 0, bool subject = true, Scan_Parse_Parameter_Type* param = 0)</code> Defined in 'gntlsfnc.web'. The <code> bool subject </code> argument should be <code> false </code> when this function is used for the issuer's certificate.	[Function]
<code>void* listen_auth (void* v)</code> Defined in 'lstnfncs.web'. Thread function, called via 'pthread_create' in 'main' of 'optdbsrv'. This function "listens" in an endless "accept" loop on a port (5556 per default) for incoming connections with authentication/authorization using X.509 certificates. Theoretically, any application could contact the server, but in practice it only makes sense for 'optdbcli' (or 'optdbcli_1') to do this. After contact is initiated, a GNUTLS session is started and X.509 certificates are exchanged for mutual authentication/authorization. If this fails, the connection is broken off. If it succeeds, it will a new thread with the function 'connect_func' as its thread function. First, however, it creates a new object of type 'Scan_Parse_Parameter_Type'. A pointer to this object is passed to 'pthread_create' as a parameter and on to 'connect_func'. 'listen_func' sets several data members of this 'Scan_Parse_Parameter_Type'; in particular,	[Function]

`'listen_func'` creates a connection to the MySQL database and this connection is represented by `'MYSQL* Scan_Parse_Parameter_Type::mysql_ptr'`. (If `'listen_func'` fails to create the connection to the database, the thread will exit unsuccessfully.

The new thread is created with the attribute `'PTHREAD_CREATE_DETACHED'`, so that `'listen_func'` doesn't have to join with it. There's no need to check whether `'connect_func'` succeeds or fails and any errors or warnings will be logged.

Theoretically, if no errors occur, `'listen_auth'` could run forever, waiting for and handling incoming connections. However, it does contain some "clean-up" code following the main "accept" loop.

`void* listen_non_auth (void* v)` [Function]

Defined in `'lstnfncs.web'`.

Thread function, called via `'pthread_ctreate'` in `'main'` of `'optdbsrv'`. It is analogous to `'listen_func'` (see above), except that `'listen_non_auth'` doesn't require any form of authentication/authorization on the part of the client. It is assumed that any authentication/authorization required has been performed in some other way before the client application has been called.

The default port on which `'listen_non_auth'` listens is 5557.

`int tcp_connect (string ip_address = "127.0.0.1", string port_str = "5556")` [Function]

Defined in `'helper.web'`.

`void tcp_close (int sd)` [Function]

Defined in `'helper.web'`.

`void* get_in_addr (struct sockaddr *sa)` [Function]

Defined in `'helper.web'`.

12 X509_Cert_Type Reference

`class X509_Cert_Type` is defined in `'x509cert.web'`.

The following classes and functions are friends of `X509_Cert_Type`.

Classes:

`Distinguished_Name_Type`

Chapter 19 [Distinguished_Name_Type Reference], page 66

`Scan_Parse_Parameter_Type`

Chapter 21 [Scan_Parse_Parameter_Type Reference], page 72.

Functions:

`connect_func`

Chapter 7 [Server Connection Function Reference], page 43.

`extract_dn_fields`

Section 11.4 [TLS Functions], page 53

`verify_certificate_chain`

Section 11.4 [TLS Functions], page 53

12.1 Data Members

<code>string organization</code>	[Private variables]
<code>string organizationalUnitName</code>	
<code>string commonName</code>	
<code>string countryName</code>	
<code>string localityName</code>	
<code>string stateOrProvinceName</code>	
<code>unsigned int serialNumber</code>	
<code>time_t Validity_notBefore</code>	
<code>time_t Validity_notAfter</code>	
<code>X509_Cert_Type* issuer_cert</code>	
<code>string user_name</code>	
<code>unsigned int user_id</code>	

12.2 Member Functions

<code>void X509_Cert_Type (void)</code>	[Default constructor]
---	-----------------------

```

void X509_Cert_Type (unsigned int sserialNumber,                                [Constructor]
    X509_Cert_Type *iissuer_cert = 0,
    string oorganization = "",
    string oorganizationalUnitName = "",
    string ccommonName = "",
    string ccountryName = "",
    string llocalityName = "",
    string sstateOrProvinceName = "",
    unsigned int uuser_id = 0,
    string uuser_name = "",
    time_t VValidity_notBefore = 0,
    time_t VValidity_notAfter = 0)

int set (unsigned int sserialNumber,                                           [Function]
    X509_Cert_Type* iissuer_cert = 0,
    string oorganization = "",
    string oorganizationalUnitName = "",
    string ccommonName = "",
    string ccountryName = "",
    string llocalityName = "",
    string sstateOrProvinceName = "",
    unsigned int uuser_id = 0,
    string uuser_name = "",
    time_t VValidity_notBefore = 0,
    time_t VValidity_notAfter = 0)

void show (string s = "", Scan_Parse_Parameter_Type * param = 0, bool show_issuer = false) [Function]

bool is_in_database (Scan_Parse_Parameter_Type * param = 0)                    [Function]

int insert (Scan_Parse_Parameter_Type * param = 0)                            [Function]

string get_database_username (Scan_Parse_Parameter_Type * [Function]
    param = 0)

```

13 Proxy Credential Function Reference

The variables, constants and functions described in this chapter are declared and/or defined in ‘dhprxcrt.web’.

string certtool_path [Constant]
 File-local in ‘dhprxcrt.web’. The value of this constant depends on conditionally compiled code. On `optinum-srv.gwdg.de`, it’s ‘/opt/optinum/dbsrvcli/bin/certtool’. Otherwise it’s `certtool`, which implies that the path to the latter must be listed in the ‘PATH’ variable defined in the user’s environment.

int handle_proxy_cred (*string local_proxy_cred_filename* = *proxy_cred_filename*) [Function]

Return values:

- 0 Success
- 1 Local proxy credential filename is empty
- 2 Failed to open local proxy credential file
- 3 certtool failed
- 4 fread error: Failed to read output of `certtool`.
- 5 Too many characters in `certtool` output (> 1048575)
- 6 Failed to verify certificate chain: Expired
- 7 Failed to verify certificate chain: Other reason
- 8 popen error: Call to `certtool -i` failed
- 9 fread error: Failed to read output of `certtool -i`
- 10 Failed to find “Validity” information in proxy credential
- 11 Failed to find “Validity: Not Before” information in proxy credential
- 12 Failed to find “Validity: Not After” information in proxy credential
- 13 popen error: Calls to `date` failed
- 14 fscanf or getline error: Failed to read output of calls to `date`
- 15 Failed to set time in “seconds since the epoch” correctly
- 16 proxy credential not yet valid: Present time < “Not Before” value
- 17 proxy credential expired: Present time > “Not After” value

int extract_distinguished_name (*string cert_filename*, *string &dn*, *bool is_proxy* = *false*) [Function]

14 Scanner Server

The input file for the server's scanner is `'scanner.web'`.

14.1 Scanner Server Variables

`bool DEBUG_SCANNER` [Variable]

14.2 Scanner Server Functions

`int yylex (YYSTYPE *lvalp, yyscan_t parameter)` [Function]

`int yywrap (yyscan_t parameter)` [Function]

`int yyerror (void* v, char const* s)` [Function]

15 Parser Server

Terminal symbols (a.k.a., “tokens”) in Bison are implemented as preprocessor macros, defined in the header file created by Bison when it’s invoked using the `-d` or `--defines` option. This header file must be included in all C or C++ source files that refer to these terminal symbols. Since CPP (the C preprocessor) has no concept of namespaces, it is therefore necessary to ensure that the names of these terminal symbols differ from those of any other entities defined in the source code. For this reason, all of the terminal symbols defined by the programmer (as opposed to Bison itself) have the suffix “_YY”. Similarly, the terminal symbols in the parser for the client have the suffix “_ZZ” (see Chapter 17 [Parser Client], page 62).

15.1 Parser Server Symbols, Rules and States

The contents of the file ‘`parser.output`’, which is generated automatically by GNU Bison and contains information about the server parser, is contained in a separate document.

15.2 Parser Server Functions

`int yyparse (yyscan_t parameter)` [Function]
Definition generated by GNU Bison.

16 Scanner Client

The input file for the client's scanner is `'scnrclnt.web'`.

16.1 Scanner Client Variables

`bool DEBUG_SCANNER` [Variable]

16.2 Scanner Client Functions

`int zzlex (YYSTYPE *lvalp, yyscan_t parameter)` [Function]

`int zzwrap (yyscan_t parameter)` [Function]

`int zzerror (void* v, char const* s)` [Function]

17 Parser Client

Unlike the server's parser, the client's parser only processes input that comes from the peer, not input written by the user or generated by an external program. The user controls the behaviour of the client by means of Section 2.1 [Command Line Arguments and Options], page 6, not by means of an input file. (It would be possible to change this in the future, if it would ever seem desirable.) There is therefore no need for the input format of the client's parser to be particularly user-friendly. For example, all letters in keywords are uppercase. This is a safety measure that makes it less likely that an unquoted string would conflict with a keyword.

The server uses threads to process the commands sent to it from the client, so that the order of its responses to the client doesn't necessarily (and probably won't) correspond to the order of the commands. For this reason, the responses must contain more information than would otherwise be the case. For example, in the rule

```
SERVER_ZZ PREREQUISITES_ZZ FOR_ZZ ENTRY_ZZ entry_string_list
FOUND_ZZ prerequisite_list,
```

the *non-terminal symbols* 'entry_string_list' and 'prerequisite' list contain the names of the packages specified and possibly their version numbers, so that it's possible to associate this response with the command that caused it.

Terminal symbols (a.k.a., "*tokens*") in Bison are implemented as preprocessor macros, defined in the header file created by Bison when it's invoked using the `-d` or `--defines` option. This header file must be included in all C or C++ source files that refer to these terminal symbols. Since CPP (the C preprocessor) has no concept of namespaces, it is therefore necessary to ensure that the names of these terminal symbols differ from those of any other entities defined in the source code. For this reason, all of the terminal symbols defined by the programmer (as opposed to Bison itself) have the suffix "`_ZZ`". Similarly, the terminal symbols in the parser for the server have the suffix "`_YY`" (see Chapter 15 [Parser Server], page 60).

17.1 Parser Client Symbols, Rules and States

The contents of the file 'prsrcInt.output', which is generated automatically by GNU Bison and contains information about the client parser, is contained in a separate document.

17.2 Parser Client Functions

```
int zzparse (yyscan_t parameter)
```

[Function]

Definition generated by GNU Bison.

18 Entry_Type Reference

class `Entry_Type` is defined in `'entries.web'`.

class `Entry_Type` represents a software package within the Installer application (`'optdbcli/optdbsrv'`) and corresponds to the database table `'Entries'`. See Section 30.6 [Entries Database Table], page 99. That is, the data members of `Entry_Type` correspond to the fields in the `'Entries'` database table. This implies that the type of a data member of `'Entry_Type'` must be compatible with the type of its corresponding field in the database table.

When a user adds software package information with the `'START_ENTRY ... END_ENTRY'` command (see Section 4.4.1 [Adding Software Package Information], page 20), the data for each package (except for data from files, which is transferred separately) is first stored in an `Entry_Type` object and then written to the database by the function `Scan_Parse_Parameter_Type::write_entries`. See Section 21.2.3.1 [Functions for Adding Entries], page 75.

It would be possible to reverse the procedure and create an `Entry_Type` object using the data from a row of the `'Entries'` table, but in practice, this is never required. The function `Scan_Parse_Parameter_Type::get_entries` does create `Entry_Type` objects containing the data from the `'package_name'` and `'package_version'` columns of the `'Entries'` table. See Section 21.2.3.5 [Fetching Entry Information], page 78.

The following classes and functions are friends of class `Entry_Type`.

Classes:

`Scan_Parse_Parameter_Type`

Chapter 21 [Scan_Parse_Parameter_Type Reference], page 72.

`less_timeout`

`equal_package_name_version_globus_site`

`less_package_name`

`less_package_version`

`less_package_name_version`

`less_package_name_globus_site_version`

`equal_package_name_globus_site`

Section 18.3 [Comparison Classes for Entry_Type], page 65.

Functions:

`exchange_data_with_client`

Section 7.1 [Exchange Data With Client], page 43.

`remote_install_rule_func`

Chapter 23 [Parser Function Reference], page 83.

`install_packages`

Chapter 10 [Software Installation], page 49.

`remote_install_func`

Section 10.4 [Remote Software Installation], page 51.

`yyparse` Section 15.2 [Parser Server Functions], page 60.

18.1 Data Members

```

int entry_id
string package_name
string package_version
string download_url
string download_command
string maintainer_name
string maintainer_email_address
string installation_script
int install_order
vector<Entry_Type> prerequisites
bool private_flag
bool encrypted_flag
string authorization
string installation_script_encrypted
string download_url_encrypted
string download_command_encrypted
string owner
unsigned int owner_key_id
time_t created
time_t last_modified
Scan_Parse_Parameter_Type* param
int remote_install_timeout
pthread_t remote_install_thread_id
pid_t remote_install_pid
    Two-element array, i.e., pid_t remote_install_pid[2].

struct timespec remote_install_timespec
int remote_install_retval
bool install_or_reinstall_flag
string stdout_filename
string stderr_filename
string globus_site
bool deleted

```

[Private variables]

18.2 Member Functions

18.2.1 Constructors

```

void Entry_Type (void)
void Entry_Type (string a, string b = "")
void Entry_Type (int id, string a = "", string b = "")
void Entry_Type (const Entry_Type &e)
void Entry_Type (const Entry_Type &e)

```

[Default constructor]
[Constructor]
[Constructor]
[Copy constructor]
[Destructor]

18.2.2 Operators

```
const Entry_Type& operator= (const Entry_Type &e)      [Assignment Operator]
int operator== (const Entry_Type &e)                  [const Operator]
    Equality predicate.
int operator< (const Entry_Type &e)                    [const Operator]
    Less-than predicate.
int operator> (const Entry_Type &e)                    [const Operator]
    Greater-than predicate.
```

18.2.3 Other Functions

```
int extract_prerequisites (Scan_Parse_Parameter_Type *      [const Function]
    param, stringstream & out_strm)
int show (string s = "", Scan_Parse_Parameter_Type* param = 0, [const Function]
    bool show_prereqs = true)
```

18.3 Comparison Classes for Entry_Type

class less_timeout, class less_package_name, class less_package_version, class less_package_name_version, class less_package_name_globus_site_version and class equal_package_name_globus_site class equal_package_name_version_globus_site are defined in 'entries.web'.

class Entry_Type is a friend of all of these classes.

18.3.1 Operators for Comparison Classes for Entry_Type

```
int less_timeout::operator() (const Entry_Type& e1,      [public const Operators]
    const Entry_Type& e2)
int less_package_name::operator() (const Entry_Type& e1, const
    Entry_Type& e2)
int less_package_version::operator() (const Entry_Type& e1, const
    Entry_Type& e2)
int less_package_name_version::operator() (const Entry_Type& e1, const
    Entry_Type& e2)
int less_package_name_globus_site_version::operator() (const
    Entry_Type& e1, const Entry_Type& e2)
int equal_package_name_globus_site::operator() (const Entry_Type& e1,
    const Entry_Type& e2)
int equal_package_name_globus_site::operator() (const Entry_Type& e,
    const pair<string, string> &s)
int equal_package_name_version_globus_site::operator() (const
    Entry_Type& e1, const Entry_Type& e2)
```

19 Distinguished_Name_Type Reference

class `Distinguished_Name_Type` is defined in `'dstngnmt.web'`.

A *Distinguished Name* consists of items of information from an X.509 certificate (or *proxy credential*). It is used by Globus Toolkit in the `'grid-mapfile'` for associating certificates with users' login names on Globus front-end computers (often referred to as *Globus front-ends*).

Example:

```
"/C=DE/O=GridGermany/OU=Gesellschaft fuer wissenschaftliche \
Datenverarbeitung mbH/CN=Laurence Finston"
```

class `Distinguished_Name_Type` contains data members for storing the individual items of information and various functions for constructing, setting and comparing `Distinguished_Name_Type` objects. See also Chapter 12 [X509_Cert_Type Reference], page 56.

The class `Scan_Parse_Parameter_Type` (see Chapter 21 [Scan_Parse_Parameter_Type Reference], page 72) and the functions `client_connect_auth` and `client_connect_non_auth` (see Chapter 8 [Client Connection Function Reference], page 44) are friends of `Distinguished_Name_Type`.

19.1 Data Members

```
string organization                                [Private variables]
string organizationalUnitName
string commonName
string countryName
string localityName
string stateOrProvinceName
string user_name
int user_id
```

19.2 Member Functions

```
void Distinguished_Name_Type (void)                [Default constructor]
void Distinguished_Name_Type                        [Constructor]
    (string oorganization,
     string oorganizationalUnitName = "",
     string ccommonName = "",
     string ccountryName = "",
     string llocalityName = "",
     string sstateOrProvinceName = "",
     unsigned int uuser_id = 0,
     string uuser_name = "")
int set (string distinguished_name_str, string* user_name = 0)    [Function]
void operator= (const X509_Cert_Type &x)            [Assignment operator]
    Defined in 'x509cert.web'.
```

<code>bool operator== (const Distinguished_Name_Type &d)</code> Equality operator.	[const Operator]
<code>bool operator!= (const Distinguished_Name_Type &d)</code> Inequality operator.	[const Operator]
<code>void clear (void)</code>	[Function]
<code>void show (string s = "")</code>	[const Function]

20 Response_Type Reference

`class Response_Type` is defined in `'rspnstp.web'`.

Objects of type `Response_Type` are used by both the server and the client to control their actions and to carry information from function-to-function within the program (in the case of the client) or the thread (in the case of the server).

As explained in Chapter 21 [Scan_Parse_Parameter_Type Reference], page 72, an object of type `class Scan_Parse_Parameter_Type` exists for each instance of `'optdbcli'` and each thread in `'optdbsrv'`. This object represents the “state”, so to speak, of a run of the client or a thread in the server.

`class Scan_Parse_Parameter_Type` contains a data member `deque<Response_Type> response_deque`. See Section 21.1 [Scan_Parse_Parameter_Type Data Members], page 73. `deque` is a *container class* from the C++ standard template library. A `deque` is like a stack, except that items can be pushed onto and popped off of both ends.

Many parser rules and functions within `'optdbsrv'` and `'optdbcli'` create “responses” in the form of objects of type `Response_Type` and push them onto the front or the back of `response_deque` during the course of execution: Onto the front, if an action needs to be carried out right away and onto the back, if it can wait until all the other pending operations have been done.

Normally, it is intended that the programmer only push items onto the front or back of `Scan_Parse_Parameter_Type::response_deque`. Finer control of the order in which responses are processed is not usually needed, especially considering that one will not normally know what responses are on the `deque`.

However, as a special case, the function `deque_exchange_front_2` (see Chapter 26 [Utility Function Reference], page 86) swaps the front two items on `response_deque`. Currently, this is only used in the server-side parser rule `'statement: CLIENT_YY SENDING_YY KEY_PAIR_YY STRING_YY STRING_YY'`, where it causes messages to be sent to the client in an order which seems to the author to make more sense than would otherwise be the case.

The communication between client and server is handled by the function `exchange_data_with_server` (see Section 8.1 [Exchange Data With Server], page 44) on the client side and `exchange_data_with_client` (see Section 7.1 [Exchange Data With Client], page 43) on the server side. As long as `Scan_Parse_Parameter_Type::response_deque` contains one or more responses, (i.e., objects of type `class Response_Type`), these functions pop the one at the front of the `deque` off, examine it, and carry out the necessary actions. Functions may be called that cause one or more new responses to be added to the front or back of the `deque`, either due to the functioning of the program itself, or because of messages sent from the peer. When there are no more responses left to be processed on either side, the communication between server and client is ended.

The following classes and functions are **friends** of `class Response_Type`.

Classes:

`Scan_Parse_Parameter_Type`

Chapter 21 [Scan_Parse_Parameter_Type Reference], page 72.

Functions:

client_connect_auth
 client_connect_non_auth
 exchange_data_with_server

Chapter 8 [Client Connection Function Reference], page 44.

connect_func
 connect_func_non_auth
 exchange_data_with_client

Chapter 7 [Server Connection Function Reference], page 43.

remote_install_rule_func

Chapter 23 [Parser Function Reference], page 83.

remote_install_func

Section 27.1 [Remote Installation], page 87.

sub_distribute_key_pair

Section 27.2 [GPG Key Distribution], page 88.

yyparse Section 15.2 [Parser Server Functions], page 60.

zzparse Section 17.2 [Parser Client Functions], page 62.

20.1 Data Members

string local_filename [Private data members]

string local_filename_1

string remote_filename

string entry_name

string version

int entry_id

int type [Private data member]

Value should be set to one of the “_TYPE” constants described below. If **type** is **COMMAND_ONLY_TYPE**, then only the string stored in **command** (see below) is sent to the peer.

string distinguished_name_str [Private data members]

int install_order

string command [Private data member]

bool entries_all [Private data members]

bool entries_single

bool entries_public

bool entries_user

bool entries_with_prerequisites

bool entries_with_dependents

int output_format

int entries_ctr

bool certificate_user

bool certificates_all


```

bool privileges_user
bool privileges_all
string local_stdout_filename
string local_stderr_filename
string remote_stdout_filename
string remote_stderr_filename
string globus_site
set<string> globus_site_set
map<int, int> show_level_map
unsigned int show_environment_shellscript_options

```

```

static const int NULL_RESPONSE_TYPE = 0           [Public static constants]
static const int INSTALLATION_SCRIPT_TYPE = 1
static const int INSTALLATION_SCRIPT_ENCRYPTED_TYPE = 2
static const int COMMAND_ONLY_TYPE = 3
static const int FETCH_INSTALLATION_SCRIPT_TYPE = 4
static const int FETCH_DOWNLOAD_INFO_TYPE = 5
static const int FETCH_AUTHORIZATION_TYPE = 6
static const int FETCH_PREREQUISITES_TYPE = 7
static const int FETCH_ENVIRONMENT_SHELLSCRIPT_TYPE = 8
static const int SEND_ENVIRONMENT_SHELLSCRIPTS_TYPE = 9
static const int RETURN_ENVIRONMENT_SHELLSCRIPT_TYPE = 10
static const int VALIDATE_DISTINGUISHED_NAME_TYPE = 11
static const int SHOW_ENTRIES_TYPE = 12
static const int SHOW_CERTIFICATES_TYPE = 13
static const int SHOW_ENVIRONMENT_SHELLSCRIPTS_TYPE = 14
static const int SHOW_GLOBUS_SITES_TYPE = 15
static const int SHOW_PRIVILEGES_TYPE = 16
static const int DOWNLOAD_URL_TYPE = 17
static const int DOWNLOAD_COMMAND_TYPE = 18
static const int DOWNLOAD_URL_ENCRYPTED_TYPE = 19
static const int DOWNLOAD_COMMAND_ENCRYPTED_TYPE = 20
static const int AUTHORIZATION_TYPE = 21
static const int PROXY_CREDENTIAL_TYPE = 22
static const int REMOTE_INSTALL_TYPE = 23
static const int SECRET_KEY_TYPE = 24
static const int PUBLIC_KEY_TYPE = 25
static const int KEY_PAIR_TYPE = 26
static const int FETCH_KEY_PAIR_TYPE = 27
static const int DISTRIBUTE_KEY_PAIR_TYPE = 28

```

Constants for setting the int type data member.

```

static const int OUTPUT_FORMAT_PLAIN_TEXT = 0       [Public static constants]
static const int OUTPUT_FORMAT_HTML = 1
static const int OUTPUT_FORMAT_TEX = 2
static const int OUTPUT_FORMAT_JAVASCRIPT = 3

```

Constants for setting the int output_format data member.

20.2 Member Functions

`void Response_Type (void)`

[Default constructor]

`int show (string s)`

[Function]

21 Scan_Parse_Parameter_Type Reference

`class Scan_Parse_Parameter_Type` is defined in ‘`scprpmtplib`’.

An object of type `class Scan_Parse_Parameter_Type` exists for each instance of ‘`optdbcli`’ and each thread in ‘`optdbsrv`’. This object represents the “state”, so to speak, of a run of the client or a thread in the server.

For this reason, many of the functions defined in the Installer package are member functions of `class Scan_Parse_Parameter_Type`. If they were not, they would need to have a pointer or reference to `class Scan_Parse_Parameter_Type` as an argument or information from an `class Scan_Parse_Parameter_Type` would have to be passed to them as arguments.

In particular, functions that access the MySQL database are usually member functions of `Scan_Parse_Parameter_Type`,¹ because each object of type `Scan_Parse_Parameter_Type` on the server-side receives a connection to the database.

This occurs in the functions `listen_auth` and `listen_non_auth`.
See Section 11.4 [TLS Functions], page 53.

The following classes and functions are friends of `Scan_Parse_Parameter_Type`.

Classes:

`Entry_Type`

Chapter 18 [Entry_Type Reference], page 63

`GPG_Key_Pair_Type`

Chapter 24 [GPG_Key_Pair_Type Reference], page 84

`X509_Cert_Type`

Chapter 12 [X509_Cert_Type Reference], page 56

Functions:

`accept_data`

Chapter 23 [Parser Function Reference], page 83

`client_connect_auth`

`client_connect_non_auth`

Chapter 8 [Client Connection Function Reference], page 44.

`connect_func`

`connect_func_non_auth`

Chapter 7 [Server Connection Function Reference], page 43.

`exchange_data_with_client`

Chapter 7 [Server Connection Function Reference], page 43.

`exchange_data_with_server`

Chapter 8 [Client Connection Function Reference], page 44

`extract_dn_fields`

Section 11.4 [TLS Functions], page 53

¹ `cull_database` is an exception, because it must run independently of any client/server communication. That is, it may run times when no `Scan_Parse_Parameter_Type` objects exist on the server-side. See Section 27.3 [Cull Database], page 88.

`hidden_install`
 Section 10.2 [Hidden Software Installation], page 50

`install_packages`
 Chapter 10 [Software Installation], page 49.

`init_globus_site_map`
 Chapter 26 [Utility Function Reference], page 86

`listen_auth`
`listen_non_auth`
 Section 11.4 [TLS Functions], page 53

`parse_input`
 Chapter 23 [Parser Function Reference], page 83

`remote_install_func`
 Section 27.1 [Remote Installation], page 87.

`remote_install_rule_func`
 Chapter 23 [Parser Function Reference], page 83.

`sub_distribute_key_pair`
 Section 27.2 [GPG Key Distribution], page 88.

`verify_cert2`
`verify_certificate_chain`
`verify_last_cert`
 Section 11.4 [TLS Functions], page 53

`yylex` Section 14.2 [Scanner Server Functions], page 59

`yyparse` Section 15.2 [Parser Server Functions], page 60.

`zzlex` Section 16.2 [Scanner Client Functions], page 61

21.1 Data Members

`bool` `PARSER_DEBUG` [Private variables]
`int` `thread_ctr`
`pthread_t` `thread_id`
`int` `start_condition`
`unsigned int` `MAX_BUFFER`
`string` `date_time_str`
`string` `in_filename`
`string` `out_filename`
`ostream*` `out_strm`
`stringstream` `msg_strm`
`Scan_Parse_Parameter_Type*` `up`

`int` `fd` [Private variable]
 File descriptor. In `connect_func`, passed as an argument to `gnutls_transport_set_ptr`. See Chapter 7 [Server Connection Function Reference], page 43, and Section “Core functions” in *The GnuTLS Manual*.

```

vector<class Entry_Type> entries [Private variables]
vector<class Entry_Type> emendations
vector<class Entry_Type> install_vector
gnutls_certificate_credentials_t * cert_cred_ptr
gnutls_anon_server_credentials_t * anon_server_cred_ptr
gnutls_anon_client_credentials_t * anon_client_cred_ptr
gnutls_x509_crt_t * ca_list
unsigned int ca_list_size
MYSQL * mysql_ptr
string user_name
unsigned int user_id
string user_dn

string effective_user_dn [Private variables]
string effective_user_name
unsigned int effective_user_id
    See Section 4.5 [Setting Effective User (optdbcli)], page 29, and Section 30.5 [Delegates
    Database Table], page 99.

deque<Response_Type> response_deque [Private variable]
pthread_mutex_t response_deque_mutex [Private variable]
    See Chapter 20 [Response_Type Reference], page 68.

gnutls_session_t * session [Private variables]
string prerequisite_list
bool server_finished
bool client_finished
vector<string> prerequisite_string_vector
vector<string> entry_string_vector
string download_url_string
string download_command_string
string remote_ip_address
int remote_port
unsigned int show_options
string proxy_cred_filename
set<string> globus_site_set
Globus_Site_Info_Type globus_site_options

static const unsigned int MAX_KEY_PAIR_SIZE [Private variable]
    Initialized to 65536 ( $= 2^{16}$ ).

char key_pair_str [Private variable]
    Array with 65536 elements.

string key_name [Private variables]
unsigned long int key_id
string session_id
Session_Data_Type * session_data
map<int, int> show_level_map

```

```

set<string> show_environment_shellscript_globus_site_set
string environment_globus_site
string show_environment_shellscript_globus_site
stringstream environment_strm
unsigned int environment_admin_flag
unsigned int show_environment_shellscript_options
vector<string> environment_shellscript_filename_vector

```

21.2 Member Functions

21.2.1 Constructors and Destructor

```

void Scan_Parse_Parameter_Type (void) [Default constructor]
~Scan_Parse_Parameter_Type (void) [Destructor]

```

21.2.2 Functions for Managing Users

```

string get_database_username (X509_Cert_Type &cert) [Functions]
string get_database_username (int user_id)
int validate_distinguished_name (Response_Type &response) [Function]
int check_grid_mapfile (string distinguished_name_str, [Function]
    X509_Cert_Type *cert = 0, string *user_name = 0, bool break_on_found
    = false)
int check_delegation (char *dn_str) [Function]
int show_certificates (Response_Type &response, char buffer[]) [Function]
int lookup (const Distinguished_Name_Type &dn) [Function]

```

21.2.3 Functions for Entries

21.2.3.1 Adding Entries

```

int write_entries (void) [Function]
    Defined in 'spptwren.web'. Writes software database entries to MySQL database,
    which implies that it is only used by optdbsrv.

```

Called in `parse_input` (see Chapter 23 [Parser Function Reference], page 83) after `yyparse` exits successfully (see Section 15.2 [Parser Server Functions], page 60), if the `vector<Entry_Type> param->entries` is non-empty, where `param` is a pointer to `Scan_Parse_Parameter_Type`, passed to `parse_input` as an argument.

The `Entry_Type` objects on the `vector param->entries` contain information about software packages sent to `optdbsrv` by the user by means of 'START_ENTRY ... END_ENTRY' commands in the input to `optdbcli`. See Section 4.4.1 [Adding Software Package Information], page 20. For example, this input:

```

START_ENTRY
PACKAGE_NAME      m4
PACKAGE_VERSION 1.4.15

```

```

DOWNLOAD_URL ftp://ftp.gwdg.de/pub/gnu/ftp/gnu/m4/m4-1.4.15.tar.gz
INSTALLATION_SCRIPT <install_m4.sh>
PUBLIC
END_ENTRY

```

causes the following SQL command to be generated and executed in `write_entries`:

```

insert into Entries (user_id, entry_id, package_name,
                    package_version, download_url,
                    download_command, maintainer_name,
                    maintainer_email_address, private,
                    created, last_modified, encrypted,
                    owner)
values (1, 1, 'm4', '1.4.15',
        'ftp://ftp.gwdg.de/pub/gnu/ftp/gnu/m4/m4-1.4.15.tar.gz',
        NULL, NULL, NULL, false, '2012-05-31 11:35:22',
        '2012-05-31 11:35:22', false, NULL)

```

Please note that this command does *not* store the file `'install_m4.sh'` containing the installation script in this row of the `'Entries'` table: `write_entries` generates requests for the installation script and any other files belonging to this database entry, which are sent to `optdbcli` by `exchange_data_with_client`, after `write_entries` and its caller `parse_input`, have exited. See Section 7.1 [Exchange Data With Client], page 43. Any data contained in files are thus sent separately. This reason for this is to avoid problems that may arise when transferring data. There is no predefined limit on the amount of data that a file may contain, so the amount of data to be transferred might be very large, which could be a potential source of problems. Sending data from files separately ensures that the database entry will not be corrupted at the time of creation because of a problem in transferring the data from the files.

int cond_append (*Entry_Type &e*) [Function]

Defined in `'sccprmpmtp.web'`. Called in `zzparse` (see Section 17.2 [Parser Client Functions], page 62) in the following rules:.

```

<statement> → SERVER_ZZ SENDING_ZZ <installation script>
              STRING_ZZ FOR_ZZ ENTRY_ZZ <entry string list>
              ORDER_ZZ INTEGER_ZZ

```

```

<statement> → SERVER_ZZ SENDING_ZZ <download info expression>
              FOR_ZZ ENTRY_ZZ <entry string list> ORDER_ZZ INTEGER_ZZ

```

```

<statement> → SERVER_ZZ SENDING_ZZ DOWNLOAD_URL_ENCRYPTED_ZZ
              FOR_ZZ ENTRY_ZZ <entry string list> ORDER_ZZ INTEGER_ZZ

```

```

<statement> → SERVER_ZZ SENDING_ZZ DOWNLOAD_COMMAND_ENCRYPTED_ZZ
              FOR_ZZ ENTRY_ZZ <entry string list> ORDER_ZZ INTEGER_ZZ

```

```

<statement> → SERVER_ZZ SENDING_ZZ AUTHORIZATION_ZZ
              FOR_ZZ ENTRY_ZZ <entry string list> ORDER_ZZ INTEGER_ZZ

```

```

(statement) → NO_ZZ INSTALLATION_SCRIPT_ZZ FOR_ZZ ENTRY_ZZ
              (entry string list) ORDER_ZZ INTEGER_ZZ

```

That is, this function is called in the course of the *local installation* of a software package. See Section 10.3 [Local Software Installation], page 51.

It conditionally appends `Entry_Type e` to `vector<Entry_Type> this->entries`. This function ensures that there is only one `Entry_Type` object per package on `entries` (where package name and package version uniquely identify a package).

It tries to find an object in `entries` equal to the argument `Entry_Type e`. The C++ Standard Template Library function `find`² uses the equality operator `Entry_Type::operator==`, which returns `true` if the `string` data member `package_name` and `package_version` of two `Entry_Type` objects are equal.

If `entries` contains a matching element, the values of `int Entry_Type::install_order` are compared and the existing `Entry_Type` on the vector has its `install_order` member set to the maximum of the two values. This ensures that it will be installed before any packages that require it.

21.2.3.2 Modifying Entries

```

int add_installation_script (string curr_filename,           [Function]
                             string curr_package_name,
                             string curr_package_version = "",
                             int encrypted = 0)

```

```

int add_authorization (string curr_string,                   [Function]
                       string curr_package_name,
                       string curr_package_version = "")

```

These functions are needed by the web application, because data from files must first be sent to the computer where the web application is running and from there to the server. When the data from a file is sent to the web application, it is stored in a temporary file whose name is not known in advance. The web application therefore saves the data, sends the commands for creating entries that it has received from the user, and appends commands for adding data from the files in the appropriate

```

int add_download_info (string curr_string,                   [Function]
                       string curr_package_name,
                       string curr_package_version = "",
                       int download_info_type = 0)

```

```

int write_emendations (void)                                [Function]

```

21.2.3.3 Deleting Entries

See also Section 4.4.2 [Deleting Software Package Information (optdbcli)], page 24.

```

int delete_entries (string package_name = "", string         [Function]
                   package_version = "")

```

Deletes all of the entries with ‘`package_name`’ and ‘`package_version`’ (if the latter is present) for the current user. Normally, it shouldn’t be possible for there to be

² With arguments of type `vector<Entry_Type>::iterator` and `Entry_Type`, e.g.,
`vector<Entry_Type>::iterator iter = find(entries.begin(), entries.end(), e).`

more than one entry for a given ‘package_name’ with the same ‘package_version’ (or an empty ‘package_version’). However, if there is, this function will delete all of them. It also deletes all rows in the Prerequisites table where the ‘entry_id’ or ‘prerequisite_id’ corresponds to the values of ‘Entries.entry_id’ found for ‘package_name’, ‘package_version’ for the current user.

```
int mark_entries (bool deleted = true,                                     [Function]
                  string package_name = "",
                  string package_version = "")
```

Defined in ‘sptenfn.web’. Marks or unmarks all of the entries with |package_name| and |package_version| (if the latter is present) for the current user (identified by |curr_user_id|) as deleted (unless it’s already been marked). Normally, it shouldn’t be possible for there to be more than one entry for a given |package_name| with the same non-empty |package_version|. However, if there is, this function will mark or unmark all of them.

21.2.3.4 Showing Entries

```
int show_entries (Response_Type &response, char buffer[])               [Function]
    See also Section 4.4.3 [Showing Software Package Information (optdbcli)], page 28.
```

21.2.3.5 Fetching Entry Information

```
int get_entries (vector<Entry_Type> &ev, string package_name,           [Function]
                 string package_version = "")
int fetch_installation_script (Response_Type &response)                [Function]
int fetch_download_info (Response_Type &response)                      [Function]
int fetch_authorization (Response_Type &response)                      [Function]
int get_encrypted (const int &entry_id, string *key_name = 0,          [Function]
                  string *owner = 0)
```

21.2.4 Functions for Prerequisites

```
int fetch_prerequisites (Response_Type &response)                      [Function]
int sub_fetch_prerequisites (MYSQL_RES *result,                        [Function]
                             unsigned int row_ctr,
                             unsigned int field_ctr,
                             Entry_Type &entry)
int request_prerequisites (string package_name, string                 [Function]
                           package_version = "")
int add_prerequisite (int curr_entry_id, const Entry_Type &e)          [Function]
int add_prerequisite (string package_name,                             [Function]
                     string package_version,
                     string prerequisite_name,
                     string prerequisite_version)
```

The first version is called in ‘Scan_Parse_Parameter_Type::write_entries’ and in the second version, which is called in ‘yyparse’.

Return values:

- 0 Success
- 1 SQL query, `mysql_fetch_row` or `strtol` failed
- 2 No entry that matches this prerequisite
- 3 Ambiguous match for prerequisite
- 4 Duplicate prerequisite

```
int delete_prerequisites (bool delete_all,           [Function]
                        string package_name,
                        string package_version,
                        string prerequisite_name = "",
                        string prerequisite_version = "")
```

21.2.5 Database Functions

```
int check_database_tables (void)                   [Function]
    Defined in 'spptdbfn.web'.
```

```
int lock_tables (vector<string> *table_vector, bool unlock = false, [Function]
                string table = "")
```

```
int lock_table (string s)                          [inline function]
```

```
int unlock_tables (void)                          [inline function]
```

`lock_tables` is defined in 'spptdbfn.web'. The others are defined within the `class` declaration of `Scan_Parse_Parameter_Type` in 'scprpmtmp.web'. They simply call `lock_tables` with appropriate arguments.

`lock_table` (with a `string` argument) is defined in order to make it more convenient to lock a single table, i.e., so the programmer doesn't have to declare a `vector<string>` and push the name of the table onto it just in order to call `lock_tables`.

`unlock_tables` doesn't require any arguments, because the MySQL command for unlocking tables doesn't take any, that is, all tables locked by a MySQL client session are unlocked.

21.2.6 Globus Site Functions

The functions in this subsection are used for adding, deleting and editing data pertaining to Globus sites. This data is stored in the database tables 'Globus_Sites' and 'Globus_Sites_Admins'. See Section 30.8 [Globus_Sites Database Table], page 100, and Section 30.13 [Globus_Sites_Admins Database Table], page 101. They are declared in 'scprpmtmp.web' and defined in 'spptgsfn.web'.

These functions use the `Scan_Parse_Parameter_Type` member function `check_privileges` (see Section 21.2 [Scan_Parse_Parameter_Type Member Functions], page 75) to determine whether the user is permitted to perform the action in question, i.e, whether or not he or she may access and/or change the relevant data in the 'Globus_Sites' and 'Globus_Sites_Admins' database tables. If not, the server sends an error code to the client and the latter issues an error or a warning message, depending on the particular action requested.

<code>int add_globus_site (void)</code>	[Function]
<code>int mark_globus_site (bool delete_gs, string globus_site, int enable_optional = 0)</code>	[Function]
<code>int show_globus_sites (Response_Type &response, char buffer[])</code>	[Function]
<code>int delete_globus_sites (bool finally)</code>	[Function]
<code>int enable_globus_site (string globus_site, bool enable = true)</code>	[Function]
<code>int disable_globus_site (string globus_site)</code> Calls <code>enable_globus_site</code> (see above) with <code>enable = false</code> .	[inline function]
<code>int edit_globus_site (string globus_site, int set_or_unset)</code>	[Function]
<code>int check_globus_site_admin (unsigned int user_id, string globus_site)</code>	[Function]
<code>int edit_globus_site_admin (string globus_site, int admin_user_id, int action)</code>	[Function]

21.2.7 Functions for Privileges

<code>int check_privileges (unsigned int &privilege_val, bool quit_on_success = false, string privilege = "", string globus_site = "", int *save_globus_site_id = 0)</code> Used by various other <code>Scan_Parse_Parameter_Type</code> member functions for Globus sites. See Section 21.2.6 [Globus Site Functions], page 79, above.	[Function]
<code>int show_privileges (Response_Type &response, char buffer[])</code>	[Function]

21.2.8 Environment Shellscript Functions

<code>int fetch_environment_shellscripts (Response_Type &response)</code>	[Function]
<code>int store_environment_shellscript (void)</code>	[Function]
<code>int delete_environment_shellscripts (int id = 0, string globus_site = "")</code>	[Function]
<code>int show_environment_shellscripts (Response_Type &response, char buffer[])</code>	[Function]
<code>int enable_environment_shellscript (unsigned int id, bool disable_flag = false)</code>	[Function]

21.2.9 Functions for GPG Keys

<code>string get_key_name (const string &owner)</code>	[Function]
<code>int store_key_pair (const char *filename)</code>	[Function]
<code>int store_public_key (const char *filename)</code>	[Function]
<code>int fetch_key_pair (Response_Type &response)</code>	[Function]
<code>int get_key_pair (void)</code>	[Function]

<code>int install_key_pair (const char *key_name, const char *filename)</code>	[Function]
<code>int get_optinum_installer_gpg_pubkey_fingerprint (void)</code>	[Function]
<code>int distribute_key_pair (Response_Type &response)</code>	[Function]
<code>int check_public_key (void)</code>	[Function]
<code>int delete_key_pair (void)</code>	[Function]
<code>int delete_public_key (void)</code>	[Function]

21.2.10 Session Data Functions

<code>int get_session_data (void)</code>	[Function]
--	------------

21.2.11 Other Member Functions

<code>int send_data_to_peer (Response_Type &response, bool delete_local_file = false)</code>	[Function]
Defined in 'spptsdtp.web'. Called in the functions <code>exchange_data_with_client</code> and <code>exchange_data_with_server</code> . See Section 7.1 [Exchange Data With Client], page 43, and Section 8.1 [Exchange Data With Server], page 44.	
<code>int submit_mysql_query (string query, MYSQL_RES *&result, unsigned int *row_ctr = 0, unsigned int *field_ctr = 0, long *affected_rows = 0)</code>	[Function]
<code>void show (string s = "Scan_Parse_Parameter_Type:")</code>	[Function]

22 Globus_Site_Info_Type Reference

class `Session_Data_Type` is defined in `'glsinftp.web'`. class `Scan_Parse_Parameter_Type` and the functions `main`, `yyparse` and `check_globus_site` are friends of class `Globus_Site_Info_Type`. See Chapter 21 [Scan_Parse_Parameter_Type Reference], page 72, Section 15.2 [Parser Server Functions], page 60, and Section 29.1 [Start Remote Installation], page 92.

class `Scan_Parse_Parameter_Type` has a data member of type `Globus_Site_Info_Type`. This object is used in parser rules for the server's parser (i.e., `yyparse`) for storing the values of options for commands such as `'add_globus_site'` and `'edit_globus_site'`.

22.1 Data Members

```
int globus_site_id [Private variables]
string hostname
string ip_address
string interactive_node_name
string interactive_node_ip_address
string append_ld_library_path
string append_path
string optdbcli_path
bool replace

signed short enabled [Private variable]
    Using a signed short for the type of this variable makes it possible to determine
    whether it's been set or not:
    0          Not set
    -1         Disabled
    1          Enabled
```

22.2 Member Functions

```
Globus_Site_Info_Type (void) [Default constructor]

int show (string s) [Function]

int clear (void) [Function]
```

23 Parser Function Reference

These functions are defined in ‘`prsrfncs.web`’.

```
int parse_input (FILE *fp, char *buffer, Scan_Parse_Parameter_Type *param) [Function]

int accept_data (Scan_Parse_Parameter_Type *param, char *filename, [Function]
                 int entry_id, int file_type, string *return_filename = 0)

int remote_install_rule_func (Scan_Parse_Parameter_Type *param, [Function]
                              int install_or_reinstall, string package_name, string package_version, string
                              globus_site, string stdout_filename, string stderr_filename, int timeout, int
                              reinstall_prerequisites, string link_expression)
```

24 GPG_Key_Pair_Type Reference

class `GPG_Key_Pair_Type` is defined in `'scprpmtplib'`. As a convenience, `Gpg_Key_Pair_Type` is defined to be a synonym for `GPG_Key_Pair_Type`:

```
typedef GPG_Key_Pair_Type Gpg_Key_Pair_Type
```

The class `Scan_Parse_Parameter_Type` (see Chapter 21 [Scan_Parse_Parameter_Type Reference], page 72) and the function `sub_distribute_key_pair` (see Section 27.2 [GPG Key Distribution], page 88) are friends of `GPG_Key_Pair_Type`.

24.1 GPG_Key_Pair_Type Data Members

```
Scan_Parse_Parameter_Type * param           [Private variables]
string globus_site
int return_value
unsigned int sub_thread_ctr
pthread_t ptid
```

24.2 GPG_Key_Pair_Type Member Functions

```
GPG_Key_Pair_Type (void)                   [Default constructor]
void show (string s = "", bool show_param = false) [Function]
```

25 Session_Data_Type Reference

class Session_Data_Type is defined in 'sssndttp.web'. class Scan_Parse_Parameter_Type is a friend of class Session_Data_Type. See Chapter 21 [Scan_Parse_Parameter_Type Reference], page 72.

25.1 Data Members

string session_id [Private variables]

25.2 Session_Data_Type Member Functions

Session_Data_Type (void) [Default constructor]

Session_Data_Type (const Session_Data_Type &s) [Copy constructor]

Session_Data_Type (string) [Constructor]

~Session_Data_Type (void) [Destructor]

const Session_Data_Type operator= (const Session_Data_Type &s) [Assignment operator]

int show (string s = "", Scan_Parse_Parameter_Type *param = 0) [const function]

26 Utility Function Reference

<code>int init_globus_site_map (void)</code> Defined in 'lstnfncs.web'.	[Function]
<code>void delete_temp_files (void)</code>	[Function]
<code>void delete_fifos (void)</code>	[Function]
<code>const char* get_dir (uid_t uid)</code>	[Function]
<code>int check_dir (const string path, bool exit_on_found = false)</code>	[Function]
<code>int dir_is_empty (const string path)</code>	[Function]
<code>int get_date_time (time_t* t = 0, char *d = 0, time_t s = 0)</code>	[Function]
<code>int create_temp_file (string& filename, ofstream *out_strm = 0, string prefix = "FF")</code>	[Function]
<code>string get_proxy_cred_filename (string user_name)</code>	[Function]
<code>template <class C> int deque_exchange_front_2 (deque<C> &d)</code>	[Template function]
<code>int set_debug_level (bool &DEBUG)</code>	[Function]

27 Thread Function Reference

27.1 Remote Installation

`void* remote_install_func (void *v)` [Thread function]
 Defined in ‘`thrdfnecs.web`’.

Return values:

- 0 Success
- 1 In first child process (call to `sudo` for `strtinst` or `strtinst.sh`): `exec1` returned. It shouldn’t return at all. (see Section 29.1 [Start Remote Installation], page 92.)
- 2 First child process failed to exit.

1000 + exit status of first child process

First child process exited unsuccessfully.

If the call to `strtinst` in the child process fails because its call to `handle_proxy_cred` failed, `strtinst` returns 100 + the error code returned by `handle_proxy_cred`.

(Depending on the platform, it may not be possible for `remote_install_func` to invoke `strtinst` via `sudo`, in which case the “wrapper script” `strtinst.sh` is invoked instead. `strtinst.sh` simply returns the exit status of `strtinst`.)

If this function fails because `strtinst` failed, 1000 + the latter’s error code is returned.

By this means, the caller of this function can determine whether the ultimate cause of the failure was an error in `handle_proxy_cred`, and if so, what the specific error was. That is, if ‘`r`’ is the exit status of this function, and $(1000 - 100 - r) > 0$, then $(1000 - 100 - r)$ is the exit status of `handle_proxy_cred`.

The only caller of this function as of February 9, 2012 is `exchange_data_with_client`, which is defined in ‘`connect.web`’. See Section 7.1 [Exchange Data With Client], page 43. It calls this function by means of `pthread_create`. See Chapter 7 [Server Connection Function Reference], page 43.

- 3 Filename returned by child process too long (> 511 characters)
- 4 In second child process (call to `sudo` for `instchwn` or `instchwn.sh`): `exec1` returned. It shouldn’t return at all.
- 5 Second child process failed to exit

10,000 + exit status of second child process

Second child process exited unsuccessfully

(Depending on the platform, it may not be possible for `remote_install_func` to invoke `instchwn` via `sudo`, in which case the “wrapper script”

`instchwn.sh` is invoked instead. `instchwn.sh` simply returns the exit status of `instchwn`.)

If this function fails because `instchwn` failed, 10,000 + the latter's error code is returned.

By this means, the caller of this function can determine whether the ultimate cause of the failure was an error in `instchwn`, and if so, what the specific error was.

As of February 9, 2012, `exchange_data_with_client`, simply tests whether this function returns a value > 10,000 in order to determine whether it failed because of a failure of `instchwn`. It does not test for the specific cause.

- 6 No Globus site specified
- 7 Globus site not found on `globus_site_map`

27.2 GPG Key Distribution

`void* sub_distribute_key_pair (void *v)` [Thread function]

27.3 Cull Database

`void* cull_database (void *v)` [Thread function]

Defined in `'thrdfncs.web'`. Deletes expired rows (see below) from the database tables `'Entries'`, `'Session_Data'` and `'Globus_Sites'`. These deletions may cause rows in `'Prerequisites'` and `'Globus_Sites_Admins'` to become unnecessary, so these rows are deleted as well. See Chapter 30 [Database Tables and Views], page 97.

The tables `'Entries'`, `'Globus_Sites'` and `'Session_Data'` each have a column containing a timestamp value. In `'Entries'` and `'Globus_Sites'`, it is the column `'last_modified'` while in `'Session_Data'`, it is the column `'timestamp'`. A row is expired when the value in this column is less (i.e., older) than one day.

Currently, `cull_database` is called (asynchronously) in a (detached) thread of its own every two hours, so that a row will probably not be deleted immediately when the row expires. On the other hand, there should never be any expired rows that have expired more than 26 hours before.

28 Hidden Installation Reference (scrinstl)

28.1 Global Variables

<code>vector<string> delete_files</code>	[Variable]
<code>char process_owner_dir[128]</code>	[Variable]
<code>bool proxy_cred_verified</code> Default false.	[Variable]
<code>pthread_mutex_t cerr_mutex</code>	[Variable]
<code>string grid_mapfile_filename</code> Default set by means of conditionally compiled code, based on the Globus front-end, on which 'scrinstl' will be running.	[Variable]

28.2 Global Functions

<code>int main (int argc, char **argv)</code> Defined in 'scrinstl.web'.	[Function]
<code>void finish (void)</code> Defined in 'scrinstl.web'.	[Function]
<code>void* connect_func (void *v)</code> Defined in 'cnctfnc.web'.	[Function]
<code>void* wait_func (void *v)</code> Defined in 'cnctfnc.web'.	[Function]
<code>int handle_proxy_cred (string proxy_cred_filename, string ca_cert_filename)</code> Defined in 'hdprxcrt.web'. Return values:	[Function]
0 Success	
1 Local proxy credential filename is empty	
2 Failed to open local proxy credential file	
3 <popen call="" certtool="" error:="" failed<="" p="" to=""></popen>	
4 fread error: Failed to read output of certtool.	
5 Too many characters in certtool output (> 1048575)	
6 Failed to verify certificate chain: Expired	
7 Failed to verify certificate chain: Other reason	
8 popen error: Call to 'certtool -i' failed	
9 fread error: Failed to read output of 'certtool -i'	

10	Failed to find “Validity” information in proxy credential	
11	Failed to find “Validity: Not Before” information in proxy credential	
12	Failed to find “Validity: Not After” information in proxy credential	
13	popen error: Calls to <code>date</code> failed	
14	<code>fscanf</code> or <code>getline</code> error: Failed to read output of calls to <code>date</code>	
15	Failed to set time in “seconds since the epoch” correctly	
16	proxy credential not yet valid: Present time < “Not Before” value	
17	proxy credential expired: Present time > “Not After” value	
<hr/>		
<code>int</code>	<code>gen_enc_rd_str (string owner, char * buffer_enc, size_t size_enc, char * buffer_pln, size_t size_pln)</code>	[Function]
Generate encrypted random string. Defined in ‘ <code>rdmstrfs.web</code> ’. Currently not used. A random string generated by this function can be used in a “handshake” procedure between ‘ <code>scrinstl</code> ’ and ‘ <code>optdbcli</code> ’.		
<hr/>		
<code>void</code>	<code>show_user_info (int uid = 0, const char * name = 0, const char * message = 0)</code>	[Function]
Defined in ‘ <code>utilfnscs.web</code> ’.		
<hr/>		
<code>int</code>	<code>get_uid (char * name, char * dir)</code>	[Function]
Defined in ‘ <code>utilfnscs.web</code> ’.		
<hr/>		
<code>const char*</code>	<code>get_dir (uid_t uid)</code>	[Function]
Defined in ‘ <code>utilfnscs.web</code> ’.		
<hr/>		
<code>const char*</code>	<code>get_name (uid_t uid)</code>	[Function]
<hr/>		
<code>int</code>	<code>decrypt (string in_filename, unsigned int key_id, char * buffer = 0, size_t size = 0, string out_filename = “”)</code>	[Function]
<hr/>		
<code>int</code>	<code>check_checksum (string filename, string checksum)</code>	[Function]
<hr/>		
<code>void</code>	<code>lock_cerr_mutex (void)</code>	[inline functions]
<code>void</code>	<code>unlock_cerr_mutex (void)</code>	
<hr/>		
<code>void</code>	<code>delete_temp_files (void * v)</code>	[Function]
<hr/>		
<code>void</code>	<code>wait_for_child_processes (void * v)</code>	[Function]
<hr/>		
<code>string</code>	<code>get_datestamp (void)</code>	[Function]
<hr/>		
<code>int</code>	<code>check_distinguished_name (string caller, string caller_dn)</code>	[Function]

28.3 Distinguished_Name_Type

`struct Distinguished_Name_Type` is defined in ‘`dstngnmt.web`’. It is nearly identical to `class Distinguished_Name_Type` from the ‘`dbsrvcli`’ (i.e., ‘`optdbcli/optdbsrv`’) package. See Chapter 19 [Distinguished_Name_Type Reference], page 66. Aside from `Distinguished_Name_Type` being declared as a `class` in ‘`dbsrvcli`’ and as a `struct` in ‘`scrinstl`’, the version in ‘`dbsrvcli`’ has `friend` declarations for functions that only exist in that package, and a couple of member functions that aren’t needed in ‘`scrinstl`’.

28.3.1 Distinguished_Name_Type Data Members

```
string organization [Public variables]
string organization
string organizationalUnitName
string commonName
string countryName
string localityName
string stateOrProvinceName
string user_name
int user_id
```

28.3.2 Distinguished_Name_Type Member Functions

```
void Distinguished_Name_Type (void) [Default constructor]
void Distinguished_Name_Type [Constructor]
    (string oorganization,
     string oorganizationalUnitName = "",
     string ccommonName = "",
     string ccountryName = "",
     string llocalityName = "",
     string sstateOrProvinceName = "",
     unsigned int uuser_id = 0,
     string uuser_name = "")
int set (string distinguished_name_str, string* user_name = 0) [Function]
bool operator== (const Distinguished_Name_Type &d) [const Operator]
    Equality operator.
bool operator!= (const Distinguished_Name_Type &d) [const Operator]
    Inequality operator.
void clear (void) [Function]
void show (string s = "") [const Function]
```

28.4 Parameter_Type

`struct Parameter_Type` is defined in ‘`paramtp.web`’. It is currently only used for storing a file descriptor for a socket in its sole data member `int sock`. It has no member functions at present. It could therefore easily be dispensed with, but it may be useful for something in the future. For example, if more complex communication between ‘`scrinstl`’ and ‘`optdbcli`’ becomes necessary, an object of type `Parameter_Type` could be passed to a scanner/parser pair as an argument to `yylex` and `yyparse`, as this is done in other parts of the OptiNum Installer package.

28.4.1 Parameter_Type Data Members

```
int sock [Public variable]
    The socket returned by accept in the main accept loop in main.
```

29 Auxiliary Program and Shellsript Reference

29.1 Start Remote Installation

`'strtinst.web'`.

`int main (int argc, char *argv[])` [Function]
Exit status values:

- 0 Success
- 1 Checking for Globus site failed
- 2 `getpwuid_r` failed
- 3 Cannot determine user name (`pwbuf.pw_name` is empty)
- 4 Cannot determine home directory (`pwbuf.pw_dir` is empty)
- 5 Home directory path too long (has too many characters)
- 6 X.509 proxy credential not found
- 100 + return value of `handle_proxy_cred`
X.509 proxy credential invalid. (see Chapter 13 [Proxy Credential Function Reference], page 58)
- 7 Incorrect number of arguments
- 8 `mkstemp` error: Failed to create temporary files
- 9 `fchmod` error: Failed to change mode of wrapper script file
- 10 Globus site not found
- 11 Globus site disabled
- 12 Invalid data for Globus site
- 13 Failed to write wrapper script
- 14 Failed to write RSL file
- 15 Invalid host (not `optinum-srv.gwdg.de` or `pcfinston`)
- 16 Installation process failed (call to `system`)
- 17 `atexit` failed
- 18 MySQL library couldn't be initialized
- 19 `mysql_init` failed
- 20 `popen` error
- 21 `fread` error
- 22 `mysql_real_connect` failed
- 23 `mysql_select_db` failed

```
int check_globus_site (MYSQL *mysql_ptr, string globus_site,      [Function]
                      Globus_Site_Info_Type & globus_site_info)
    Defined in 'strtinst.web'.
```

- 0 Success
- 1 mysql_query, mysql_store_result or mysql_fetch_row failed
- 2 mysql_query returned no rows, i.e., Globus site not found
- 3 Invalid data in row

29.2 Kill Process

The program 'killprcss' (source code in 'killprcss.web') is called by the function `exchange_data_with_client` (see Chapter 7 [Server Connection Function Reference], page 43.)

```
int main (int argc, char **argv)                                [Function]
```

29.3 Install GPG Key Pair

The program 'instkypr' (source code in 'instkypr.web') is called by the function `sub_distribute_key_pair` (see Chapter 27 [Thread Function Reference], page 87) via 'sudo', either directly or indirectly by the wrapper script 'instkypr.sh'. The latter is needed on platforms where 'sudo' can't start a login shell without the user having to enter a password or where it doesn't allow environment variables to be passed to the shell using 'env'.

```
const unsigned int MAX_BUFFER                                    [Global constant]
    Initialized to 65536.
```

```
int main (int argc, char *argv[])                               [Function]
```

29.4 Change Owner for Installed Files

Source code in file 'instchwn.web'.

```
int main (int argc, char **argv)                                [Function]
```

29.5 Generate Passwords or Passphrases

`optpsgen` generates one or more passwords or passphrases from randomly chosen characters. Options control the type of characters used, whether whitespace may be included, and other characteristics of the passwords or passphrases. Additionally, checksums may be output, using one of several checksum functions, i.e., md5, sha1, sha224, sha256, sha384, or sha512.

The files 'optpsgen.web' and 'optpsgsb.web' contain the source code.

29.5.1 Options

Options:

'--help' Outputs a message explaining usage and listing these options and exits.

`--debug-level INTEGER`

If `INTEGER > 0`, debugging information is output. Currently, there is only one “debugging level”, i.e., the magnitude of `INTEGER` otherwise makes no difference.

`--input-filename FILENAME`

For testing or debugging. The file `FILENAME` will be used instead of `/dev/urandom` (the default) or `/dev/random` (when the `--extra-random` option is used).

`--alpha`

`--alphanum, --alnum`

`--graph`

`--printable`

These options determine what types of characters may appear in the passwords or passphrases, i.e., alphabetical, alphanumeric, “graphical”, or printable, respectively. They correspond to the C functions `isalpha`, `isalnum`, `isgraph` and `isprint`. That is, the characters allowed depend on the current locale.

`--blank`

`--space`

`--no-tabs`

Intended for use with the options `--alpha`, `--alphanum` and `--alnum`. If `--space` is used, then whitespace characters are also allowed, while if `--blank` is used, only the actual space character (ASCII 32) and the tab character (ASCII 9) are allowed. `--no-tabs` causes tabs to be suppressed. `--blank --no-tabs` therefore has the effect of allowing the actual space character, but no other whitespace characters.

These options have no effect if used together with `--printable` or `--graph`, because `--graph` is equivalent to `--printable` minus whitespace. If desired, space characters may be included in passwords or passphrases generated using `--graph` (or `--printable`) by using the `--max-block-size` option (see below).

`--length INTEGER`

Specifies the length of the password or passphrase. Default is 8 characters.

`--extra-random`

Use `/dev/random` instead of `/dev/urandom` as the source of characters. This improves the quality of the “randomness”, but may cause the program to block, if not enough entropy is present in the system. A message to this effect is issued.

`--min-block-size [INTEGER]`

`--max-block-size [INTEGER] (default 8)`

Set the minimum and/or maximum size of “blocks” of non-whitespace characters. The argument is optional. The default for the minimum block size is 4, while that for the maximum is 8. Please note that these defaults only apply *if the corresponding option is specified*. Otherwise, there is no minimum or maximum block size.

These options have no effect if used with ‘`--graph`’, because the latter excludes whitespace entirely.

‘`--no-start-space`’

‘`--no-end-space`’

Prevent whitespace from occurring at the beginning or end of the password, respectively.

‘`--delimiters [ARG]`’

The password or passphrase will be output with a “delimiter” at the beginning and end. ARG is optional. If not used, ‘`''`’ will be used on both sides. If present, it should be a character or string. If it is a character or a string containing only one character, this character will be used on both sides. If it is a string containing more than one character, the first character will be used at the beginning and the second at the end. If there are more than two characters, the remaining ones are ignored.

Delimiters can be useful if whitespace may appear at the beginning and/or end of the password or passphrase.

‘`--exclude-chars STRING`’

STRING argument required. It is a list of characters which should *not* appear in the password or passphrase.

‘`--checksum [ARG]`’

Output a checksum for the generated password or passphrase. If ARG is not present, ‘`sha1sum`’ is used to generate the checksum. Valid values for ARG are ‘`md5`’, ‘`sha1`’, ‘`sha224`’, ‘`sha256`’, ‘`sha384`’ or ‘`sha512`’.

‘`--iterations INTEGER`’

The number of passwords or passphrases (and optionally checksums) to generate.

29.5.2 Global Variables

`extern const unsigned int DEFAULT_PASSWD_LEN = 8` [Constant]
Default length of password or passphrase.

`unsigned int passwd_len` [Variable]
Set to `DEFAULT_PASSWD_LEN` at the beginning of `main`.

`int debug_level = 0` [Variable]
Set by the ‘`--debug`’ option. See Section 29.5.1 [optpsgen Options], page 93.

`extern const unsigned int ALPHA_TYPE = 1` [Constants]
`extern const unsigned int ALPHANUM_TYPE = 2`
`extern const unsigned int GRAPH_TYPE = 4`
`extern const unsigned int PRINTABLE_TYPE = 8`
`extern const unsigned int BLANK_TYPE = 16`
`extern const unsigned int SPACE_TYPE = 32`
`extern const unsigned int NO_TABS_TYPE = 64`
`extern const unsigned int NO_START_SPACE_TYPE = 128`

```
extern const unsigned int NO_END_SPACE_TYPE = 256
```

Constants for controlling what characters may appear in the password or passphrase.

```
extern const unsigned int MD5_TYPE = 1 [Constants]
```

```
extern const unsigned int SHA1_TYPE = 2
```

```
extern const unsigned int SHA224_TYPE = 3
```

```
extern const unsigned int SHA256_TYPE = 4
```

```
extern const unsigned int SHA384_TYPE = 5
```

```
extern const unsigned int SHA512_TYPE = 6
```

```
extern const int DEFAULT_MIN_BLOCK_SIZE = 4 [Constants]
```

```
extern const int DEFAULT_MAX_BLOCK_SIZE = 8
```

```
int min_block_size = 0 [Variables]
```

```
int max_block_size = 0
```

```
unsigned int block_ctr = 0
```

```
bool extra_random = false [Variable]
```

```
string delim_start [Variables]
```

```
string delim_end
```

```
string in_filename [Variable]
```

```
vector<char> exclude_char_vector [Variable]
```

```
unsigned int checksum_type = 0 [Variable]
```

```
unsigned int iterations = 1 [Variable]
```

29.5.3 Functions

```
int main (int argc, char **argv) [Function]
```

```
int handle_options (int argc, char **argv) [Function]
```

29.6 Purge Installed Files

```
int main (int argc, char ** argv) [Function]
```

Defined in ‘secure/src/prginstl.web’. **Not currently in use.** This program deletes files created by a “software provider” for a “customer” which are located below the home directory of the latter, but belong to the former, so that the “customer” cannot delete them without root privileges. Therefore, the ‘**sudoers**’ file must contain an entry for this program, which makes it possible for users to delete such files.

However, this situation no longer occurs, because the ownership of such files is changed, so that they belong to the “customer”. This program has been left in the package for the time being, because it may turn out to be useful for something.

30 Database Tables and Views

30.1 Users

```

user_id    int primary key
user_name  varchar(128) unique not null

```

30.2 User_Info

This view contains the most commonly needed information about users from the ‘Users’ and ‘Certificates’ tables. It is can be more convenient to use this view rather than to formulate the query for extracting this information from the two tables.

```

user_id    int. From Users table.
user_name  varchar(128). From Users table.
commonName varchar(64). From Certificates table.
organizationalUnitName
            varchar(64). From Certificates table.

```

30.3 Certificates

```

certificate_id
            int primary key
user_id    int references Users(user_id)
issuer_cert_id
            int references Certificates(certificate_id)
is_ca      boolean not null (Is certification authority)
is_proxy   boolean not null (Is proxy credential)
serialNumber
            integer unsigned (Fields from here on use names from the X.509
                             specifications)
organization
            varchar(64)
organizationalUnitName
            varchar(64)
commonName
            varchar(64)
countryName
            char(2)

```

```

localityName
    varchar(64)

stateOrProvinceName
    varchar(64)

Validity_notBefore
    datetime

Validity_notAfter
    datetime

```

30.4 Privileges

The ‘Privileges’ database table is used for privileges that don’t require a one-to-many, many-to-one or many-to-many association of values. For example, if ‘delegate = 1’ for user 12, then this user is permitted act as a delegate for all other users (a “one-to-all” association, so to speak). If however, user 12 should only be allowed to act as a delegate for certain users, the *association table* ‘Delegates’ must be used. See Section 30.5 [Delegates Database Table], page 99, below.

```

user_id    int primary key unique not null references Users(user_id)

superuser
    boolean not null default 0
    If ‘superuser’ = 1, user ‘user_id’ has all of the permissions controlled by the
    other columns of this table. It is therefore equivalent to having all of the other
    columns set to 1.

show_certificates
    boolean not null default 0

show_privileges
    boolean not null default 0

add_globus_site
    boolean not null default 0

delete_globus_site
    boolean not null default 0

enable_globus_site
    boolean not null default 0

edit_globus_site
    boolean not null default 0

add_globus_site_admin
    boolean not null default 0

delete_globus_site_admin
    boolean not null default 0

delegate   boolean not null default 0.

```

The user may act as a delegate for any other user. This column is accessed by the function `Scan_Parse_Parameter_Type::check_delegation`. See Section 21.2 [Scan_Parse_Parameter_Type Member Functions], page 75.

30.5 Delegates

The user referenced by `delegate_id` may act on behalf of the user referenced by `effective_user_id`. That is, the former may add, delete and edit entries, and start a remote installation (see Section 10.4 [Remote Software Installation], page 51) on behalf of the latter.

This table is accessed by the function `Scan_Parse_Parameter_Type::check_delegation`. See Section 21.2 [Scan_Parse_Parameter_Type Member Functions], page 75.

```
delegate_id
    int not null references Users(user_id)
effective_user_id
    int not null references Users(user_id)
```

30.6 Entries

```
user_id    int not null references Users(user_id)
entry_id   int primary key
package_name
    varchar(256) not null
package_version
    varchar(256)
version_ctr
    int default null
download_url
    varchar(256)
download_command
    varchar(256)
installation_script
    mediumtext
maintainer_name
    varchar(256)
maintainer_email_address
    varchar(256)
private    boolean not null default true
checked_by_admin
    boolean not null default false
created    timestamp default null
```

```

last_modified
    timestamp default null
    When a row is first inserted, 'last_modified' is set to the same value as
    'created', i.e., the current timestamp in UTC. This makes it possible to sort
    rows according to this field. To determine whether a row has been modified,
    the 'last_modified' and 'created' fields must be compared.

timezone char(3) default 'UTC'
owner    varchar(256)
encrypted
    boolean default 0

download_url_encrypted
    blob

download_command_encrypted
    blob

installation_script_encrypted
    blob

authorization
    blob

deleted boolean not null default false

```

30.7 Prerequisites

```

entry_id int not null references Entries(entry_id)
prerequisite_id
    int not null references Entries(entry_id)

```

30.8 Globus_Sites

```

globus_site_id
    int primary key

hostname varchar(128) unique not null
ip_address
    varchar(128) unique not null

enabled boolean not null default 1

```

30.9 Key_Pairs

```

key_pair_id
    int unsigned not null primary key

user_id int not null references Users(user_id)
key_name varchar(256) not null
key_pair blob

```

30.10 Public_Keys

```

user_id    int primary key references Users(user_id)
key_name   varchar(256) not null
key_id     int unsigned not null

```

30.11 Session_Data

```

session_id
    varchar(256) not null
user_id    int not null references Users(user_id)
effective_user_id
    int references Users(user_id)
user_name   varchar(128) references Users(user_name)
effective_user_name
    varchar(128) references Users(user_name)
timestamp
    timestamp default 0

```

30.12 Environment_Shellscripts

```

environment_shellscript_id
    int primary key
user_id     int not null default 0 references Users(user_id)
globus_site_id
    int not null default 0 references Globus_Sites(globus_site_id)
shellscript
    mediumtext
admin       boolean not null default 0. Can only be set if the user is an administrator of
            the Globus site.
disabled    boolean not null default 0. If 1 (true), the shellscript will not be returned by
            the server to the client, and therefore not executed, when one or more packages
            are installed on the Globus site referred to by globus_site_id.

```

30.13 Globus_Sites_Admins

For each row, the user referred to by `user_id` is an administrator of the Globus site referred to by `globus_site_id`. The use of a table separate from `Globus_Sites` makes it possible for there to be multiple administrators for a given Globus site.

```

globus_site_id
    int references Globus_Sites(globus_site_id)
admin_id     int references Users(user_id)

```


Glossary

C

CA See “Certification Authority”.

Certification Authority.
Issues X.509 certificates.

CGI See “Common Gateway Interface”. See also “FastCGI”.

Common Gateway Interface

C++ Standard Template Library

F

FastCGI An extension to CGI (“Common Gateway Interface”).

G

GNUTLS A software library for implementing communication among computers over the internet using the TLS (*Transport Layer Security*) protocol. See also TLS.

H

handshake

P

peer A term used with reference to the communication between a client and a server: Depending on the current point of view, the *peer* is the “other program”. That is, from the point of view of the client, the server is the peer, and vice versa.

proxy credential, X.509
See X.509 proxy credential.

S

Standard Template Library, C++
See C++ Standard Template Library

T

TLS Transport Layer Security. See also GNUTLS.

U

URL Uniform or Universal Resource Locator

X

X.509 Certificate

X.509 proxy credential
A file containing an X.509 certificate issued by a user (i.e., not a Certification Authority) and the *private key* for the certificate. Usually used for authorization

by application and only valid for a short time, because of the security risk involved in including the private key.

Lists of Source Files

Client/Server Source File List

Client/Server CWEB Files

- `'cmdlnopt.web'`
Code for command line options
- `'cnnectcli.web'`
GNUTLS connections for the client
- `'connect.web'`
GNUTLS connections for the server
- `'dbsrvcli.web'`
CWEB driver file. Contains no C++ code.
- `'dhprxcrt.web'`
- `'dstngnmt.web'`
- `'entries.web'`
Declaration of `class Entry_Type` and member function definitions.
- `'ex_rfc2818.web'`
GNUTLS functions. Taken from the GNUTLS distribution and adapted. (!! CHECK)
- `'ex_verify.web'`
GNUTLS functions. Taken from the GNUTLS distribution and adapted. (!! CHECK)
- `'glblfnsc.web'`
Global functions.
- `'glblvrbl.web'`
Global variables and constants.
- `'glsinftp.web'`
Declaration of `class Globus_Site_Info_Type` and member function definitions.
- `'gntlsfnc.web'`
GNUTLS functions. Taken from the GNUTLS distribution and adapted. (!! CHECK)
- `'helper.web'`
GNUTLS functions. Taken from the GNUTLS distribution and adapted. (!! CHECK)
- `'instchwn.web'`
This file contains the `main` (and only) function for the program `'instchwn'`, which changes the owner for files installed using “hidden installation”. See Section 10.2 [Hidden Software Installation], page 50.

<code>'instkypr.web'</code>	Install GPG key pair.	
<code>'instllfn.web'</code>	Installation.	
<code>'kllprcss.web'</code>		
<code>'lstnfncs.web'</code>	"Listen" functions for the server.	
<code>'oldsckt.web'</code>	Functions for using sockets. Not used.	
<code>'optdbcli.web'</code>	The main function for the client.	
<code>'optdbrsv.web'</code>	The main function for a rudimentary version of the server that can be called directly from the local host, i.e., without using a client.	
<code>'optdbsrv.web'</code>	The main function for the server.	
<code>'optwbsrv.web'</code>	The functions <code>main</code> , <code>init</code> and <code>parse_args</code> for the web application.	
<code>'parser.web'</code>	Bison code for the server's parser function (<code>yyparse</code>).	
<code>'prsrclnt.web'</code>	Bison code for the client's parser function (<code>zzparse</code>).	
<code>'prsrfncs.web'</code>	Functions for use in the parsers (<code>yyparse</code> and <code>zzparse</code>).	
<code>'prswbsrv.web'</code>	The GNU Bison input file for the web application <code>'optwbsrv'</code> .	
<code>'rspnstp.web'</code>		
<code>'scanner.web'</code>		
<code>'scnrclnt.web'</code>		
<code>'scnwbsrv.web'</code>	The Flex input file for the web application <code>'optwbsrv'</code> .	
<code>'scprpmtip.web'</code>		
<code>'spptdbfn.web'</code>	<code>Scan_Parse_Parameter_Type</code> member function definitions for use with the database.	
<code>'spptenfn.web'</code>	Definition of <code>Scan_Parse_Parameter_Type::mark_entries</code> . Section 21.2.3 [Functions for Entries], page 75.	See

`'spptenvf.web'`

`'spptgsfn.web'`

`'spptkyfn.web'`

`'spptprvl.web'`

Scan_Parse_Parameter_Type member function definitions for “privileges”. See Section 30.4 [Privileges Database Table], page 98.

`'spptsdtp.web'`

`'spptshen.web'`

`'spptwren.web'`

`'sssndttp.web'`

`'strtinst.web'`

`'thrdfncs.web'`

`'utilfncs.web'`

`'x509cert.web'`

Client/Server Shellscrip

`'checkproxy.sh'`

`'delopttf.sh'`

`'gen_proxy.sh'`

Generate proxy credentials. `'gen_proxy.sh'` uses the `'grid-proxy-init'` program from the Globus Toolkit distribution. **Please note** that if the environment variable `'GLOBUS_LOCATION'` is not set, this shellscrip sets it to `"/usr/local/globus-4.0.8"`. If this directory does not exist or does not contain `'grid-proxy-init'` (in the subdirectory `"/bin/"`) or the shellscrips `'globus-user-env.sh'` and `'globus-devel-env.sh'` (in the subdirectory `"/etc/"`), this shellscrip will fail.

`'instchwn.sh'`

Shellscrip. Wrapper scrip for calling the program `'instchwn'` (see `'instchwn.web'`, above), which changes the owner for files installed using “hidden installation”. See [Client/Server (optdbcli/optdbsrv) CWEB Files], page 104.

`'instkypr.sh'`

Install GPG key pair. Wrapper scrip for the program `'instkypr'` (see above).

`'instoipk.sh'`

GPG Public Key for `'optinum_installer'` and set trust level to “full”.

`'kill_optdbsrv.sh'`

`'kyprinst.sh'`

`'prpencfs.sh'`

`'clprpenc.sh'`

`'setupkyp.sh'`

Create GPG key pair using Distinguished Name from certificate. Sign the public key belonging to `'optinum_installer (scrinstl)'` with it. The filename of the certificate may be passed as the first (currently only) argument to this shell scrip. Otherwise, the default is `"/usercert.pem"`, if it exists. If it does not exist, this shellscrip issues an error message and exits with exit status 1.

`'strtinst.sh'`

Wrapper script for calling `'strtinst'`. See Section 29.1 [Start Remote Installation], page 92.

Hidden Installation Source File List

`'cnnctfnc.web'`
`'dstngnmt.web'`
`'glblvrbl.web'`
`'hdprxcrt.web'`
`'paramtp.web'`
`'prginstl.web'`
`'rdmstrfs.web'`
`'scrclnt.web'`
`'scrinstl.web'`
`'scrsrvr.web'`
`'secure.web'`
`'utilfnsc.web'`

Hidden Installation CWEB Files

`'cnnctfnc.web'`
`'dstngnmt.web'`
`'glblvrbl.web'`
`'hdprxcrt.web'`
`'paramtp.web'`
`'prginstl.web'`
`'rdmstrfs.web'`
`'scrclnt.web'`
`'scrinstl.web'`
`'scrsrvr.web'`
`'secure.web'`
`'utilfnsc.web'`

Hidden Installation Shellscripts

`'restart_scrinstl.sh'`
`'start_scrinstl.sh'`
`'setup_logs.sh'`
`'gnkeybtc.sh'`

Web Application Source File List

`'auxfltp.web'`

Declaration of `'struct Auxiliary_File_Type'`.

`'ckietype.web'`
`'dbfuncs.web'`
`'optwbsrv.web'`
`'optwbsub.web'`
`'optwtyps.web'`
`'prswbsrv.web'`
`'scnwbsrv.web'`
`'test_js.web'`
`'thrdfncs.web'`
`'utilfncs.web'`

Application CWEB Files

`'auxfltp.web'`
`'ckietype.web'`
`'dbfuncs.web'`
`'optwbsrv.web'`
`'optwbsub.web'`
`'optwebap.web'`
`'optwtyps.web'`
`'prswbsrv.web'`
`'scnwbsrv.web'`
`'thrdfncs.web'`
`'utilfncs.web'`

Application Shellscripts

`'kill_optwbsub.sh'`
`'restart_optwbsub.sh'`
`'setup_logs.sh'`
`'start_optwbsub.sh'`
`'prinscrp.sh'`
`'prspstdt.sh'`
`'rnstlsrv.sh'`

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released

under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any,

be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their

titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.3  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover  
Texts. A copy of the license is included in the section entitled “GNU  
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with  
the Front-Cover Texts being list, and with the Back-Cover Texts  
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

GNU General Public License

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users’ Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a. The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b. The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c. You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d. If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c. Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d. Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e. Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source.

The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a. Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b. Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c. Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

- d. Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e. Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f. Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance.

However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so

available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others’ Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.
Copyright (C) year name of author
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or (at
your option) any later version.
```

```
This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program. If not, see http://www.gnu.org/licenses/.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
program Copyright (C) year name of author
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.
```

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, your program’s commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Variable Index

(
(Globus_Site_Info_Type)	
append_ld_library_path	82
(Globus_Site_Info_Type) append_path	82
(Globus_Site_Info_Type) enabled	82
(Globus_Site_Info_Type) globus_site_id	82
(Globus_Site_Info_Type) hostname	82
(Globus_Site_Info_Type)	
interactive_node_ip_address	82
(Globus_Site_Info_Type)	
interactive_node_name	82
(Globus_Site_Info_Type) ip_address	82
(Globus_Site_Info_Type) optdbcli_path	82
(Globus_Site_Info_Type) replace	82
A	
ALPHA_TYPE = 1	95
ALPHANUM_TYPE = 2	95
anon_client_cred_ptr	74
anon_client_cred_ptr	
(Scan_Parse_Parameter_Type)	73
anon_server_cred_ptr	74
anon_server_cred_ptr	
(Scan_Parse_Parameter_Type)	73
append_ld_library_path	82
append_path	82
authenticated_connection = false	40
authorization	64
authorization (Entry_Type)	64
AUTHORIZATION_TYPE (Response_Type)	70
AUTHORIZATION_TYPE = 21	70
B	
BLANK_TYPE = 16	95
block_ctr = 0	96
C	
ca_filenames	13, 40
ca_list	74
ca_list (Scan_Parse_Parameter_Type)	73
ca_list_size	74
ca_list_size (Scan_Parse_Parameter_Type)	
.....	73
cache_db	52
cache_db_ptr	52
cerr_mutex	39, 89
cert_cred_ptr	74
cert_cred_ptr (Scan_Parse_Parameter_Type)	
.....	73
cert_filenames	13, 40
certificate_user	69
certificate_user (Response_Type)	69
certificates_all	69
certificates_all (Response_Type)	69
certtool_path	58
checksum_type = 0	96
client_finished	74
client_finished (Scan_Parse_Parameter_Type)	
.....	73
command	69
command (Response_Type)	69
COMMAND_ONLY_TYPE (Response_Type)	70
COMMAND_ONLY_TYPE = 3	70
commonName	56, 66, 91
commonName (Distinguished_Name_Type)	66
commonName (Distinguished_Name_Type)	
(scrinstl)	91
commonName (X509_Cert_Type)	56
countryName	56, 66, 91
countryName (Distinguished_Name_Type)	66
countryName (Distinguished_Name_Type)	
(scrinstl)	91
countryName (X509_Cert_Type)	56
create_links_value = 0	40
created	64
created (Entry_Type)	64
crl_filenames	13, 41
crl_list	52
crl_list_size	52
D	
date_time_str	73
date_time_str (Scan_Parse_Parameter_Type)	
.....	73
debug_level = 0	95
DEBUG_SCANNER	47, 59, 61
DEFAULT_CA_FILENAME	38
DEFAULT_CERT_FILENAME	40
DEFAULT_CRL_FILENAME = ""	38
DEFAULT_ERR_LOG_FILENAME =	
"/tmp/optdbcli_error.log"	38
DEFAULT_KEY_FILENAME	40
DEFAULT_LOG_FILENAME = "/tmp/optdbcli.log"	
.....	38
DEFAULT_MAX_BLOCK_SIZE = 8	96
DEFAULT_MIN_BLOCK_SIZE = 4	96
DEFAULT_PASSWD_LEN = 8	95
DEFAULT_PORT_NUM_AUTH = 5556	38
DEFAULT_PORT_NUM_NON_AUTH = 5557	38
DEFAULT_PORT_STR_AUTH = "5556"	38
DEFAULT_PORT_STR_NON_AUTH = "5557"	38
delete_files	89
deleted	64
deleted (Entry_Type)	64
delim_end	96

delim_start	96	DOWNLOAD_URL_ENCRYPTED_TYPE = 19	70
DH_BITS = 1024	52	download_url_string	74
distinguished_name	39	download_url_string	
distinguished_name_str	39, 69	(Scan_Parse_Parameter_Type)	73
distinguished_name_str (Response_Type)	69	DOWNLOAD_URL_TYPE (Response_Type)	70
Distinguished_Name_Type::commonName	66	DOWNLOAD_URL_TYPE = 17	70
Distinguished_Name_Type::commonName			
(scrinstl)	91		
Distinguished_Name_Type::countryName	66	E	
Distinguished_Name_Type::countryName		effective_user_dn	74
(scrinstl)	91	effective_user_dn	
Distinguished_Name_Type::localityName	66	(Scan_Parse_Parameter_Type)	73
Distinguished_Name_Type::localityName		effective_user_id	74
(scrinstl)	91	effective_user_id	
Distinguished_Name_Type::organization	66	(Scan_Parse_Parameter_Type)	73
Distinguished_Name_Type::organization		effective_user_name	74
(scrinstl)	91	effective_user_name	
Distinguished_Name_		(Scan_Parse_Parameter_Type)	73
Type::organizationalUnitName	66	emendations (Scan_Parse_Parameter_Type) ...	73
Distinguished_Name_		enabled	82
Type::organizationalUnitName (scrinstl)		encrypted_flag	64
.....	91	encrypted_flag (Entry_Type)	64
Distinguished_Name_		entries (Scan_Parse_Parameter_Type)	73
Type::stateOrProvinceName	66	entries_all	69
Distinguished_Name_		entries_all (Response_Type)	69
Type::stateOrProvinceName (scrinstl)		entries_ctr	69
.....	91	entries_ctr (Response_Type)	69
Distinguished_Name_Type::user_id	66	entries_public	69
Distinguished_Name_Type::user_id (scrinstl)		entries_public (Response_Type)	69
.....	91	entries_single	69
Distinguished_Name_Type::user_name	66	entries_single (Response_Type)	69
Distinguished_Name_Type::user_name		entries_user	69
(scrinstl)	91	entries_user (Response_Type)	69
distinguished_name_vector	39	entries_with_dependents	69
DISTRIBUTE_KEY_PAIR_TYPE (Response_Type)		entries_with_dependents (Response_Type) ...	69
.....	70	entries_with_prerequisites	69
DISTRIBUTE_KEY_PAIR_TYPE = 28	70	entries_with_prerequisites (Response_Type)	
dn_fields	39	69
dn_username_map	39	entry_id	64, 69
download_command	64	entry_id (Entry_Type)	64
download_command (Entry_Type)	64	entry_id (Response_Type)	69
download_command_encrypted	64	entry_name	69
download_command_encrypted (Entry_Type) ...	64	entry_name (Response_Type)	69
DOWNLOAD_COMMAND_ENCRYPTED_TYPE		entry_string_vector	74
(Response_Type)	70	entry_string_vector	
DOWNLOAD_COMMAND_ENCRYPTED_TYPE = 20	70	(Scan_Parse_Parameter_Type)	73
download_command_string	74	Entry_Type::authorization	64
download_command_string		Entry_Type::created	64
(Scan_Parse_Parameter_Type)	73	Entry_Type::deleted	64
DOWNLOAD_COMMAND_TYPE (Response_Type)	70	Entry_Type::download_command	64
DOWNLOAD_COMMAND_TYPE = 18	70	Entry_Type::download_command_encrypted ...	64
download_fail_continue	40	Entry_Type::download_url	64
download_url	64	Entry_Type::download_url_encrypted	64
download_url (Entry_Type)	64	Entry_Type::encrypted_flag	64
download_url_encrypted	64	Entry_Type::entry_id	64
download_url_encrypted (Entry_Type)	64	Entry_Type::globus_site	64
DOWNLOAD_URL_ENCRYPTED_TYPE (Response_Type)		Entry_Type::install_or_reinstall_flag	64
.....	70	Entry_Type::install_order	64

Entry_Type::installation_script 64
 Entry_Type::installation_script_encrypted
 64
 Entry_Type::last_modified 64
 Entry_Type::maintainer_email_address 64
 Entry_Type::maintainer_name 64
 Entry_Type::owner 64
 Entry_Type::owner_key_id 64
 Entry_Type::package_name 64
 Entry_Type::package_version 64
 Entry_Type::param 64
 Entry_Type::prerequisites 64
 Entry_Type::private_flag 64
 Entry_Type::remote_install_pid 64
 Entry_Type::remote_install_retval 64
 Entry_Type::remote_install_thread_id 64
 Entry_Type::remote_install_timeout 64
 Entry_Type::remote_install_timespec 64
 Entry_Type::stderr_filename 64
 Entry_Type::stdout_filename 64
 Entry_Type> 74
 environ 47
 environment_admin_flag 75
 environment_admin_flag
 (Scan_Parse_Parameter_Type) 73
 environment_globus_site 75
 environment_globus_site
 (Scan_Parse_Parameter_Type) 73
 environment_shellscript_filename_vector .. 75
 environment_shellscript_filename_vector
 (Scan_Parse_Parameter_Type) 73
 environment_strm 75
 environment_strm (Scan_Parse_Parameter_Type)
 73
 err_log_filename 40
 err_log_strm 39
 err_log_strm_mutex 39
 errors_occurred = 0 40
 exclude_char_vector 96
 extra_random = false 96

F

fd 73
 fd (Scan_Parse_Parameter_Type) 73
 fetch = false 40
 FETCH_AUTHORIZATION_TYPE (Response_Type)
 70
 FETCH_AUTHORIZATION_TYPE = 6 70
 FETCH_DOWNLOAD_INFO_TYPE (Response_Type)
 70
 FETCH_DOWNLOAD_INFO_TYPE = 5 70
 FETCH_ENVIRONMENT_SHELLSCRIPT_TYPE
 (Response_Type) 70
 FETCH_ENVIRONMENT_SHELLSCRIPT_TYPE = 8 70
 FETCH_INSTALLATION_SCRIPT_TYPE
 (Response_Type) 70
 FETCH_INSTALLATION_SCRIPT_TYPE = 4 70

FETCH_KEY_PAIR_TYPE (Response_Type) 70
 FETCH_KEY_PAIR_TYPE = 27 70
 FETCH_PREREQUISITES_TYPE (Response_Type)
 70
 FETCH_PREREQUISITES_TYPE = 7 70
 fifo_vector 40
 FTP_PROTOCOL = 1 38

G

global_variable_mutex 39
 globus_site 40, 64, 70, 84
 globus_site (Entry_Type) 64
 globus_site (Response_Type) 69
 globus_site_id 82
 Globus_Site_Info_Type::append_ld_library_
 path 82
 Globus_Site_Info_Type::append_path 82
 Globus_Site_Info_Type::enabled 82
 Globus_Site_Info_Type::globus_site_id 82
 Globus_Site_Info_Type::hostname 82
 Globus_Site_Info_Type::interactive_node_ip_
 address 82
 Globus_Site_Info_Type::interactive_node_
 name 82
 Globus_Site_Info_Type::ip_address 82
 Globus_Site_Info_Type::optdbcli_path 82
 Globus_Site_Info_Type::replace 82
 globus_site_map 40
 globus_site_options 74
 globus_site_options
 (Scan_Parse_Parameter_Type) 73
 globus_site_set 70, 74
 globus_site_set (Response_Type) 69
 globus_site_set (Scan_Parse_Parameter_Type)
 73
 gpg_homedir 40
 GPG_Key_Pair_Type::param 84
 GRAPH_TYPE = 4 95
 grid_mapfile_filename 39, 89

H

hostname 82
 HTTP_PROTOCOL = 2 38
 HTTPS_PROTOCOL = 3 38

I

IN_BUFFER_SIZE = 1048576 47
 in_filename 73, 96
 in_filename (Scan_Parse_Parameter_Type) ... 73
 input_filename 39
 install = false 40
 install_directory 40
 install_fail_continue 40
 install_or_reinstall_flag 64
 install_or_reinstall_flag (Entry_Type) 64

install_order 64, 69
 install_order (Response_Type) 69
 install_vector (Scan_Parse_Parameter_Type)
 73
 installation_script 64
 installation_script (Entry_Type) 64
 installation_script_encrypted 64
 installation_script_encrypted (Entry_Type)
 64
 INSTALLATION_SCRIPT_ENCRYPTED_TYPE
 (Response_Type) 70
 INSTALLATION_SCRIPT_ENCRYPTED_TYPE = 2 70
 INSTALLATION_SCRIPT_TYPE (Response_Type)
 70
 INSTALLATION_SCRIPT_TYPE = 1 70
 int 64
 int install_order (Entry_Type) 64
 interactive_node_ip_address 82
 interactive_node_name 82
 ip_address 82
 is_client 39
 is_server 39
 issuer_cert 56
 issuer_cert (X509_Cert_Type) 56
 iterations = 1 96

K

key_filenames 13, 40
 key_id 74
 key_name 74
 key_name (Scan_Parse_Parameter_Type) 73
 key_pair_str 74
 key_pair_str[65536]
 (Scan_Parse_Parameter_Type) 73
 KEY_PAIR_TYPE (Response_Type) 70
 KEY_PAIR_TYPE = 26 70

L

last_modified 64
 last_modified (Entry_Type) 64
 link_directory 40
 local_filename 69
 local_filename (Response_Type) 69
 local_filename_1 69
 local_filename_1 (Response_Type) 69
 local_stderr_filename 70
 local_stderr_filename (Response_Type) 69
 local_stdout_filename 70
 local_stdout_filename (Response_Type) 69
 localityName 56, 66, 91
 localityName (Distinguished_Name_Type) 66
 localityName (Distinguished_Name_Type)
 (scrinstl) 91
 localityName (X509_Cert_Type) 56
 log_filename 40
 log_strm 39

log_strm_mutex 39
 long key_id (Scan_Parse_Parameter_Type) ... 73

M

maintainer_email_address 64
 maintainer_email_address (Entry_Type) 64
 maintainer_name 64
 maintainer_name (Entry_Type) 64
 max_block_size = 0 96
 MAX_BUFFER 73, 93
 MAX_BUFFER (Scan_Parse_Parameter_Type) 73
 MAX_FILENAME = 128 38
 MAX_KEY_PAIR_SIZE 74
 MAX_KEY_PAIR_SIZE
 (Scan_Parse_Parameter_Type) 73
 MAX_PROTOCOL = 4 38
 MAX_SESSION_DATA_SIZE = 512 52
 MAX_SESSION_ID_SIZE = 32 52
 MAX_USER_ID = 128 38
 MD5_TYPE = 1 96
 min_block_size = 0 96
 msg_strm 73
 msg_strm (Scan_Parse_Parameter_Type) 73
 mysql_password_encrypted 52
 mysql_ptr 74
 mysql_ptr (Scan_Parse_Parameter_Type) 73

N

NO_END_SPACE_TYPE = 256 95
 NO_START_SPACE_TYPE = 128 95
 NO_TABS_TYPE = 64 95
 NULL_PROTOCOL = 0 38
 NULL_RESPONSE_TYPE (Response_Type) 70
 NULL_RESPONSE_TYPE = 0 70

O

optdbcli_path 82
 optimum_installer_gpg_pubkey_fingerprint
 40
 organization 56, 66, 91
 organization (Distinguished_Name_Type) 66
 organization (Distinguished_Name_Type)
 (scrinstl) 91
 organization (X509_Cert_Type) 56
 organizationalUnitName 56, 66, 91
 organizationalUnitName
 (Distinguished_Name_Type) 66
 organizationalUnitName
 (Distinguished_Name_Type) (scrinstl)
 91
 organizationalUnitName (X509_Cert_Type) ... 56
 out_filename 73
 out_filename (Scan_Parse_Parameter_Type)
 73
 out_strm 73

out_strm (Scan_Parse_Parameter_Type) 73
 output_file_strm 39
 output_filename 39
 output_format 69
 output_format (Response_Type) 69
 OUTPUT_FORMAT_HTML (Response_Type) 70
 OUTPUT_FORMAT_HTML = 1 70
 OUTPUT_FORMAT_JAVASCRIPT (Response_Type)
 70
 OUTPUT_FORMAT_JAVASCRIPT = 3 70
 OUTPUT_FORMAT_PLAIN_TEXT (Response_Type)
 70
 OUTPUT_FORMAT_PLAIN_TEXT = 0 70
 OUTPUT_FORMAT_TEX (Response_Type) 70
 OUTPUT_FORMAT_TEX = 2 70
 owner 64
 owner (Entry_Type) 64
 owner_certificate = "" 39
 owner_key_id (Entry_Type) 64

P

package_name 64
 package_name (Entry_Type) 64
 package_names 40
 package_version 64
 package_version (Entry_Type) 64
 package_versions 40
 param 64, 84
 param (Entry_Type) 64
 param (GPG_Key_Pair_Type) 84
 Parameter_Type::sock 91
 PARSER_DEBUG 73
 PARSER_DEBUG (Scan_Parse_Parameter_Type)
 73
 passwd_len 95
 port_num_auth 39
 port_num_non_auth 39
 port_str_auth 39
 port_str_non_auth 39
 prerequisite_list 74
 prerequisite_list
 (Scan_Parse_Parameter_Type) 73
 prerequisite_string_vector 74
 prerequisite_string_vector
 (Scan_Parse_Parameter_Type) 73
 prerequisites 64
 prerequisites_reinstall_value = 0 40
 PRINTABLE_TYPE = 8 95
 private_flag 64
 private_flag (Entry_Type) 64
 privileges_all 70
 privileges_all (Response_Type) 69
 privileges_user 69
 privileges_user (Response_Type) 69
 process_owner_dir[128] 89
 proxy_cred_filename 40, 74

proxy_cred_filename
 (Scan_Parse_Parameter_Type) 73
 proxy_cred_verified 89
 proxy_cred_verified = false 40
 PROXY_CREDENTIAL_TYPE (Response_Type) 70
 PROXY_CREDENTIAL_TYPE = 22 70
 ptid 84
 PUBLIC_KEY_TYPE (Response_Type) 70
 PUBLIC_KEY_TYPE = 25 70

R

reinstall_flag = false 40
 remote_filename 69
 remote_filename (Response_Type) 69
 remote_install_pid 64
 remote_install_pid (Entry_Type) 64
 remote_install_retval 64
 remote_install_retval (Entry_Type) 64
 remote_install_thread_id 64
 remote_install_thread_id (Entry_Type) 64
 remote_install_timeout 64
 remote_install_timeout (Entry_Type) 64
 remote_install_timespec 64
 remote_install_timespec (Entry_Type) 64
 REMOTE_INSTALL_TYPE (Response_Type) 70
 REMOTE_INSTALL_TYPE = 23 70
 remote_ip_address 74
 remote_ip_address
 (Scan_Parse_Parameter_Type) 73
 remote_port 74
 remote_port (Scan_Parse_Parameter_Type) ... 73
 remote_stderr_filename 70
 remote_stderr_filename (Response_Type) ... 69
 remote_stdout_filename 70
 remote_stdout_filename (Response_Type) ... 69
 replace 82
 resolve_dependencies = true 40
 response_deque 74
 response_deque (Scan_Parse_Parameter_Type)
 73
 response_deque_mutex 74
 response_deque_mutex
 (Scan_Parse_Parameter_Type) 73
 Response_Type::AUTHORIZATION_TYPE 70
 Response_Type::certificate_user 69
 Response_Type::certificates_all 69
 Response_Type::command 69
 Response_Type::COMMAND_ONLY_TYPE 70
 Response_Type::distinguished_name_str 69
 Response_Type::DISTRIBUTE_KEY_PAIR_TYPE .. 70
 Response_Type::DOWNLOAD_COMMAND_ENCRYPTED_
 TYPE 70
 Response_Type::DOWNLOAD_COMMAND_TYPE 70
 Response_Type::DOWNLOAD_URL_ENCRYPTED_TYPE
 70
 Response_Type::DOWNLOAD_URL_TYPE 70
 Response_Type::entries_all 69

Response_Type::entries_ctr	69
Response_Type::entries_public	69
Response_Type::entries_single	69
Response_Type::entries_user	69
Response_Type::entries_with_dependents ...	69
Response_Type::entries_with_prerequisites	69
Response_Type::entry_id	69
Response_Type::entry_name	69
Response_Type::FETCH_AUTHORIZATION_TYPE ..	70
Response_Type::FETCH_DOWNLOAD_INFO_TYPE ..	70
Response_Type::FETCH_ENVIRONMENT_ SHELLSCRIPT_TYPE	70
Response_Type::FETCH_INSTALLATION_SCRIPT_ TYPE	70
Response_Type::FETCH_KEY_PAIR_TYPE	70
Response_Type::FETCH_PREREQUISITES_TYPE ..	70
Response_Type::globus_site	69
Response_Type::globus_site_set	69
Response_Type::install_order	69
Response_Type::INSTALLATION_SCRIPT_ ENCRYPTED_TYPE	70
Response_Type::INSTALLATION_SCRIPT_TYPE ..	70
Response_Type::KEY_PAIR_TYPE	70
Response_Type::local_filename	69
Response_Type::local_filename_1	69
Response_Type::local_stderr_filename	69
Response_Type::local_stdout_filename	69
Response_Type::NULL_RESPONSE_TYPE	70
Response_Type::output_format	69
Response_Type::OUTPUT_FORMAT_HTML	70
Response_Type::OUTPUT_FORMAT_JAVASCRIPT ..	70
Response_Type::OUTPUT_FORMAT_PLAIN_TEXT ..	70
Response_Type::OUTPUT_FORMAT_TEX	70
Response_Type::privileges_all	69
Response_Type::privileges_user	69
Response_Type::PROXY_CREDENTIAL_TYPE	70
Response_Type::PUBLIC_KEY_TYPE	70
Response_Type::remote_filename	69
Response_Type::REMOTE_INSTALL_TYPE	70
Response_Type::remote_stderr_filename	69
Response_Type::remote_stdout_filename	69
Response_Type::RETURN_ENVIRONMENT_ SHELLSCRIPT_TYPE	70
Response_Type::SECRET_KEY_TYPE	70
Response_Type::SEND_ENVIRONMENT_ SHELLSCRIPT_TYPE	70
Response_Type::SHOW_CERTIFICATES_TYPE	70
Response_Type::SHOW_ENTRIES_TYPE	70
Response_Type::show_environment_ shellscrip_options	69
Response_Type::SHOW_ENVIRONMENT_ SHELLSCRIPTS_TYPE	70
Response_Type::SHOW_GLOBUS_SITES_TYPE	70
Response_Type::show_level_map	69
Response_Type::SHOW_PRIVILEGES_TYPE	70
Response_Type::type	69

```

Response_Type::VALIDATE_DISTINGUISHED_NAME_
    TYPE..... 70
Response_Type::version..... 69
RETURN_ENVIRONMENT_SHELLSCRIPT_TYPE
    (Response_Type) ..... 70
RETURN_ENVIRONMENT_SHELLSCRIPT_TYPE = 10.. 70
return_value ..... 84

```

S

```

SA = struct sockaddr ..... 52
save_temp_files = false ..... 39
Scan_Parse_Parameter_Type::anon_client_
    cred_ptr ..... 73
Scan_Parse_Parameter_Type::anon_server_
    cred_ptr ..... 73
Scan_Parse_Parameter_Type::ca_list ..... 73
Scan_Parse_Parameter_Type::ca_list_size .. 73
Scan_Parse_Parameter_Type::cert_cred_ptr
    ..... 73
Scan_Parse_Parameter_Type::client_finished
    ..... 73
Scan_Parse_Parameter_Type::date_time_str
    ..... 73
Scan_Parse_Parameter_Type::download_
    command_string ..... 73
Scan_Parse_Parameter_Type::download_url_
    string ..... 73
Scan_Parse_Parameter_Type::effective_user_
    dn ..... 73
Scan_Parse_Parameter_Type::effective_user_
    id ..... 73
Scan_Parse_Parameter_Type::effective_user_
    name ..... 73
Scan_Parse_Parameter_Type::emendations ... 73
Scan_Parse_Parameter_Type::entries ..... 73
Scan_Parse_Parameter_Type::entry_string_
    vector ..... 73
Scan_Parse_Parameter_Type::environment_
    admin_flag ..... 73
Scan_Parse_Parameter_Type::environment_
    globus_site ..... 73
Scan_Parse_Parameter_Type::environment_
    shellscrip_filename_vector ..... 73
Scan_Parse_Parameter_Type::environment_strm
    ..... 73
Scan_Parse_Parameter_Type::fd ..... 73
Scan_Parse_Parameter_Type::globus_site_
    options ..... 73
Scan_Parse_Parameter_Type::globus_site_set
    ..... 73
Scan_Parse_Parameter_Type::in_filename ... 73
Scan_Parse_Parameter_Type::install_vector
    ..... 73
Scan_Parse_Parameter_Type::key_name ..... 73
Scan_Parse_Parameter_Type::key_pair_
    str[65536] ..... 73
Scan_Parse_Parameter_Type::long key_id ... 73

```


Scan_Parse_Parameter_Type::MAX_BUFFER.....	73	session (Scan_Parse_Parameter_Type)	73
Scan_Parse_Parameter_Type::MAX_KEY_PAIR_		session_data	74
SIZE.....	73	session_data (Scan_Parse_Parameter_Type:)	
Scan_Parse_Parameter_Type::msg_strm.....	73	73
Scan_Parse_Parameter_Type::mysql_ptr.....	73	session_data[MAX_SESSION_DATA_SIZE]	52
Scan_Parse_Parameter_Type::out_filename..	73	session_data_mutex	39
Scan_Parse_Parameter_Type::out_strm.....	73	session_data_size.....	52
Scan_Parse_Parameter_Type::PARSER_DEBUG..	73	Session_Data_Type::session_id.....	85
Scan_Parse_Parameter_Type::prerequisite_		session_id.....	41, 74, 85
list.....	73	session_id (Scan_Parse_Parameter_Type)....	73
Scan_Parse_Parameter_Type::prerequisite_		session_id (Session_Data_Type)	85
string_vector.....	73	session_id[MAX_SESSION_ID_SIZE]	52
Scan_Parse_Parameter_Type::proxy_cred_		session_id_size.....	52
filename	73	SHA1_TYPE = 2.....	96
Scan_Parse_Parameter_Type::remote_ip_		SHA224_TYPE = 3.....	96
address.....	73	SHA256_TYPE = 4.....	96
Scan_Parse_Parameter_Type::remote_port ...	73	SHA384_TYPE = 5.....	96
Scan_Parse_Parameter_Type::response_deque		SHA512_TYPE = 6.....	96
.....	73	SHOW_CERTIFICATES_TYPE (Response_Type)....	70
Scan_Parse_Parameter_Type::response_deque_		SHOW_CERTIFICATES_TYPE = 13.....	70
mutex.....	73	SHOW_ENTRIES_TYPE (Response_Type).....	70
Scan_Parse_Parameter_Type::server_finished		SHOW_ENTRIES_TYPE = 12.....	70
.....	73	show_environment_shellscript_globus_site	
Scan_Parse_Parameter_Type::session.....	73	75
Scan_Parse_Parameter_Type::session_data..	73	show_environment_shellscript_globus_site	
Scan_Parse_Parameter_Type::session_id....	73	(Scan_Parse_Parameter_Type).....	73
Scan_Parse_Parameter_Type::show_		show_environment_shellscript_globus_site_	
environment_shellscript_globus_site ..	73	set.....	74
Scan_Parse_Parameter_Type::show_		show_environment_shellscript_globus_site_	
environment_shellscript_globus_site_set		set (Scan_Parse_Parameter_Type)	73
.....	73	show_environment_shellscript_options..	70, 75
Scan_Parse_Parameter_Type::show_		show_environment_shellscript_options	
environment_shellscript_options.....	73	(Response_Type)	69
Scan_Parse_Parameter_Type::show_level_map		show_environment_shellscript_options	
.....	73	(Scan_Parse_Parameter_Type).....	73
Scan_Parse_Parameter_Type::show_options..	73	SHOW_ENVIRONMENT_SHELLSCRIPTS_TYPE	
Scan_Parse_Parameter_Type::start_condition		(Response_Type)	70
.....	73	SHOW_ENVIRONMENT_SHELLSCRIPTS_TYPE = 14... 70	
Scan_Parse_Parameter_Type::thread_ctr.....	73	SHOW_GLOBUS_SITES_TYPE (Response_Type)....	70
Scan_Parse_Parameter_Type::thread_id.....	73	SHOW_GLOBUS_SITES_TYPE = 15.....	70
Scan_Parse_Parameter_Type::up.....	73	show_level_map	70, 74
Scan_Parse_Parameter_Type::user_dn.....	73	show_level_map (Response_Type)	69
Scan_Parse_Parameter_Type::user_id.....	73	show_level_map (Scan_Parse_Parameter_Type)	
Scan_Parse_Parameter_Type::user_name.....	73	73
scanner_trace = false.....	40	show_options	74
SECRET_KEY_TYPE (Response_Type).....	70	show_options (Scan_Parse_Parameter_Type)	
SECRET_KEY_TYPE = 24.....	70	73
SEND_ENVIRONMENT_SHELLSCRIPT_TYPE		SHOW_PRIVILEGES_TYPE (Response_Type)	70
(Response_Type)	70	SHOW_PRIVILEGES_TYPE = 16.....	70
SEND_ENVIRONMENT_SHELLSCRIPTS_TYPE = 9 ...	70	SHOW_PUBLIC_ENTRIES = 2	38
SEND_ENVIRONMENT_TYPE (Response_Type)....	70	SHOW_USER_ENTRIES = 1.....	38
serialNumber	56	sleep_value = 0.....	39
serialNumber (X509_Cert_Type)	56	sock	91
server_finished.....	74	sock (Parameter_Type)	91
server_finished (Scan_Parse_Parameter_Type)		SOCKET_ERR (err,s) if(err== -1)	
.....	73	perror(s);pthread_exit(0);.....	52
server_ip_address.....	39	SPACE_TYPE = 32.....	95
session	74	srp_dh_group2048[]	52

start_condition..... 73
 start_condition (Scan_Parse_Parameter_Type)
 73
 stateOrProvinceName..... 56, 66, 91
 stateOrProvinceName
 (Distinguished_Name_Type)..... 66
 stateOrProvinceName
 (Distinguished_Name_Type) (scrinstl)
 91
 stateOrProvinceName (X509_Cert_Type)..... 56
 stderr_filename..... 64
 stderr_filename (Entry_Type)..... 64
 stdout_filename..... 64
 stdout_filename (Entry_Type)..... 64
 sub_thread_ctr..... 84

T

temp_file_vector..... 40
 thread_ctr..... 39, 73
 thread_ctr (Scan_Parse_Parameter_Type).... 73
 thread_ctr_mutex..... 39
 thread_id..... 73
 thread_id (Scan_Parse_Parameter_Type).... 73
 TLS_SESSION_CACHE = 50..... 52
 trace_value = 0..... 40
 transfer_proxy_cred = false..... 40
 transfer_proxy_cred_filename..... 40
 type..... 69
 type (Response_Type)..... 69

U

up..... 73
 up (Scan_Parse_Parameter_Type)..... 73
 user_dn..... 74
 user_dn (Scan_Parse_Parameter_Type)..... 73
 user_id..... 56, 66, 74, 91
 user_id (Distinguished_Name_Type)..... 66
 user_id (Distinguished_Name_Type) (scrinstl)
 91

user_id (Scan_Parse_Parameter_Type)..... 73
 user_id (X509_Cert_Type)..... 56
 user_name..... 56, 66, 74, 91
 user_name (Distinguished_Name_Type)..... 66
 user_name (Distinguished_Name_Type)
 (scrinstl)..... 91
 user_name (Scan_Parse_Parameter_Type).... 73
 user_name (X509_Cert_Type)..... 56

V

VALIDATE_DISTINGUISHED_NAME_TYPE
 (Response_Type)..... 70
 VALIDATE_DISTINGUISHED_NAME_TYPE = 11..... 70
 Validity_notAfter..... 56
 Validity_notAfter (X509_Cert_Type)..... 56
 Validity_notBefore..... 56
 Validity_notBefore (X509_Cert_Type)..... 56
 vector<Entry_Type> prerequisites
 (Entry_Type)..... 64
 version..... 69
 version (Response_Type)..... 69

W

warnings_occurred = 0..... 40

X

X509_Cert_Type::commonName..... 56
 X509_Cert_Type::countryName..... 56
 X509_Cert_Type::issuer_cert..... 56
 X509_Cert_Type::localityName..... 56
 X509_Cert_Type::organization..... 56
 X509_Cert_Type::organizationalUnitName... 56
 X509_Cert_Type::serialNumber..... 56
 X509_Cert_Type::stateOrProvinceName..... 56
 X509_Cert_Type::user_id..... 56
 X509_Cert_Type::user_name..... 56
 X509_Cert_Type::Validity_notAfter..... 56
 X509_Cert_Type::Validity_notBefore..... 56

Data Type Index

C

CACHE 52
 class Distinguished_Name_Type 56, 66
 class Entry_Type 63, 72
 class equal_package_name_globus_site 65
 class equal_package_name_globus_site 63
 class equal_package_name_version_globus_site
 63, 65
 class Globus_Site_Info_Type 82
 class GPG_Key_Pair_Type 72, 84
 class less_package_name 63, 65
 class less_package_name_globus_site_version
 63, 65
 class less_package_name_version 63, 65
 class less_package_version 63, 65
 class less_timeout 63, 65
 class Response_Type 68
 class Scan_Parse_Parameter_Type 56, 63, 66,
 68, 72, 84
 class Session_Data_Type 85
 class X509_Cert_Type 56, 72

D

deque 68
 Distinguished_Name_Type, class 56, 66
 Distinguished_Name_Type, struct 90

E

Entry_Type, class 63, 72
 equal_package_name_globus_site, class 63,
 65
 equal_package_name_version_globus_site,
 class 63, 65

G

Globus_Site_Info_Type, class 82
 GPG_Key_Pair_Type, class 72, 84

L

less_package_name, class 63, 65
 less_package_name_globus_site_version, class
 63, 65
 less_package_name_version, class 63, 65
 less_package_version, class 63, 65
 less_timeout, class 63, 65

P

Parameter_Type, struct 91

R

Response_Type, class 68

S

Scan_Parse_Parameter_Type, class ... 56, 63, 66,
 68, 72, 84
 Session_Data_Type, class 85
 stack 68
 struct Distinguished_Name_Type 90
 struct Parameter_Type 91

X

X509_Cert_Type, class 56, 72

Function Index

~

~Entry_Type (Destructor) 64
 ~Scan_Parse_Parameter_Type 75
 ~Session_Data_Type 85
 ~Session_Data_Type (destructor) 85

A

accept_data 72, 83
 add_authorization 77
 add_download_info 77
 add_globus_site 80
 add_installation_script 77
 add_prerequisite 78

B

bin2hex 54

C

check_checksum 90
 check_database_tables 79
 check_delegation 75
 check_dir 86
 check_distinguished_name 90
 check_globus_site 93
 check_globus_site_admin 80
 check_grid_mapfile 75
 check_privileges 80
 check_public_key 81
 clear 67, 82, 91
 clear (Distinguished_Name_Type) 67
 clear (Distinguished_Name_Type) (scrinstl) 91
 clear (Globus_Site_Info_Type) 82
 client_connect_auth 44, 68, 72
 client_connect_non_auth 44, 66, 68, 72
 cond_append 76
 connect_func 43, 56, 68, 72, 89
 connect_func_non_auth 43, 68, 72
 create_links 50
 create_temp_file 86
 cull_database 88

D

decrypt 90
 delete_entries 77
 delete_environment_shellscripts 80
 delete_fifos 86
 delete_globus_sites 80
 delete_key_pair 81
 delete_prerequisites 79

delete_public_key 81
 delete_temp_files 86, 90
 deque_exchange_front_2 86
 dir_is_empty 86
 disable_globus_site 80
 Distinguished_Name_Type 66, 91
 Distinguished_Name_Type (Constructor) 66
 Distinguished_Name_Type (Constructor) (scrinstl) 91
 Distinguished_Name_Type::clear 67
 Distinguished_Name_Type::clear (scrinstl) 91
 Distinguished_Name_Type::Distinguished_Name_Type 66
 Distinguished_Name_Type::Distinguished_Name_Type (scrinstl) 91
 Distinguished_Name_Type::operator!= 67
 Distinguished_Name_Type::operator!= (scrinstl) 91
 Distinguished_Name_Type::operator= 66
 Distinguished_Name_Type::operator== 67
 Distinguished_Name_Type::operator== (scrinstl) 91
 Distinguished_Name_Type::set 66
 Distinguished_Name_Type::set (scrinstl) 91
 Distinguished_Name_Type::show 67
 Distinguished_Name_Type::show (scrinstl) 91
 distribute_key_pair 81

E

edit_globus_site 80
 edit_globus_site_admin 80
 enable_environment_shellscript 80
 enable_globus_site 80
 Entry_Type 64
 Entry_Type (Constructor) 64
 Entry_Type (Copy constructor) 64
 Entry_Type (Default constructor) 64
 Entry_Type::~Entry_Type 64
 Entry_Type::Entry_Type 64
 Entry_Type::extract_prerequisites 65
 Entry_Type::operator< 65
 Entry_Type::operator= 65
 Entry_Type::operator== 65
 Entry_Type::operator> 65
 Entry_Type::show 65
 equal_package_name_globus_site::operator() 65
 equal_package_name_version_globus_site::operator() 65
 exchange_data_with_client 43, 63, 68, 72
 exchange_data_with_server 44, 68, 72
 extract_distinguished_name 58

extract_dn_fields 54, 56, 72
 extract_prerequisites 65

F

fetch_authorization 78
 fetch_download_info 78
 fetch_environment_shellscripts 80
 fetch_installation_script 78
 fetch_key_pair 80
 fetch_prerequisites 78
 finish 44, 89

G

gen_enc_rd_str 90
 generate_dh_params 53
 generate_dh_params_non_auth 53
 generate_rsa_params 54
 get_database_username 57, 75
 get_database_username (X509_Cert_Type) 57
 get_date_time 86
 get_datestamp 42, 90
 get_dir 86, 90
 get_encrypted 78
 get_entries 63, 78
 get_entries (Scan_Parse_Parameter_Type) 63
 get_in_addr 55
 get_key_name 80
 get_key_pair 80
 get_name 90
 get_optinum_installer_gpg_pubkey_
 fingerprint 81
 get_proxy_cred_filename 86
 get_session_data 81
 get_uid 90
 Globus_Site_Info_Type 82
 Globus_Site_Info_Type (Default constructor)
 82
 Globus_Site_Info_Type::clear 82
 Globus_Site_Info_Type::Globus_Site_Info_
 Type (Default constructor) 82
 Globus_Site_Info_Type::show 82
 GPG_Key_Pair_Type 84
 GPG_Key_Pair_Type::GPG_Key_Pair_Type 84
 GPG_Key_Pair_Type::show 84

H

handle_options 96
 handle_proxy_cred 58, 89
 hidden_install 51, 72

I

init 47
 init_globus_site_map 72, 86
 init_maps 42

initialize_tls_session 53
 initialize_tls_session_non_auth 53
 insert 57
 insert (X509_Cert_Type) 57
 install_key_pair 81
 install_packages 49, 63, 72
 is_in_database 57
 is_in_database (X509_Cert_Type) 57

L

less_package_name::operator() 65
 less_package_name_globus_site_
 version::operator() 65
 less_package_name_version::operator() 65
 less_package_version::operator() 65
 less_timeout::operator() 65
 listen_auth 54, 72
 listen_non_auth 55, 72
 lock_cerr_mutex 42, 90
 lock_table 79
 lock_tables 79
 lookup 75

M

main 43, 44, 47, 89, 92, 93, 96
 mark_entries 78
 mark_globus_site 80

O

operator!= 67, 91
 operator!= (Distinguished_Name_Type) 67
 operator!= (Distinguished_Name_Type)
 (scrinstl) 91
 operator< 65
 operator< (Entry_Type) 65
 operator= 65, 66, 85
 operator= (Distinguished_Name_Type) 66
 operator= (Entry_Type) 65
 operator= (Session_Data_Type) 85
 operator== 65, 67, 91
 operator== (Distinguished_Name_Type) 67
 operator== (Distinguished_Name_Type)
 (scrinstl) 91
 operator== (Entry_Type) 65
 operator> 65

P

parse_args 47
 parse_input 72, 83
 print_info 54
 print_x509_certificate_info 54
 process_command_line_options 13

R

remote_install_func 63, 68, 72, 87
 remote_install_rule_func 63, 68, 72, 83
 request_prerequisites 78
 Response_Type 71
 Response_Type::Response_Type 71
 Response_Type::show 71

S

Scan_Parse_Parameter_Type 75
 Scan_Parse_Parameter_Type::~Scan_Parse_
 Parameter_Type 75
 Scan_Parse_Parameter_Type::add_
 authorization 77
 Scan_Parse_Parameter_Type::add_download_
 info 77
 Scan_Parse_Parameter_Type::add_globus_site
 79
 Scan_Parse_Parameter_Type::add_
 installation_script 77
 Scan_Parse_Parameter_Type::add_prerequisite
 78
 Scan_Parse_Parameter_Type::check_database_
 tables 79
 Scan_Parse_Parameter_Type::check_delegation
 75
 Scan_Parse_Parameter_Type::check_globus_
 site_admin 80
 Scan_Parse_Parameter_Type::check_grid_
 mapfile 75
 Scan_Parse_Parameter_Type::check_privileges
 80
 Scan_Parse_Parameter_Type::check_public_key
 81
 Scan_Parse_Parameter_Type::cond_append ... 76
 Scan_Parse_Parameter_Type::delete_entries
 77
 Scan_Parse_Parameter_Type::delete_
 environment_shellscripts 80
 Scan_Parse_Parameter_Type::delete_globus_
 sites 80
 Scan_Parse_Parameter_Type::delete_key_pair
 81
 Scan_Parse_Parameter_Type::delete_
 prerequisites 79
 Scan_Parse_Parameter_Type::delete_public_
 key 81
 Scan_Parse_Parameter_Type::disable_globus_
 site 80
 Scan_Parse_Parameter_Type::distribute_key_
 pair 81
 Scan_Parse_Parameter_Type::edit_globus_site
 80
 Scan_Parse_Parameter_Type::edit_globus_
 site_admin 80
 Scan_Parse_Parameter_Type::enable_
 environment_shellscript 80

Scan_Parse_Parameter_Type::enable_globus_
 site 80
 Scan_Parse_Parameter_Type::fetch_
 authorization 78
 Scan_Parse_Parameter_Type::fetch_download_
 info 78
 Scan_Parse_Parameter_Type::fetch_
 environment_shellscripts 80
 Scan_Parse_Parameter_Type::fetch_
 installation_script 78
 Scan_Parse_Parameter_Type::fetch_key_pair
 80
 Scan_Parse_Parameter_Type::fetch_
 prerequisites 78
 Scan_Parse_Parameter_Type::get_database_
 username 75
 Scan_Parse_Parameter_Type::get_encrypted
 78
 Scan_Parse_Parameter_Type::get_entries... 63,
 78
 Scan_Parse_Parameter_Type::get_key_name .. 80
 Scan_Parse_Parameter_Type::get_key_pair .. 80
 Scan_Parse_Parameter_Type::get_optinum_
 installer_gpg_pubkey_fingerprint 81
 Scan_Parse_Parameter_Type::get_session_data
 81
 Scan_Parse_Parameter_Type::install_key_pair
 81
 Scan_Parse_Parameter_Type::lock_tables ... 79
 Scan_Parse_Parameter_Type::lookup 75
 Scan_Parse_Parameter_Type::mark_entries .. 78
 Scan_Parse_Parameter_Type::mark_globus_site
 80
 Scan_Parse_Parameter_Type::request_
 prerequisites 78
 Scan_Parse_Parameter_Type::Scan_Parse_
 Parameter_Type 75
 Scan_Parse_Parameter_Type::send_data_to_
 peer 81
 Scan_Parse_Parameter_Type::show 81
 Scan_Parse_Parameter_Type::show_
 certificates 75
 Scan_Parse_Parameter_Type::show_entries .. 78
 Scan_Parse_Parameter_Type::show_
 environment_shellscripts 80
 Scan_Parse_Parameter_Type::show_globus_
 sites 80
 Scan_Parse_Parameter_Type::show_privileges
 80
 Scan_Parse_Parameter_Type::store_
 environment_shellscript 80
 Scan_Parse_Parameter_Type::store_key_pair
 80
 Scan_Parse_Parameter_Type::store_public_key
 80
 Scan_Parse_Parameter_Type::sub_fetch_
 prerequisites 78

Scan_Parse_Parameter_Type::submit_mysql_
 query..... 81
 Scan_Parse_Parameter_Type::validate_
 distinguished_name..... 75
 Scan_Parse_Parameter_Type::write_
 emendations..... 77
 Scan_Parse_Parameter_Type::write_entries
 75
 send_data_to_peer..... 81
 Session_Data_Type..... 85
 Session_Data_Type (constructors)..... 85
 Session_Data_Type::~Session_Data_Type
 (destructor)..... 85
 Session_Data_Type::operator=..... 85
 Session_Data_Type::Session_Data_Type
 (constructors)..... 85
 Session_Data_Type::show..... 85
 set..... 57, 66, 91
 set (Distinguished_Name_Type)..... 66
 set (Distinguished_Name_Type) (scrinstl) .. 91
 set (X509_Cert_Type)..... 57
 set_cert_files..... 42
 set_debug_level..... 86
 show..... 57, 65, 67, 71, 81, 82, 84, 85, 91
 show (Distinguished_Name_Type)..... 67
 show (Distinguished_Name_Type) (scrinstl)
 91
 show (Globus_Site_Info_Type)..... 82
 show (Session_Data_Type)..... 85
 show (X509_Cert_Type)..... 57
 show_certificates..... 75
 show_entries..... 78
 show_environment_shellscripts..... 80
 show_filename_arguments..... 13
 show_globus_sites..... 80
 show_privileges..... 80
 show_user_info..... 90
 store_environment_shellscript..... 80
 store_key_pair..... 80
 store_public_key..... 80
 sub_distribute_key_pair..... 68, 72, 84, 88
 sub_fetch_prerequisites..... 78
 submit_mysql_query..... 81

T

tcp_close..... 55
 tcp_connect..... 55

U

unlock_cerr_mutex..... 42, 90
 unlock_tables..... 79

V

validate_distinguished_name..... 75
 verify_cert2..... 53, 72
 verify_certificate..... 53
 verify_certificate_chain..... 53, 56, 72
 verify_last_cert..... 53, 72

W

wait_for_child_processes..... 90
 wait_func..... 89
 wrap_db_deinit..... 54
 wrap_db_delete..... 54
 wrap_db_fetch..... 54
 wrap_db_init..... 54
 wrap_db_store..... 54
 write_emendations..... 77
 write_entries..... 75

X

X509_Cert_Type..... 56, 57
 X509_Cert_Type (Constructor)..... 56
 X509_Cert_Type::get_database_username..... 57
 X509_Cert_Type::insert..... 57
 X509_Cert_Type::is_in_database..... 57
 X509_Cert_Type::set..... 57
 X509_Cert_Type::show..... 57
 X509_Cert_Type::X509_Cert_Type..... 56

Y

yyerror..... 48, 59
 yylex..... 47, 59, 72
 yyparse..... 16, 17, 48, 60, 63, 68, 72
 yyscan..... 17
 yywrap..... 48, 59

Z

zzerror..... 61
 zzlex..... 61, 72
 zzparse..... 16, 17, 62, 68
 zzscan..... 17
 zzwrap..... 61

Concept Index

-
- _YY (suffix) 17
- _ZZ (suffix) 17

- A**
- absolute path 1
- Abstract Syntax Notation One 2
- actions, parser 16
- Apache 45
- Apache httpd 45
- archive 1
- ASN.1 2
- association table 98
- asynchronous 88
- authentication 16
- authentication/authorization 16
- authorization 16

- B**
- b2zip 1
- Bison 60, 62
- bundle, CA certificate 45

- C**
- CA (Certification Authority) 13, 45
- CA certificate 45
- CA certificate bundle 45
- Certificate Revocation List (CRL) 13, 52
- certificate, CA (Certification Authority) 45
- certificate, X.509 66
- Certification Authority (CA) 13, 45
- CGI (Common Gateway Interface) 45
- CGI program 45
- CGI script 45
- characters, punctuation 19
- checksum 93
- command, download 50
- Common Gateway Interface (CGI) 45
- compressed archive file 1
- compression 1
- credential, proxy 36, 66
- CRL 52
- CRL (Certificate Revocation List) 13
- culling software package entries 26
- curl 45
- customer 51
- CWEB 1

- D**
- dæmon 14
- dæmon process 14
- database 2
- database, MySQL 2
- default installation script 49
- default script, software installation 49
- delegate 29, 36, 74
- delegation 29, 36, 74
- deleting prerequisites 26
- deque 68
- detached threads 88
- Distinguished Name 29, 36, 44, 66
- distribution 1
- download command 50
- download URL 50

- E**
- effective user 29, 74
- encoding, URL 45
- environment shellscrips 31
- executables 1
- expansion, shell 15, 21
- expansion, shell, suppressing 15

- F**
- FastCGI 45
- FastCGI program 45
- FastCGI script 45
- file permissions, UNIX 14
- Flex 19
- friend 56, 63, 65, 66, 68, 72, 82, 84, 85
- front-end node 33
- front-end, Globus 66

- G**
- Globus front-end 66
- Globus Toolkit 66
- GNU Bison 60, 62
- GNUTLS 16
- grammar, rules of 16
- grid 33
- grid computer 33
- grid computer cluster 33
- grid computing 33
- grid-mapfile 66
- gzip 1

- H**
- handshake 16
- hidden software installation 50
- httpd 45

I

installation script 50

K

keyword 18

L

language 16
 link, symbolic 50
 listen 54, 55
 listening 54, 55
 literate programming 2
 local software installation 49
 login names 66

M

many-to-one 98
 md5 93
 MySQL 2
 MySQL database 2

N

Name, Distinguished 66
 non-terminal symbol 62
 non-terminal symbols 17
 non-terminals 62

O

one-to-many 98
 one-to-one 98
 operating system 45
 options 18

P

package database 45
 parser actions 16
 parser functions 16
 parser rules 16, 68
 passphrase 93
 password 93
 path 1
 PATH 1
 permissions, root, UNIX 14
 permissions, UNIX 14
 POSIX threads 62
 prerequisites 49
 prerequisites, software 49
 privileges 98
 program state 68
 programming, literate 2
 proxy credential 36, 66

proxy credential, X.509 14
 punctuation 19
 punctuation characters 19

R

randomness 94
 recursion 18
 recursive rules 18
 reducing 17
 remote installation 31
 request (to webserver) 45
 response 68
 revocation list 52
 RLM, download 50
 root permissions, UNIX 14
 root, UNIX 14
 rules of grammar 16
 rules, parser 16

S

scanner (Flex) 19
 sha1 93
 sha224 93
 sha256 93
 sha384 93
 sha512 93
 shell expansion 15, 21
 shell expansion, suppressing 15
 shell expansion, wildcards 21
 software installation, default script 49
 software installation, hidden 50
 software installation, local 49
 software prerequisites 49
 software provider 51
 stack 68
 start symbol 18
 state (of program) 68
 suffixes 17
 superuser 98
 symbol, non-terminal 62
 symbol, terminal 62
 symbolic link 50
 symbols, non-terminal 17
 symbols, terminal 17
 synchronous 88

T

tarball 1
 terminal symbol 62
 terminal symbols 17
 threads 62
 threads, detached 88
 threads, POSIX 62
 TLS (Transport Layer Security) 16
 token 62

tokens (non-terminal symbols) 17
Transport Layer Security, TLS 16

U

URL 50
URL encoding 45
user, effective 29, 74
UTC 100

W

webserver 45

wildcards (shell expansion) 21
Windows 45
Windows operating systems 45

X

X.509 certificate 66
X.509 certificates 44
X.509 proxy credential 14

Z

zip 1