v22 Ada Framework User Manual

Sowebio SARL
15, rue du Temple
17310 – St Pierre d'Oléron – France

Capital 15 000 EUR – SIRET 844 060 046 00019 – RCS La Rochelle – APE 6201Z – TVA FR00844060046

Ed.	*Release*	*Comments*	
1	20230731	Initial release	sr
18	20230821	General update, v22 & UXString Api integration	sr
22	20230824	General update, Gnoga integration	sr
34	20230912	First alpha release on private github repository	sr
74	20231102	Second alpha release on public github repository	sr
77	20231128	Rewrite Setup chapter for Alire tooling	sr
120	20240318	Third alpha release on public github repository	sr
150	20241216	First beta release on public github repository	sr
152	20250107	Add Uxf package in API chapter	sr
156	20250213	Update cover and footer	sr
157	20250214	Add Pdf package, Net and Prg functions in API chapter	ls
170	20250311	Update v22 build	sr
173	20250326	General install update	xp
175	20250326	Typos	
180			

- Authors

  Stéphane Rivière - sriviere@soweb.io [CTO Sowebio]
  Théodore Gigault - https://v22.soweb.io/people [Sowebio intern 2022]
  Arthur Le Floch - https://v22.soweb.io/people [Sowebio intern 2023]
  Xavier Petit - contact@xpetit.net [Contributions 2024]
  Lounès Souakri - https://v22.soweb.io/people [Sowebio intern 2025]

- Acknowledgments

  v22 is based on a number of Ada open-source libraries. See Appendices for copyrights and credits.

- Editor

  Stéphane Rivière [Number Six] - sriviere@soweb.io [CTO Sowebio]

  The “Excuse me I'm French” speech - The main author of this manual is a Frenchman with basic English skills. Frenchmen are essentially famous as frog eaters[1]. They have recently discovered that others ~~forms of communication~~ languages are widely used on earth. So, as a frog eater, I've tried to write some stuff in this foreign dialect loosely known here under the name of English. However, it's a well known fact that frogs don't really speak English. So your help is welcome to correct this bloody manual, for the sake of the wildebeests, and penguins too.

- Syntax notation

  Inside a command line:
  - A parameter between brackets **[ ]** is optional;
  - Two parameters separated by **|** are mutually exclusives.

  An important notice:

  ⇨ This is an important notice !

- Edition

  1 180 - 2025-06-26

[1] We could be famous as designers of the Concorde, Ariane rockets, Airbus planes or even Ada computer language but, definitely, Frenchmen have to wear beret with bread baguette under their arm to go eating frogs in a smokey tavern. That's *le cliché* :)

https://this-page-intentionally-left-blank.org

Contents

Introduction

Keep It Simple, Stupid.
Clarence Leonard "Kelly" Johnson

1 About v22 framework

1.1 Ready to use in production

v22 is a general purpose, KISS[2] oriented, modular Ada framework for GNU/Linux Debian and derivatives[3] service, console and web programs.

v22 is composed of many packages in charge of UTF-8 strings, program and OS functions, HTTP[S]/WS[S] web framework, console handling and text files, advanced network, MySQL and SQLite high level binding, logging and configuration files handling.

Originally based on the v20 library, the v22 framework represents a major step forward in the following areas:
- UTF-8 compatibility;
- Simplified string processing (only one UTF-8 String type is used);
- Internationalization;
- New and extended database API;
- Extended database access to MySQL, in addition to SQLite;
- Improved concurrent access and performance for SQLite;
- Unlimited simultaneous SQLite and MySQL connections for a single program;
- New LGPLv3 licensing instead of GPLv3;
- New FSF GCC development environment not tied anymore to GPLv2 license;
- And much more.

1.2 Cooperative and open

v22's native dependencies are Gnoga, Simple_Components, UXStrings and Zanyblue.

v22 is both a high-level framework and an extension to the lower level components cited above.

[2] Keep It Simple, Stupid - https://en.wikipedia.org/wiki/KISS_principle - In memory of http://www.nasonline.org/publications/biographical-memoirs/memoir-pdfs/johnson-clarence.pdf the genius father of titanium Blackbirds.

[3] Debian and derivatives features are limited to package management. Package management could be extended to other distributions. More generally, it would be possible and even desirable to adapt v22 to BSD or Mac OS systems. Any help on these subjects is welcome, as the main author of v22 only uses GNU/Linux Debian.

v22 has been designed to:
- Use unmodified components;
- Not "reinvent the wheel". Component API are to be used first;
- Offer higher-level functions, or functions that do not exist in the components.

v22 must be used with the APIs of:
- UXStrings to process UTF-8 strings;
- Gnoga to connect to MySQL and/or SQLite databases;
- Gnoga to provide a Web interface;
- Simple_Components for network layer and more;
- Zanyblue if internationalization is desired.

In short:
- UXStrings is used throughout v22. The v22.Uxs package extends UXStrings functionality. The v22.Sql package extends the functionality of Gnoga.Server.Database. The v22.Gui graphics framework is based on Gnoga.Gui.
- v22's architecture allows it to be open to additional packages, depending on the software development required.

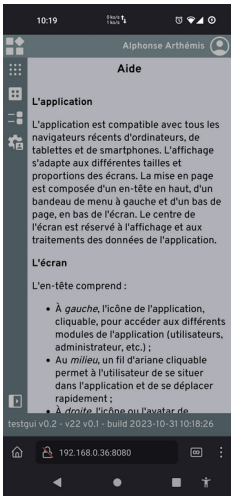
2 Everybody wants screenshots

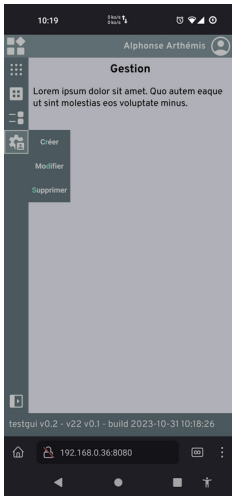
v22 is as much console-based as it is web-based. Of course, it's the web that concerns us. v22 is fully responsive for smartphones and tablets.

We show some screenshots of PC workstation too. They are similar but automatically arranged on two columns by responsive design. v22 was tested on Linux Ubuntu 22.04 LTS with Firefox and Chrome browsers, MacOS High Sierra and Windows 10 with various browsers.

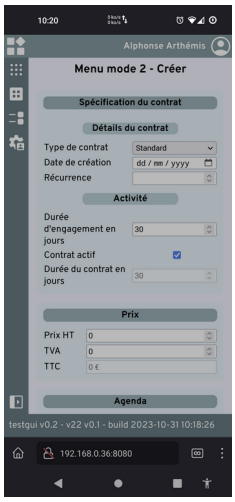
Below are a few screenshots of the demo program, on an android smartphone. v22 was tested against a 2013 Nexus 5 (with a recent Firefox browser), a 2015 stock Nexus 5x, a 2020 degooglized Pixel 5 (Firefox), various recent and standard Samsung (Chrome) and iPhone (Safari).

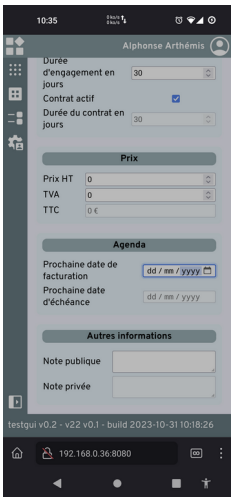
2.1 Login, help and sub-menu screens

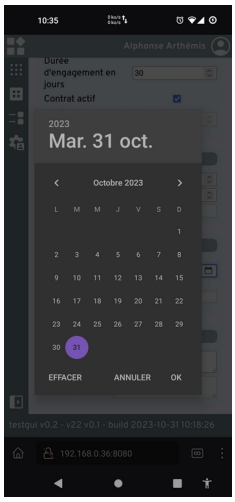


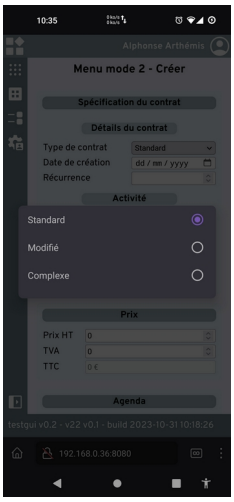


2.2 Full page screens

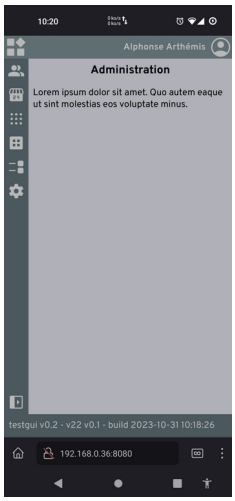


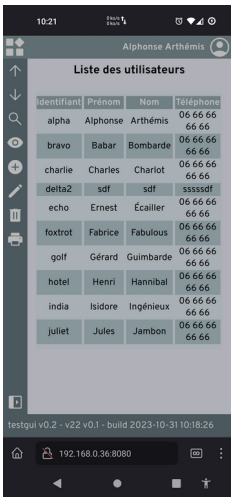


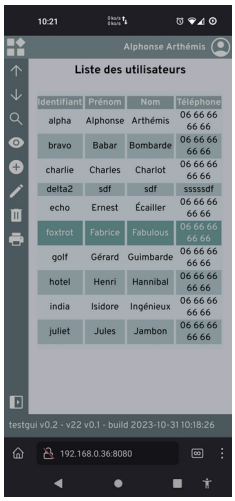


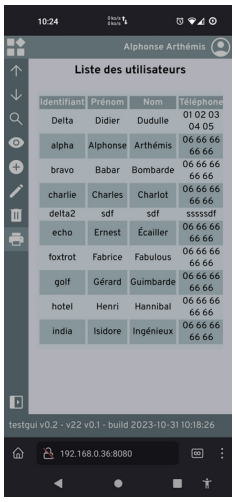


2.3 Admin users management

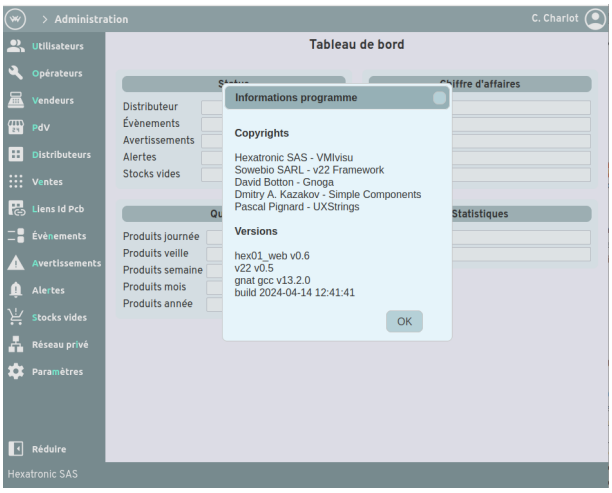


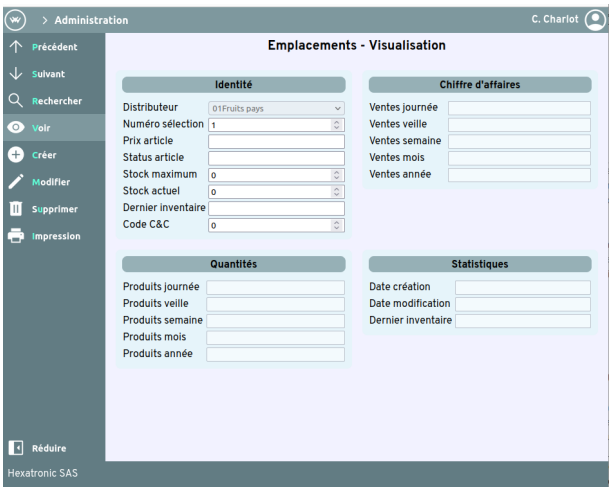






2.4 PC Workstation screenshots





Theses screens shows the extended left menu with automatic generated keyboard shortcuts.

3 About the Ada Community

At first, thanks to the Ada Community, definitely one of the best.

3.1 Inspiration, packages used, ideas, help and more

AdaCore Ada compiler: https://www.adacore.com/community

David Botton: https://www.linkedin.com/in/david-botton-3741b210 [Gnoga]

Dmitry A. Kazakov: http://www.dmitry-kazakov.de/ada/components.htm [Simple Components, used in Gnoga]

Gautier de Montmollin: https://github.com/zertovitch [Gnoga maintainer and much more]

Jean-Pierre Rosen: https://adalog.fr [Ada teacher, writer and much more]

Jeffrey R. Carter: https://github.com/jrcarter/PragmARC [PragmAda Reusable Components]

Michael Rohan: https://sourceforge.net/projects/zanyblue [Internationalization, used in Gnoga and v22]

Pascal Pignard: https://github.com/Blady-Com [UXStrings used in Gnoga and v22, Gnoga maintainer and much more]

3.2 Special thanks

Special thanks to Ada gurus Daniel Feneuille, Gautier de Montmollin, Pascal Pignard and Jean-Pierre Rosen. The chapter heading quotes are extracted from Murphy's Law

and other reasons why things go wrong - A. Bloch. They come from https://www.ada-log.fr site created by Jean-Pierre Rosen.

4 v22 history

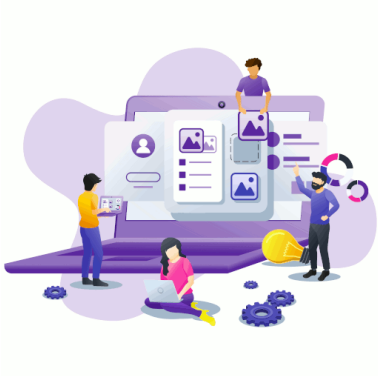
We own the copyrights for v89, v90, v93, v95, v04, v20 and v22.

Some work in v22 framework is derived from theses libraries.

Ver.	*Languages*	*Processor*	*OS*	*Context*	*Copyright*	*Users*
v87	Clipper	i386	MsDos	ST Formation	Proprietary	CEA-DAM, CEA & EDF
v89	Clipper/C/Asm	i386	MsDos	Atlansys	Proprietary	ETDE, SAMU & EDF
v90	Clipper/C/Asm	i386	MsDos	Atlansys	Proprietary	Military, NGO & EDF
v93	C++	i386	Windows	Atlansys	Proprietary	Research
v95	Delphi	i386	Windows	Astriane	Proprietary	Military & NGO
v96	Asm	st62xx	Embedded	MRT	Proprietary	Military & Civilian
v97	Asm	pic17c44	Embedded	MRT	Proprietary	Military & Civilian
v04	Ada	i386	Windows	AIDE	GMGPL	Education
v20	Ada	x86-64	Linux	Sowebio	GPL v3	Terminal GP
v22	Ada	x86-64	Linux	Sowebio	LGPL v3	Terminal & Web GP

Setup

Doubling the number of programmers on a late project does not make anything else than double the delay.
Second Brook's Law



1 GNAT toolchain

To install an Ada toolchain suitable for v22, get **ADEL** [Ada Development Environment on Linux] manual at https://github.com/sowebio/adel-doc and apply the second chapter **GNAT Toolchain**.

2 Dependencies

2.1 Ada dependencies

v22 mainly depends on Gnatcoll, Gnoga, UXStrings, Zanyblue, LibcURL, PdfWriter.

Gnoga mainly depends on Simple_Components, UXStrings and Zanyblue.

So there's a strong community of dependencies between v22 and Gnoga.

Gnoga, Simple_Components, UXStrings, Zanyblue, PdfWriter are included in v22.

2.2 System Dependencies

Most of the packages below are only required for Gnoga tests and demonstrations.

⇨ v22 uses its own source version of SQLite and *doesn't need libsqlite3-* packages.*

MySQL support in v22 *requires libmariadb3, libmariadb-dev and libmariadb-dev-compat* for connection to a MariaDB database, which is preferred to MySQL, whose support on Debian and Ubuntu systems is considered rather low. But the final choice is yours.

Although this documentation always refers to MySQL, v22 is compatible with both MySQL, through its native libmysqlclient, and MariaDB, thanks to the libmariadb-dev-compat package, which emulates the libmysqlclient layer.

v22 needs libcurl4 and libcurl4-openssl-dev if v22.Crl package is used.

Finally, the libgnutls30 package is required to implement a TLS connection such as the https and wss protocols. If your application is behind a TLS proxy, this library is no longer needed.

The v22 framework provides the functionality for a v22 application to control and install the packages required for its proper execution.

	Paquet	***Commentaire***
	bind9-dnsutils	Net.Send_Mail [nslookup]
Email	sendemail	Net.Send_Mail [sendEmail.pl] https://github.com/mogaal/sendemail
LibcURL	libcurl4	LibcURL – v22.Crl [obsolete]
	libcurl4-openssl-dev	LibcURL development files witj OpenSSL [obsolete]
MySQL	libmariadb3	MariaDB database client library – v22.Sql
	libmariadb-dev	MariaDB database development files
	libmariadb-dev-compat	MariaDB Connector/C, compatibility symlink avec MySQL
	mariadb-client	Optionnel, pour le client CLI MySQL
	mariadb-server	Optionnel, pour avoir le serveur en local sur sa station
TLS/https	libgnutls30	GNU TLS library - main runtime library – v22.Gui
Gtk	pkg-config	Manage compile and link flags for libraries
	libwebkit2gtk-4.0-37	Web content engine library for GTK
	libwebkit2gtk-4.0-dev	Web content engine library for GTK - development files
	libgtk-3-0	GTK graphical user interface library
	libgtk-3-dev	Development files for the GTK library
	libjavascriptcoregtk-4.0-dev	JavaScript engine from WebKitGTK - development files

2.2.1 Minimal setup reminder for Ubuntu 22.04 LTS, Debian 11 & 12

```
user@system: sudo apt install libmariadb3 libmariadb-dev libmariadb-dev-compat
```

3 v22 installation

3.1 Build framework and test programs

Assuming you wish to install v22 under <your path> with a GNAT toolchain already installed, do the following from a command line interpreter.

Open a terminal:

```
user@system: cd <your path>

user@system: git clone --recurse-submodules https://github.com/sowebio/v22

user@system: cd ./v22
user@system: alr build

ⓘ Building v22=0.1.0-dev/v22.gpr...
Setup
   [mkdir]        object directory for project v22
Compile
   [Ada]          testgui.adb
```

```
   [Ada]          testapi.adb
   [C]            sqlite3.c
   [Ada]          s-memory.adb
   [Ada]          testapi_cfg.adb
   [Ada]          testapi_msg.adb
   [Ada]          testapi_sql.adb
   [Ada]          testapi_sys.adb
   [Ada]          testapi_tio.adb
   [Ada]          uxstrings.adb
   [Ada]          v22.adb
   [Ada]          v22-cfg.adb
   [Ada]          v22-fls.adb
   [Ada]          v22-msg.adb
   [Ada]          v22-net.adb
   [Ada]          v22-prg.adb
   [Ada]          v22-sql.adb
   [Ada]          v22-sys.adb
   [Ada]          v22-tio.adb
   [Ada]          v22-uxs.adb
   .../...
   [Ada]          generic_unbounded_ptr_array.adb
Bind
   [gprbind]      testgui.bexch
   [Ada]          testgui.ali
   [gprbind]      testapi.bexch
   [Ada]          testapi.ali
Link
   [archive]      libv22.a
   [index]        libv22.a
   [link]         testgui.adb
   [link]         testapi.adb
✓ Build finished successfully in 42.56 seconds.
```

3.2 Directories

v22 comes with some inner directories:

Packages	***Description***
prg	test binary place, with dontdelete.me test file for trailing comments preservation
doc	place of sow - v22 Ada Library User Manual.pdf and others documentation files
doc-generated	API doc generated by GNATStrudio with GNATDoc
obj/debug obj/fast obj/small obj/style	build directories
src	sources of v22
src/sys	specials system files as s-memory.adb, the GNATColl memory monitory hook
src-testapi	v22 api test program
src-testgui	v22 gui test program

3.3 Notes

- V22 init

```
user@system: alr init --bin --in-place v22
user@system: alr with gnatcoll
```

Check the newly created alire.toml at the root of the project and add “executables” parameter:

```
name = "v22"
description = ""
version = "0.1.0-dev"

authors = ["Your Name"]
maintainers = ["Your Name <example@example.com>"]
maintainers-logins = ["github-username"]
licenses = ""
website = ""
tags = []

executables = ["test/testapi", "test/testgui"]

[[depends-on]]
gnatcoll = "^24.0.0"
```

- LibcURL [obsolete]

  v22.Crl package is still functional in archive but obsoleted and not included anymore.

```
user@system: cd /usr/lib/x86_64-linux-gnu

- On debian 11 the link libcurl.so to libcurl.so.4.7.0 is missing

user@system: ln -s libcurl.so.4.7.0 libcurl.so
```

Architecture

Weinberg's Second Law : If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization.
Gerald Weinberg

1 Introduction

v22 is currently in beta *stage*, so its architecture could be subject to *slights changes*. User feedback is *encouraged in order to improve it*.

2 Requirements

A Linux workstation or server and an Ada compiler from the GNAT/GCC family are required.

A GNU/Linux Debian or Debian based like Ubuntu or Mint is recommended.

Latest GNAT FSF from Alire is recommended.

3 Coding guidelines

3.1 General

English should be used, both for identifiers, comments and documentation.

Naming is capitalized using underscore for compound name. ex: Entry_Value.

Comments leaves two spaces between -- and text comment.

The v22 code tries to adhere to the recommendations of the *Ada 95 Quality and Style book*. Like many Ada programmers, we believe that clarity and standardization of code is a respectful practice for later readers.

One exception to this is the non-alignment of operators vertically, the main author of v22 finding that the extra spaces, sometimes very numerous, detract from the readability of a left-to-right reading.

An other exception is the source code length. 72 columns is not mandatory anymore with our large displays. The standard v22 margin has been extended to 131 columns.

3.2 Messages

- Msg.Info ("This is an information message.")

  Information messages starts with a capital and ends with a dot. Ending message with three dots are only allowed when a user input is waited.

```
20231030-102625 - INIT    - INF - TestGui.SQL_Ping > Armed for 3600s cycles
```

- Msg.Debug ("This is a debug message")

  Debug messages are free form messages to ease debugging. They are activated by the Msg.Set_Debug (On) procedure.

```
20231030-105547 - INIT    - DBG - Load Database_Pragma: journal_mode=WAL
```

- Msg.Error ("v22.Fls.Function_Name > Error creating mount point: " & Target_To_Mount & " linked to " & Mount_Point & " on " & Remote_Host)

  Error messages starts with the library or program hierarchy, following by a "greater than" sign, following by the error message, terminated with a colon and a space, then, optionally, followed by more information.

```
20231030-102818 - INIT    - ERR - v22.Tio.Open_Conf > Can't apply permissions to file
```

3.3 Naming

We've tried to stay close to Ada naming by using routine names such as Put, Put_Line or New_Line where appropriate.

We've also tried to remain consistent in the names of getters and setters. In any case, we're listening to what users have to say to improve the v22 name space.

3.4 Behavior

Unlike the Ada runtime does, the text mode *Open* function of v22 now logically opens in *File_In* mode (read mode).

4 Design

v22 is designed as a KISS[4] working framework. It does not attempt to reproduce the outstanding granularity of the Ada runtime but tries to reuse the best existing libre software resources to offer a simple yet highly productive environment.

4.1 Representations and types

It is a representation in the sense of an image, not a type, but intimately linked to the latter.

[4] Keep It Simple and *Safe* in our context.

Date is a representation with an associated type Ada String or SQL database Varchar(10).

Date Time Stamp is a representation with an associated type Ada String or SQL database Varchar(15).

Money is a defined Ada type, translated to Ada type Long_Long_Integer when storing to SQL database Bigint.

v22 representations and types are:

v22 Type	*Database*	*Storing format*
Date	Varchar(10)	YYYY-MM-DD
DateTime	Varchar(15)	YYYYMMDD-HHMMSS
Money	Bigint	-999_999_999_999.99 to 999_999_999_999.99

4.1.1 Date

The v22 Date is in ISO_8601 format.

It could be wised to prefix date identifiers (variables, columns or fields tables) with **Date_**.

4.1.2 DateTime

The v22 Date Time Stamp, or DTS, expresses the server's local time, following the daylight saving time, since its main use is to time-stamp logs and events.

It could be wised to prefix date time stamps identifiers (variables, columns or fields tables) with **DTS_**.

The DTS v22 format is sortable but shorter than the variable ISO_8601 format, with (8601-1:2019) or optionally without (8601:2004) the "T" between the date an time parts.

4.1.3 Money

MONEY is **type** Money **is delta** 0.01 **digits** 14, stored in database as Bigint at the lowest unit, usually cents for euros or dollars. Ada Bigint is Long_Long_Integer.

4.2 Types (program)

Name	*Packages*	*Description*
Boolean	Standard	
Character	Standard	
Float	Standard	Scientific computing
Integer	Standard	32 bits (as Long_Integer)
Long_Long_Integer	Standard	64 bits
Money	v22	Financial computing
String	UXStrings	Unbounded UTF-8 string subtyping by Pascal Pignard
Geo		Geographic Coordinates.

Name	Packages	Description
	handling ok	

4.3 Types [databases]

Nothing is really well defined in SQL, and the standard [not freely available] is betrayed in every implementation. SQLite's typing is particularly strange in this respect.

v22 gets around this problem, at type level, by limiting itself to basic six types, with normalized extra v22 data types such as Money, Datetime, Date.

We must not use MySQL-specific data types to remain compatible with SQLite.

Common types between MariaDB and SQLite used in v22:

MySQL	SQLite	Compatibility notes
Bigint	Bigint	-2^{63} to $2^{63} - 1$
Blob	Blob	65535 max
Integer	Integer	-2147483648 to 2147483647 [32bits wide, sign included]
Decimal	Decimal[i,d]	Decimal SQLite affinity REAL
Double	Double	Double SQLite affinity REAL
Float	Float	Float SQLite affinity REAL
Text	Text	65535 max UTF-8, UTF-16BE, UTF-16LE
Varchar[n]	Varchar[n]	65535 max

BLOB of huge sizes should be always stored as regular files.

4.4 Packages

Name	Packages	Description
v22	Base	
Bio	Binary I/O	Binary IO: Binary files, locking, etc.
Cfg	Configuration files	Simple and user friendly config files handling
Crl	cURL interface	cURL related
Fls	File system	File system related
Gui	Graphic User Interface	High level framework for Gnoga by David Botton
Msg	Logging	Log - Terminal and file log - on top of Tio
Net	Network	Remote command via SSH, Tor, Email & SMS sending, API handling for clients like hosters Ovh or software like Matomo, etc.
Pdf	Pdf handling	Routines for Ada-Pdf-Writer by Gautier de Montmollin
Prg	Program	Program and user related
Sql	SQL database	MySQL and SQLite high level implementation for Gnoga
Sys	System	Operating System related
Tio	Text I/O	Text Input/Output and Text files related
Uxf	UXStrings add-ons	UTF-8 extension routines for UXStrings by Pascal Pignard
Uxs	UXStrings add-ons	UTF-8 extension routines for UXStrings by Pascal Pignard
	Already coded	

4.5 Functions

About strings, v22 functions always handles and return String [UXString].

4.6 Databases

v22 handles two SQL databases flavors : MySQL and derivatives, like MariaDB or PerconaDB and SQLite.

v22 does not act deliberately as an ORM [no magic please[5]], but offers selected features for rapid development of database applications

4.7 Exceptions

V22 comes with an extended postmortem exception handler. This displays the trace not only on the screen, but also in a file bearing the application's name and .err extension.

A wide range of information is displayed and recorded:

```
--------------------------------------------------------------------------------

Exception time         : 20230921-123354
Program uptime         :  0h0 0m0 0s
Program build DT stamp : build 2023-09-21 12:33:15
Program name & version : testgui v0.1
Library name & version : v22 v0.1
Start directory        : /home/sr/Sowebio/informatique/dev/gpl/github/v22/tests
Home directory         : /home/sr
Ada mem. alloc. [bytes]: Ada Cur: [ 4132 ] Max: [ 4708 ]
All mem. alloc. [bytes]: All Cur: [ 15204352 ] Max: [ 15204352 ]

raised V22.RAISE_EXCEPTION.V22_EXCEPTION_TEST : v22.adb:81
[./testgui]
0x496f9e v22__raise_exception at v22.adb:81
0x41c1b1 _ada_testgui at testgui.adb:830
0x488d8f main at b__testgui.adb:1173
[/lib/x86_64-linux-gnu/libc.so.6]
0x7fb0fee29d8e
0x7fb0fee29e3e
[./testgui]
0x415973 _start at ???
0xfffffffffffffffe

--------------------------------------------------------------------------------

20230921-123354 - INIT    - MSG - testgui > Exception detected, finalize application

20230921-123354 - SQL T1  - MSG - Closing SQLITE database: v22_testapi_1
20230921-123354 - SQL T1  - MSG - Closing SQLITE database: v22_testapi_2
```

When an exception occurs, the databases connections are cleanly closed, the console text cursor is restored and specific exit status are set.

[5] https://kurapov.ee/eng/tech/ORM-is-harfmul-pattern

- Implementation notes

  In the GNAT Pro and CE releases, AdaCore has integrated the GNU/GCC utility addr2-line into the GNAT runtime. Addr2line is called to display the subroutine names and line numbers of the call stack.

  This feature is missing in the GNAT FSF version. This is a recurring complaint on the net, some people offer workarounds, but none of themwork here (Linux, GNAT FSF 11). Since the CE versions are terminated, there were two ways to restore this useful feature; the clean one, rebuild a full FSF compiler with this feature, and the fast dirty one. We obviously choose the latter.

  When handling an exception, v22 search if the addr2line utility exists on the system and then, for each line indicating a "???", we run, on the executable itself, a command addr2line -e <executable name> <address> and analyze the return. If the latter is valid (example: /home/sr/.../debug/gnx-utl_sv.adb:1174, as opposed to "addr2line: DWARF error: can't find .debug_ranges section." and/or "??:?", ??:0"), the function replaces "???" by "gnx-utl_sv.adb:1174".

  If the program was compiled with GNAT Pro or CE, this is transparent and if add2line was not found, the "???" are left as they are.

Components

Variables won't; Constants aren't.
Osborn Law

1 Gnoga

Most of the information in this chapter is taken from the Gnoga documentation, written by Gnoga author David Botton.

1.1 What is Gnoga

Defining Gnoga is an important first step to using it. Gnoga is a framework and associated tools for developing GUI applications using the Ada language, leveraging web technologies to provide a rendering engine.

Gnoga should not be confused with a web development framework. While Gnoga is very capable of creating web applications and perform in that role and is even more capable than most web development frameworks, using Gnoga for web applications is only one possibility.

Native applications for desktop and mobile are just as easy to create, all using the same code base.

1.2 How does Gnoga work?

A Gnoga application can be divided into three parts:
- The application code written in Ada using the Gnoga framework and tools;
- The communication layer;
- The GUI rendering surface, an HTML-5 compliant browser or an embedded widget for native platform development.

The communication layer is not passive as in typical web programming using HTTP, nor is it using Ajax calls to simulate a live-active connection. It is an active, open connection using HTML5 web sockets or direct access at the API level to an embedded widget.

Since the communication layer is always active there is a stateful constant connection to the rendering surface as there is in traditional desktop GUI development even if the rendering surface is a remote browser.

The idea of using publishing technologies to display a GUI is not new. Next used Postscript; Mac OS X uses PDF; and Gnoga uses HTML 5.

1.3 Singleton and multi-connect applications

There are two basic application types in Gnoga, Singleton applications and Multi Connect applications. Singleton applications are ideal for single user desktop use. They allow only a single connection and exit when that connection is lost. Since the application will not be accessed in parallel by other connections implementation is easier in that there is no need to protect data except from parallel incoming events from a single user (In Gnoga events are not serialized and are fired as they are generated concurrently).

1.4 Concurrency and exceptions

While this example works, there is an issue. In Gnoga and in particular in Multi-Connect applications, concurrency and exceptions must always be considered, given that:

- While highly unlikely, it is possible for our Users vector to be accessed concurrently, something that that standard Ada containers are not designed to handle. If this were a production application, Users should be protected;

- While again highly unlikely given our application, it is possible for a view to be destroyed during the On_Change event. This could result in trying to call User_View.Draw on an already deallocated object. Therefore, it would be a good idea to capture exceptions in the On_Change event at the very least, or as part of protecting the Users vector a means be included to insure the validity of the User_View before calling Draw.

Gnoga will handle most exceptional situations not handled in your code, but creating a solid Multi-Connect application should include considerations for the above in all designs. Improving on our example is left as an exercise to the reader.

1.4.1 Advanced: The "Connection" Parameter and GUI elements on Stack

The extra parameter "Connection" in our controller procedure "Default" can be used when you wish to block the connection procedure until the connection is closed. The two common uses of Connection.Hold to block until connection loss are:

- To add clean up code on connection loss to the connection procedure; this could also have been added to the On_Destroy event for Main_Window.

- To prevent finalization of statically defined GUI elements within the connection procedure until the connection has been lost.

An example of this second method would allow us to rewrite the skeleton procedure as:

```ada
```

```
   procedure Default
     [Main_Window : in out Gnoga.Gui.Window.Window_Type'Class;
      Connection  : access
        Gnoga.Application.Multi_Connect.Connection_Holder_Type]
   is
      View : GnogaBoard.View.Default_View_Type;
   begin
      View.Create [Main_Window];
      View.Click_Button.On_Click_Handler [On_Click'Access];

      Connection.Hold;
   end Default;
```

1.4.2 Advanced: Per Connection App Data

In the multi-connect example above we use the connection's main view to store data specific for each user connection. It is often convenient to have a data structure containing the data specific to a connection. Gnoga offers a way to associate data to a connection and allow access to those data through any GUI element on that connection.

The following is an example:

```
``` ada
   type App_Data is new Connection_Data_Type with
      record
         Main_Window : Window.Pointer_To_Window_Class;
         Hello_World : aliased Common.DIV_Type;
      end record;
   type App_Access is access all App_Data;

   procedure On_Click [Object : in out Gnoga.Gui.Base.Base_Type'Class;
                       Event  : in     Gnoga.Gui.Base.Mouse_Event_Record]
   is
      App : App_Access := App_Access [Object.Connection_Data];
   begin
      App.Hello_World.Text ["I've been clicked"];
   end On_Click;

   procedure On_Connect
     [Main_Window : in out Gnoga.Gui.Window.Window_Type'Class;
      Connection  : access
        Gnoga.Application.Multi_Connect.Connection_Holder_Type]
   is
      App : App_Access := new App_Data;
   begin
      Main_Window.Connection_Data [App];
      App.Main_Window := Main_Window'Unchecked_Access;
      App.Hello_World.Create [Main_Window, "Click Me"];
      --  By default Connection_Data is assumed to be a dynamic object
      --  and freed when the connection is closed.  To use static app
      --  data pass Dynamic => False
      ...
   end On_Connect;
```
```

1.4.3 Multi Connect Applications for a Single User

A Multi-Connect application allows multiple connections to the same application at the same time. This does not always imply multiple users, it could even be the same user with multiple browser windows connected to the same application. When using a multi-connect application as a single-user desktop application you simply need to restrict access to the application to the local machine and provide some way for the application to know it is time to shutdown. Some tips:

- In the On_Connect procedure check if there has already been a connection and if a reconnect is tried return a view that indicates application is already running.

- If limiting to only one main window, use that window's On_Destroy event to tell the application to shut down using Gnoga.Application.Multi_Connect.End_Application if you do not provide some other way to exit the application, or track connections and end the application when all connections are closed.

- Limit connections to the local machine only: in initialize use Initialize (Host => "127.0.0.1");

1.4.4 Gnoga Types

By convention in Gnoga all types are usually defined in the following way and using the following naming convention:

```
type Some_Type is ....;
type Some_Access is access all Some_Type;
```

if Some_Type is a tagged type:

```
type Pointer_to_Some_Class is access all Some_Type'Class
```

1.4.5 Directory structure when developing apps

If you use the gnoga_make tool it will set up a development directory structure in addition to creating a skeleton application. (See the Singleton and Multi-Connect examples). For reference the following directory structure is the basic structure:

```
App Dir
  |
  |___ bin - your gnoga app binary
  |
  |___ html - boot.html (or other boot loader used)
  |
  |___ js - must contain jquery.min.js and boot.js
  |
  |___ css - optional, all files served as CSS files
  |
  |___ img - optional, files served as graphics files
  |
  |___ src - Source code for your gnoga app
  |
  |___ obj - Build objects
```

```
  |
  |___ templates - optional, if using Gnoga.Server.Template_Parser
  |
  |___ upload - optional, optional directory for incoming files
```

If any of the subdirectories are missing, the files expected to be in them are assumed to be in html. If the html subdirectory is also missing these files are assumed to be in App Dir. The executable can be in the bin directory or in App Dir.

1.4.6 Application, Types, Gui, Server, Client

See Gnoga API Application, Types, Gui, Server, Client for more details.

1.4.7 Plugins and Modules

Users can write and publish to the Gnoga Marketplace two Gnoga-specific UI extension types, Plugins and Modules.

Plugins, including jQuery, jQueryUI, Boot_Strap, and Ace_Editor, are bindings to JavaScript libraries for use on the client side.

Modules are unique Gnoga-based UI elements written with Gnoga.

1.4.8 Tags Bound in Gnoga

While Gnoga is not exactly HTML in Ada, knowing the relationships may be of assistance in developing your application:

- HTML5 Tags Bound as Gui Elements in Gnoga

HTML tags	***GUI Elements***
<a>,<hr>, ,<button>, ,<img>,<meter>, <progress>,<p>	Element.Common - see also Gui.View for and
<canvas>	Element.Canvas
<svg>	Element.SVG
<form>,<input>,<textarea>,<select>,<datalist>, <legend>,<label>,<option>,<optgroup>	Element.Form
<fieldset>	Element.Form.Fieldset
<audio>,<video>,<source>*,<track>*	Element.Multimedia
<iframe>	Element.IFrame
<html>,<body>,<head>	Access through Window_Type.Document
, , ,<dl>,<dd>,<dt>	Element.List
<address>,<article>,<aside>,<header>,<main>, <nav>,<p>,<pre>,<section>	Element.Section
<code>,,,<dfn>,<samp>,<kbd>, <var>,<marked>,<del>,<ins>,<s>, <q>,<big>,<small>,<time>,<tt>,<wbr>	Element.Phrase
<link>,<style>,<title>	Element.Style_Block, The Style property on Element_Type Document.Load_CSS, Document.Title Since content is generated by code
<table>,<caption>,<td>,<tr>,<th>,<col>, <colgroup>,<tfoot>,<thead>	Element.Table

* Not needed

- HTML5 Tags Unbound as Gui Elements in Gnoga

  Note: All tags can be bound and used with Element_Type.Create_With_HTML. For various reasons as described here, they are not bound specifically.

HTML tags	*GUI Elements*
<map>,<area>	No specific bindings currently for image maps; best generated with an automated tool as regular HTML.
<bdi>,<bdo>,<ruby>,<rp>,<rt>, <details>, <output>,<figure>,<flgcaption>	Text formatting tags are not bound. Have no application specific use. Span_Type should be used to contain text that needs interaction or interactive styling.
<object>,<embed>,<script>,<noscript>, <param>,<applet>	No bindings are made for external plugins or scripting tags
<base>,<meta>	base and meta only make sense for static pages
<dialog>,<keygen>,<menu>,<menuitem>	No browsers support these tags in a way worth binding yet
<frameset>,<frame>,<noframes>	No window level frame support; see Element.IFrame

1.4.9 Gnoga Concepts: in and out of the DOM

An HTML document is a hierarchical collection of objects. In a browser window, the displayed document is the browser's DOM, but it is possible within JavaScript to have objects that are not in the main DOM and have their own hierarchical collections, DOMs.

Gnoga maintains on the browser side JavaScript references to GUI elements it creates; these elements may or may not also be in the browser's DOM. When creating a new object in Gnoga, if the parent is in the browser's DOM, the child will be there as well. If not, it will just be part of the parent's individual DOM and not be visible, and even changing the visibility of that object will not make it appear on the browser window since they are not in the Browser's DOM.

To take an object and all its children out of the DOM use Gnoga.Gui.Element.Remove; to place it back in the DOM use one of the Place_* methods in Gnoga.Gui.Element.

1.4.10 Gnoga Concepts: Display, Visible, Hidden

HTML5 uses a few different independently working properties for visibility.

Visible will turn on or off the visibility of an element and its children, but the objects will still take the same space on the page.

Hidden will turn off visibility and the object and its children will no longer take up space on the page, either.

Display changes how the elements are laid out by the browser. Using Display (None) acts in the same fashion as Hidden.

1.4.11 Gnoga Concepts: Inner_HTML, Text and Value

Retrieving the contents of an Element in Gnoga differs depending on the type of Element. For form Elements the Value method is used. For others Text can be used to retrieve the text alone or Inner_HTML to retrieve the contents including any HTML tags present.

The reason there are different methods is based on the way the underlying HTML 5 works. Text and Inner_HTML are retrieving all child nodes with in the element while Value is an attribute of Form elements. So Text or Inner_HTML will return the contents of every child.

2 jQuery & jQuery-UI

2.1 Introduction

v22.Gui, using Gnoga, is JQuery and JQuery-UI based. It's a technology that's almost 20 years old, in use everywhere, very well maintained and documented, and makes no claim to anything other than being modest, durable and compatible with all browsers. In a word, JQuery and its ecosystems are KISS.

The coupling with websocket communication is Gnoga's stroke of genius from its designer, David Botton.

2.1.1 jQuery

According to the web site, “jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.”

https://jquery.com
https://api.jquery.com (api)
https://releases.jquery.com/jquery (all versions downloads)

2.2 jQuery-Ui

2.2.1 Liens

https://jqueryui.com

2.2.2 Theme creator

- Creation

Create jQuery-Ui theming a breeze with https://jqueryui.com/themeroller.

After downloaded your new custom theme, *just update the new generated jquery-ui.min.css*, which is the addition jquery-ui.structure.css plus jquery-ui.theme.css, *you don't need to include these last two files.*

Reference : https://jqueryui.com/upgrade-guide/1.11/#simplified-custom-and-quick-downloads

- Updating

To update an existing theme, *always use the embedded link included* in the jquery-ui.min.css which embeds all your custom theme settings. Example:

```
http://jqueryui.com/themeroller/?scope=&folderName=custom-theme&bgImgOpacityHeader=&bgImgOpacityContent=&bgImgOpacityDefault=&bgImgOpacityHover=&bgImgOpacityActive=&bgImgOpacityHighlight=&bgImgOpacityError=&cornerRadiusShadow=8px&offsetLeftShadow=0px&offsetTopShadow=0px&thicknessShadow=5px&opacityShadow=30&bgImgOpacityShadow=0&bgTextureShadow=flat&bgColorShadow=%23666666&opacityOverlay=30&bgImgOpacityOverlay=0&bgTextureOverlay=flat&bgColorOverlay=%23aaaaaa&iconColorError=%23cc0000&fcError=%235f3f3f&borderColorError=%23f1a899&bgTextureError=flat&bgColorError=%23fddfdf&iconColorHighlight=%2377762O&fcHighlight=%2377762O&borderColorHighlight=%23dad55e&bgTextureHighlight=flat&bgColorHighlight=%23fffa90&iconColorActive=%23ffffff&fcActive=%23ffffff&borderColorActive=%2396b0b6&bgTextureActive=flat&bgColorActive=%2396b0b6&iconColorHover=%23555555&fcHover=%232b2b2b&borderColorHover=%23cccccc&bgTextureHover=flat&bgColorHover=%23c1d5da&iconColorDefault=%23777777&fcDefault=%23454545&borderColorDefault=%23c5c5c5&bgTextureDefault=flat&bgColorDefault=%23B5D0D7&iconColorContent=%23444444&fcContent=%23333333&borderColorContent=%23dddddd&bgTextureContent=flat&bgColorContent=%23e6f4fa&iconColorHeader=%23444444&fcHeader=%23333333&borderColorHeader=%23dddddd&bgTextureHeader=flat&bgColorHeader=%2396b0b6&cornerRadius=8px&fwDefault=normal&fsDefault=1em&ffDefault=Arial%2CHelvetica%2Csans-serif
```

2.2.3 Dialog widget

v22.Gui, through the Gui.Dialog_Popup procedure, uses the Dialog widget.

https://api.jqueryui.com/dialog

https://api.jqueryui.com/position

3 UXString

Paragraphs below are from the UXString author:
- https://github.com/Blady-Com/UXStrings
- compl.land.ada [ANN] Release of UXStrings 0.5.0 – 230505 at 05:23

3.1 Introduction

v22 adopts UXString v3 implementation using Unbounded_Wide_Wide_Strings for internal representation. With this latter implementation, the characters are stored in Unicode form and the management of dynamic size uses the standard Wide_Wide_Unbounded strings library.

Performance with Gnoga is better. UXStrings2 already brought better performance in the case of strings only made up of ASCII characters [improvement by a factor 2 to 3 compared to UXStrings1]. With UXStrings3 performance in the latter case is still improved [factor 6 to 7 compared to UXStrings1] moreover in the case of strings accentuated in French and strings containing emojis the process times are also improved [factor 7 to 8 by compared to UXStrings1 or even more in the case of emo-

jis). For all cases, the global memory occupation of the Gnoga application is generally similar.

3.2 Motivation

My first motivation was to avoid the user of the Ada language from having to make a choice in the representation of character strings. With the current Ada 2012 standard, the choice must be made according to the nature of the characters handled (Character, Wide_Character or Wide_Wide_Character) and the adaptation of the string size according to the operations carried out. Moreover, depending on the libraries used, making a single choice is generally not possible, which leads to continuous conversions.

Ada GUI library Gnoga internal character strings implementation is based on both Ada types String and Unbounded_String. The native Ada String encoding is Latin-1 whereas transactions with the Javascript part are in UTF-8 encoding.

Some drawbacks come up, for instance, with internationalization of programs (see Localize Gnoga demo):

- Several conversions between String and Unbounded_String objects it isn't usable out of Latin-1 character set, characters out of Latin-1 set are blanked;
- Continuous conversions between Latin-1 and UTF-8, each sent and received transaction between Ada and Javascript parts.

Two ways of possible improvement for native Ada String: dynamic length handling and Unicode support.

3.3 Definitions

```
Supported encoding schemes

type Encoding_Scheme is (ASCII_7, Latin_1, UTF_8, UTF_16BE, UTF_16LE);

Supported UTF-16 encoding schemes

subtype UTF_16_Encoding_Scheme is Encoding_Scheme range UTF_16BE .. UTF_16LE;

Characters in ISO/IEC 646

subtype ASCII_Character is Ada.Characters.Handling.ISO_646;
subtype ASCII_Character_Array is String with
      Dynamic_Predicate => (for all Item of ASCII_Character_Array => Item in ASCII_Character);

Characters in ISO/IEC 8859-1

subtype Latin_1_Character is Character;
subtype Latin_1_Character_Array is String;

Characters in Unicode Basic Multilingual Plane

subtype BMP_Character is Wide_Character;
subtype BMP_Character_Array is Wide_String;

Characters in Unicode planes

subtype Unicode_Character is Wide_Wide_Character;
subtype Unicode_Character_Array is Wide_Wide_String;

Array of 8 bits values representing UTF encodings (UTF-8, UTF-16BE, or UTF-16LE)

subtype UTF_8_Character_Array is Ada.Strings.UTF_Encoding.UTF_String;
subtype UTF_16_Character_Array is Ada.Strings.UTF_Encoding.UTF_String;
```

```
Container type of Unicode characters with dynamic size usually named string

type UXString is tagged private with
   Constant_Indexing => Element,
   Iterable => (First => First, Next => Next, Has_Element => Has_Element, Element => Element),
   String_Literal => From_Unicode;
```

3.4 Workarounds

First possibility is using UTF-8 as internal implementation in Unbounded_String objects. The simplest way but Gnoga uses many times character indexes to parse Javascript messages that is not easy to achieved with UTF-8 which may have several lengths to represent one character. String parsing will be time consuming. Some combinations may lead to incorrect UTF-8 representation.

Second possibility is to use Unbounded_Wide_String or Unbounded_Wide_Wide_String. Using Unbounded_Wide_String is quite being in the middle of the river might as well use Unbounded_Wide_Wide_String. In this latter case the memory penalty is heavy for only few accentuated character occurrences. So back to Unbounded_Wide_String but you'll miss the so essential emojis ;-)

Third possibility is to make no choice between Latin-1, Wide and Wide_Wide characters. The object shall adapt its inner implementation to the actual content. For instance with English language the most often use case will be Latin-1 inner implementation, for French language the most often will be Latin-1 with some exceptions with BMP (Unicode Basic Multilingual Plane) implementation such as in "cœur", for Greek language the most often will be BMP implementation. The programmer won't make any representation choice when for example receiving UTF-8 messages:

```
   S2 : UXString;
   ...
   S2 := "Received: " & From_UTF_8 (Message);
```

Automatically S2 will adapt its inner representation to the received characters.
UXStrings packages

Package named UXStrings (Unicode Extended Strings) and its Text_IO child package are proposed to bring String enhancements using some Ada 2022 features.

The first part of UXStrings package contains renaming statements of current Ada types. Ada current String type is structurally an array of Latin-1 characters thus is renamed as Latin_1_Character_Array. And so on.

The second part defines the USXString type as a tagged private type which has got aspects such as Constant_Indexing, Variable_Indexing, Iterable and String_Literal, so we can write:

```
   S1, S2, S3 : UXString;
   C          : Character;
   WC         : Wide_Character;
   WWC        : Wide_Wide_Character;
   ...
   S1 := "était blah blah";
```

```
   C   := S1 (3);
   WC  := S1 (2);
   WWC := S1 (1);
   S1 (3) := WWC;
   S1 (2) := WC;
   S1 (1) := C;
   S3  := "une soirée passée à étudier les mathématiques ℕℂ𝕂...";
   for I in S3 loop
      C   := S3 (I);
      WC  := S3 (I);
      WWC := S3 (I);
       Put_Line (Character'pos (C)'img & Wide_Character'pos (WC)'img & Wide_Wide_Char-
acter'pos (WWC)'img);
   end loop;
```

The third part defines conversion functions between UXString and various encoding such as Latin-1, BMP (USC-2), Unicode (USC-4), UTF-8 or UTF-16, so we can write:

```
   S1  := From_Latin_1 ("blah blah");
   S2  := From_BMP ("une soirée passée à étudier la physique ω=Δθ/Δt...");
   S3  := From_Unicode ("une soirée passée à étudier les mathématiques ℕℂ𝕂...");
   Send (To_UTF_8 (S1) & To_UTF_8 (S3));
```

The fourth part defines various API coming from Unbounded_String such as Append, "&", Slice, "=", Index and so on.

Note: Iterable is a GNAT specific aspect.

3.5 Some thoughts

I've preferred that the API of legacy Ada "string" types to be closed to those of Ada library so that the adaptation would be easy. Note that I've renamed them as character arrays rather than strings in order to accentuate the semantic difference.

Apart from complex implementations, if you want to access a specific position you have to parse from the beginning of the UTF-8 data as UXStrings1 does. UXStrings2 always computes if the resulting data are all ASCII, so the access is then direct. UXStrings3 is internally like an Unicode array, so the access is direct.

Examples

Investment in C programs reliability will increase up to exceed the probable cost of errors or until someone insists on recoding everything in Ada.
Gilb's laws synthesis

1 TestApi

TestApi is a simple console program which shows the main features of v22 packages. TestApi is automatically build when building v22.

2 TestGui

TestGui is a simple web program which mainly shows some features of v22.Gui and v22.Sql. TestApi is automatically build when building v22.

When launching for the first time,

```
20231031-172139 - INIT    - MSG - v22 Framework - GUI test program
20231031-172139 - INIT    - MSG - Copyright [C] Sowebio SARL 2020-2023 under LGPLv3
20231031-172139 - INIT    - MSG - testgui v0.2 - v22 v0.1 - build 2023-10-31 17:21:16
```

TestGui creates an already set testgui.cfg to create a standalone SQLite database.

```
20231031-172139 - INIT    - MSG - TestGui.Init.App > Configuration file testgui.cfg has been created.
20231031-172139 - INIT    - MSG - TestGui.Init.App > Configuration file testgui.cfg loaded
```

Then GestGui also creates and populates a demo database named testgui.db.

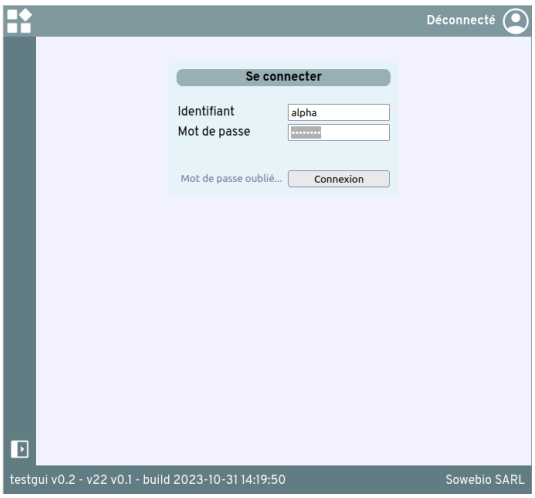
```
20231031-172139 - INIT    - MSG - Database testgui needs creation or update
20231031-172139 - INIT    - MSG - Create Table: Sys_Config - Create Column: Parameter VARCHAR(40) UNIQUE PRIMARY KEY
20231031-172139 - INIT    - MSG - Table: Sys_Config - Create Column: Value TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Config - Creating Index: Idx_Config_Parameter Parameter
20231031-172139 - INIT    - MSG - Create Table: Sys_Schema - Create Column: Table_Name VARCHAR(40)
20231031-172139 - INIT    - MSG - Table: Sys_Schema - Create Column: Column_Name VARCHAR(40)
20231031-172139 - INIT    - MSG - Table: Sys_Schema - Create Column: Column_Type TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Schema - Create Column: Column_Constraint TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Schema - Create Column: Version TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Schema - Create Column: Comment TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Schema - Creating Index: Idx_Schema_Table_Name Table_Name,Column_Name
20231031-172139 - INIT    - MSG - Create Table: Sys_Users - Create Column: Login VARCHAR(40) UNIQUE PRIMARY KEY
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: First_Name TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Last_Name TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Phone TEXT
```

```
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Email TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Password TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Grants TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Properties TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Language TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Time_Zone TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Theme TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Notes TEXT
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Created_On VARCHAR(15)
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Updated_On VARCHAR(15)
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Connection_Timeout INTEGER
20231031-172139 - INIT    - MSG - Table: Sys_Users - Create Column: Connection_Total INTEGER
20231031-172140 - INIT    - MSG - Table: Sys_Users - Create Column: Connection_Counter INTEGER
20231031-172140 - INIT    - MSG - Table: Sys_Users - Create Column: Connection_Info INTEGER
20231031-172140 - INIT    - MSG - Table: Sys_Users - Creating Index: Idx_User_Login Login UNIQUE
```

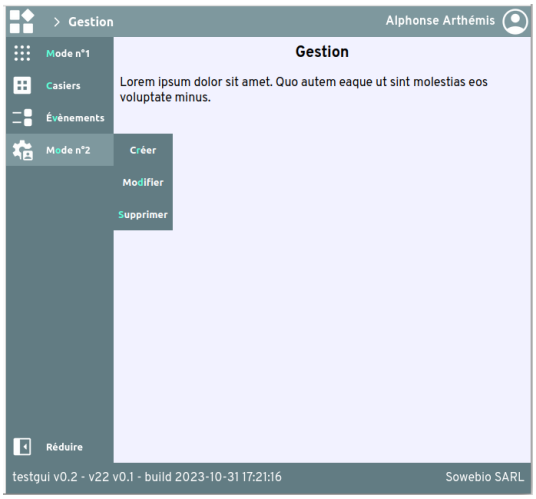
Finally, TestGui is ready to use and wait for its first connexion.

```
20231031-172140 - INIT    - MSG - TestGui.On_Connect > Starting Gnoga server
20231031-172140 - INIT    - MSG - TestGui.SQL_Ping > Armed for 3600s cycles
```

In your preferred browser, type localhost:8080 to connect to TestGui and type Identifier: alpha and Password: password to log in.

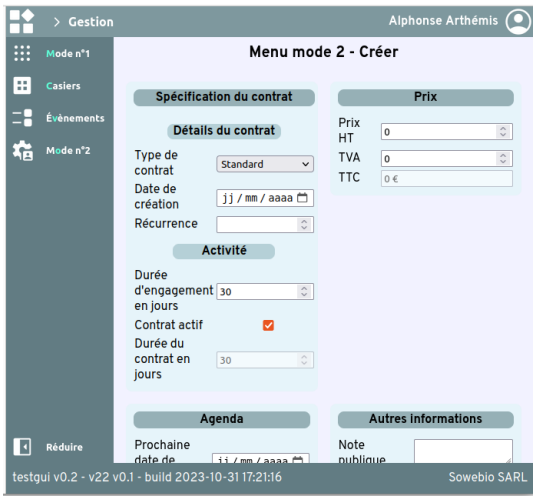


When logged, the main screen appears. Click on the icon at bottom left to expand the menu bar. Click on the User parameters icon (labelled Mode 2). A sub menu appears.



Then click on create to display a new two columns screen.

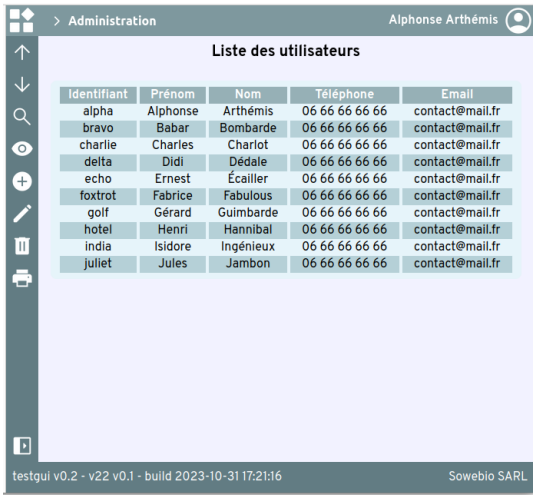
Test the drop-down menu and the calendar.



Adjust the screen width to test the responsiveness. TestGui will adapt and shrinking to one column.

Then click on the icon at the top left, then click on admin.

Click on the first icon in the menu bar (labelled Users).



Up arrow move the list backward. Down arrow move the list forward.

Click on an item list (the color change) and display the detailed view clicking on the magnifier or on the pen to update the item.

Click on the plus sign to create a new user or, after selected an item, click on the bin to delete it.

3 Event listener interfaces with Gnoga

Reproduction of ~/Gnoga/doc/articles/event_listerners.txt by Jeremiah Breeden.

An option that I have used is to create event listener interface. I extend the button class to include an event generator that maintains a list of listeners and I add an Add_Listener procedure so outside views can add themselves as listeners. I create a handler that Gnoga expects but have that handler operate the generator to fire the events out of the button class.

Then I extend whichever view I am using to implement the listening interface, override the create procedure, and have it call the Add_Listener procedure mentioned above. Then I just have to override the listener procedure that handles the event and I have both the parent view that is interested in the event and the button that sourced the original event.

It's a bit complex, but it allows me to have any structure I want be able to respond to any event just by adding itself as a listener to the event.

The big thing is Handle_On_Click gives you the containing view type which allows you to modify other controls using the on click from the button. You can also pass the on click event to other parent views who may want to modify their own layouts based on that button event. You would just need to add a public Add_On_Click_Listener (or whatever name) procedure to Listener_View.View_Type (or whatever view you create) and have it call the button's Add_On_Click_Listener procedure.

Button_For_Listeners is the extended class for Gnoga's Button_Type
Listener_View is just a test view type that would contain a Button among other controls.

button_for_listeners.ads

```
with Gnoga.Gui.Element.Common;
with Ada.Containers.Vectors;

package Button_For_Listeners is

   type Listener_Type is limited interface;

   type Button_Type is new Gnoga.Gui.Element.Common.Button_Type with private;

   procedure Add_On_Click_Listener
     (Self     : in out Button_Type;
      Listener : not null access Listener_Type'Class);

   procedure Handle_On_Click(Listener : in out Listener_Type;
                             Source   : in out Button_Type'Class) is abstract;


private
   type Listener_Class_Access is access all Listener_Type'Class;
   package Vectors is new Ada.Containers.Vectors(Natural, Listener_Class_Access);

   type Button_Type is new Gnoga.Gui.Element.Common.Button_Type with record
      On_Click_Listeners : Vectors.Vector;
   end record;

end Button_For_Listeners;
```

button_for_listeners.adb

```
with Gnoga.Gui.Base;

package body Button_For_Listeners is

   procedure Gnoga_On_Click(Object : in out Gnoga.Gui.Base.Base_Type'Class) is
      Cursor : Vectors.Cursor := Button_Type(Object).On_Click_Listeners.First;
   begin
      while Vectors.Has_Element(Cursor) loop
         Vectors.Element(Cursor).Handle_On_Click(Button_Type(Object));
         Cursor := Vectors.Next(Cursor);
      end loop;
   end Gnoga_On_Click;

   procedure Add_On_Click_Listener
     (Self     : in out Button_Type;
      Listener : not null access Listener_Type'Class) is
   begin
      Self.On_Click_Listeners.Append(Listener);
      Self.On_Click_Handler(Gnoga_On_Click'Access);
   end Add_On_Click_Listener;

end Button_For_Listeners;
```

listener_view.ads

```
with Gnoga.Gui.View;
with Gnoga.Gui.Base;
with Button_For_Listeners;

package Listener_View is
   package Buttons renames Button_For_Listeners;

   type View_Type is new Gnoga.Gui.View.View_Type
                         and Buttons.Listener_Type with private;

   overriding
   procedure Create
     (Self   : in out View_Type;
      Parent : in out Gnoga.Gui.Base.Base_Type'Class;
      ID     : in     String  := "");

private
   type Self_Ref_Type(Ref : access Buttons.Listener_Type'Class)
      is limited null record;

   type View_Type is new Gnoga.Gui.View.View_Type
                         and Buttons.Listener_Type with record
      Button : Buttons.Button_Type;
      Self_Reference : Self_Ref_Type(View_Type'Access);
   end record;

   overriding
   procedure Handle_On_Click(Listener : in out View_Type;
                             Source   : in out Buttons.Button_Type'Class);

end Listener_View;
```

listener_view.adb

```
package body Listener_View is
   procedure Create(Self   : in out View_Type;
                    Parent : in out Gnoga.Gui.Base.Base_Type'Class;
                    ID     : in     String  := "") is
   begin
      Gnoga.Gui.View.View_Type(Self).Create(Parent);
      Self.Button.Create(Self,"Test Button");
      Self.Button.Add_On_Click_Listener(Self.Self_Reference.Ref);
   end Create;

   Test_Count : Natural := 0;

   overriding
   procedure Handle_On_Click(Listener : in out View_Type;
                             Source   : in out Buttons.Button_Type'Class) is
   begin
      Listener.Button.Text("Testing: " & Integer'Image(Test_Count));
      Test_Count := Test_Count + 1;
   end Handle_On_Click;

end Listener_View;
```

Tools

Networks always go down on a Friday.
John Karr law

1 Icons making

1.1 Favicon

The favicon file should be a PNG 8 bits 32x32 pixels 300x300ppp in RGB color space and must be named favicon.ico (not .png) and located in ./html directory. This setup is compatible with Firefox, Chrome, Safari and derivatives.

1.2 Menus icons

Go to website https://fonts.google.com/icons

Settings:
- Fill = on
- Optical size = 48px

Clic on icon "x", download it in SVG, a vector size independent format, as x.svg
Repeat as needed

When finished, run ./cv (see below) in the download directory to convert theses svg icons in the proper v22 icon format (png, 128x128 px, transparent and negative):

```
Color: #ffffff
Size: 128x128
Density: 1200

Convert apps.svg > apps.png
Convert browse_activity.svg > browse_activity.png
Convert dataset.svg > dataset.png
Convert event_list.svg > event_list.png
Convert group.svg > group.png
Convert indeterminate_check_box.svg > indeterminate_check_box.png
Convert local_convenience_store.svg > local_convenience_store.png
Convert person.svg > person.png
Convert point_of_sale.svg > point_of_sale.png
Convert query_stats.svg > query_stats.png
Convert settings_account_box.svg > settings_account_box.png
Convert settings.svg > settings.png
```

```
Convert whatshot.svg > whatshot.png
```

- cv file

```
#!/bin/env bash

color="#ffffff"
size=128
density=1200

echo ""
echo "Color: $color"
echo "Size: ${size}x${size}"
echo "Density: $density"
echo ""

files="./*.svg"
for file in $files
do
  output="${file##*/}" # delete residual path [./]
  output="${output%.*}" # delete extension [.svg]
  echo "Convert $output.svg > $output.png"
   convert -background none -density $density -resize "$size"x"$size" $file /tmp/out-
put.png
  convert /tmp/output.png +level-colors $color, /tmp/output.png
  convert /tmp/output.png -fill $color -opaque white ico-$output.png
done

# eof
```

Gnoga allows you to leverage existing HTML tools to quickly design and create Forms for use in Gnoga applications.

Here is a simple step by step guide to quickly creating a bootstrap based Form using a simple online tool with Gnoga.

2 Form builder

Amended reproduction of ~/Gnoga/doc/articles/QuickandSimpleFormBuildingwith-Gnoga.pdf, author unknown.

2.1 Create form

http://bootsnipp.com/forms?version=3

Click on Form Name for the title that will be displayed above the form, for this Exam-ple: “Hello World Gnoga and Bootstrap”

Drag a “Text Input” on to the form with your mouse

Click on the Text Input and fill in:
- ID/Name : fname
- Label Text : First Name
- Placeholder : blank
- Help Text : Enter your first name
- Click required

- Leave input size at Medium
- Click Save

Drag a “Text Input” on to the form with your mouse under the First Name field

Click on the Text Input and fill in:
- ID/Name : lname
- Label Text : Last Name
- Placeholder : blank
- Help Text : Enter your last name
- Click requied
- Leave input size at Medium
- Click Save

Click on the “buttons” tab above the form elements

Drag a Single Button over under the two fields

Click on the Single Button and fill in:
- ID/Name : sbutton
- Label Text : blank
- Button Label : Ok
- Leave button type : Primary
- Click Save

2.1.1 Create Gnoga Project

Let’s create our skeleton project using gnoga_make:
- Run (add path if needed): gnoga_make new form_demo1 multi_connect
- Edit the path (if needed) to gnoga at form_demo1/src/form_demo1.gpr

Replace boot.html copy ~/gnoga/html/boot_starp3.html to replace boot.html from the html directory of your project.

From the css directory of your project, copy the bootstrap.min.css file from the bootstrap dist directory (see below).

From the js directory of your project, copy the bootstrap.min.js file from the bootstrap dist directory (see below).

Make the directory templates:
- mkdir templates

Copy the content of Rendered or View HTML tab to the body of form_demo1.html.

Edit our view spec form_demo1view.ads:
- Add with Gnoga.Gui.Element.Form;
- Replace the record contents of Default_View Type with:
- First_Name : Gnoga.Gui.Element.Form.Text_Type;
- Last_Name : Gnoga.Gui.Element.Form.Text_Type;

Edit the view form_demo1view.adb:

- In the procedure Create after the call to the parent Create add:
- Replace the Create procedure body with:
```
Gnoga.Gui.View.View_Type (View).Create (Parent, ID);
View.Load_File ("form_demo1.html");
View.First_Name.Attach_Using_Parent (View, "fname");
View.Last_Name.Attach_Using_Parent (View, "lname");
```

Edit the form_demo1controller.adb:
- Replace the On_Click with:
```
procedure On_Submit (Object : in out Gnoga.Gui.Base.Base_Type'Class);
procedure On_Submit (Object : in out Gnoga.Gui.Base.Base_Type'Class) is
View : form_demo1.View.Default_View_Type renames
form_demo1.View.Default_View_Type (Object);
begin
Gnoga.Log ("First Name : " & View.First_Name.Value);
Gnoga.Log ("Last Name : " & View.Last_Name.Value);end On_Submit;
```

- Replace the body of Default with:
```
View.Dynamic;
View.Create (Main_Window);
View.On_Submit_Handler (On_Submit'Access);
```

Now build the project by running make at the root dir of your project and open your browser to localhost:8080 and your done :)

The complete View and Controller files follow along with the form_demo1.html clip from the website:

```
with Gnoga.Gui.Base;
with Gnoga.Gui.View;
with Gnoga.Gui.Element.Form;

package form_demo1.View is

 type Default_View_Type is new Gnoga.Gui.View.View_Type with
 record
  First_Name : Gnoga.Gui.Element.Form.Text_Type;
  Last_Name : Gnoga.Gui.Element.Form.Text_Type;
 end record;

type Default_View_Access is access all Default_View_Type;
type Pointer_to_Default_View_Class is access all Default_View_Type'Class;

overriding procedure Create (View : in out Default_View_Type; Parent : in out Gnoga.Gui.Base.Base_Type'Class; ID : in String := "");

end form_demo1.View;
```

```
package body form_demo1.View is


  Create

 overridingprocedure Create (View : in out Default_View_Type; Parent : in out Gnoga.Gui.Base.Base_Type'Class; ID : in String := "") is
```

```
 begin
  Gnoga.Gui.View.View_Type (View).Create (Parent, ID);
  View.Load_File ("form_demo1.html");
  View.First_Name.Attach_Using_Parent (View, "fname");
  View.Last_Name.Attach_Using_Parent (View, "lname");
 end Create;

end form_demo1.View;
```

```
with Gnoga.Gui.Base;
with form_demo1.View;

package body form_demo1.Controller is

 procedure On_Submit (Object : in out Gnoga.Gui.Base.Base_Type'Class);
 procedure On_Submit (Object : in out Gnoga.Gui.Base.Base_Type'Class) is
  View : form_demo1.View.Default_View_Type renames
  form_demo1.View.Default_View_Type (Object);
  begin
   Gnoga.Log ("First Name : " & View.First_Name.Value);
   Gnoga.Log ("Last Name : " & View.Last_Name.Value);
  end On_Submit;

 procedure Default (Main_Window : in out Gnoga.Gui.Window.Window_Type'Class; Connec-
tion : access Gnoga.Application.Multi_Connect.Connection_Holder_Type) is
  View : form_demo1.View.Default_View_Access := new form_demo1.View.Default_View_Type;
 begin
  View.Dynamic;
  View.Create (Main_Window);
  View.On_Submit_Handler (On_Submit'Access);
 end Default; begin

 Gnoga.Application.Multi_Connect.On_Connect_Handler (Default'Access, "default");

end form_demo1.Controller;
```

2.1.2 templates/form_demo.html

```
<form class="formhorizontal">
<fieldset>

<! Form Name >
<legend>Hello World Gnoga and Bootsrap</legend>

<! Text input>
<div class="formgroup">
 <label class="colmd4 controllabel" for="fname">First Name</label>
  <div class="colmd4">
   <input id="fname" name="fname" type="text" placeholder="" class="formcontrol input-
md"required="">
  <span class="helpblock">Enter your first name</span>
 </div>
</div>

<! Text input>
<div class="formgroup">
 <label class="colmd4 controllabel" for="lname">Last Name</label>
  <div class="colmd4">
   <input id="lname" name="lname" type="text" placeholder="" class="formcontrol input-
md" required="">
   <span class="helpblock">Enter your last name</span>
```

```
 </div>
</div>

<! Button >
<div class="formgroup">
 <label class="colmd4 controllabel" for="sbutton"></label>
  <div class="colmd4">
   <button id="sbutton" name="sbutton" class="btn btnprimary">Ok</button>
  </div>
</div>

</fieldset>
</form>
```

2.1.3 Bootstrap-Form-Builder

https://github.com/minikomi/Bootstrap-Form-Builder using Bootstrap 2.3.1
http://bootsnipp.com/forms?version=3 using Bootstrap 2.3.1

2.1.4 Bootstrap library

https://getbootstrap.com

git clone https://github.com/twbs/bootstrap

git checkout tags/v2.3.2 -b v2.3.2-branch
Basculement sur la nouvelle branche 'v2.3.2-branch'

git checkout tags/v3.4.1 -b v3.4.1-branch
Basculement sur la nouvelle branche 'v3.4.1-branch'

3 Page builder

<<<TODO>>>

Evaluate the online page builder https://bootsnipp.com/builder.

Needs Bootstrap 4 for widgets.

https://this-page-intentionally-left-blank.org

API

There are 10 types of people in the world: those who understand binary and those who don't.
Anonymous

1 v22

1.1 Introduction

1.1.1 Concepts

The developer is a writer. The writer's courtesy is clarity;

- Clarity and ease of use are prioritized over speed and efficiency.

The performance of a compiled language such as Ada as well as the hardware capabilities of current systems justify – most of the times - these choices.

On a simple loop, let's recall that if HAC (a very valuable and helpful Ada subset interpreter) is (among others) 7 times faster and vastly more capable than Bash, HAC itself is 300 times slower than Ada. So... Loop to the grayed second line :)

1.1.2 Conventions

To ease developers:

- String type is a subtype of UXString. This new String UTF-8 subtype is the standard v22 type. You should always use String[6].

- All strings constants and function only returns String type.

- All strings parameters accept String type.

1.1.3 Usage

Use ./v22/v22.gpr as a stub for your own projects.

[6] Only use Standard.String when appropriate.

Use ./v22/example/ as an application template useful to create your own projects.

1.2 v22

Base package.

1.2.1 Finalize

- Description

  Finalize application, closing log file, SQL connections and restore cursor state.

  A handling_Type parameter is provide for Ctrl-C handling or exceptions.

- Usage

  procedure Finalize (Handling_Type : String := "")

- Example

```
```

1.2.2 Get_Build

- Description

  Returns the formatted build date stamp as: “build YYYY-mm-dd hh:mm:ss”.

  Date and time reflect those of the v22 main program object v22.o, which implies deleting v22.o at first before building the application.

- Usage

  function Get_Build return String

- Example

```
Log.Msg ["Date stamp build:  " & v22.Get_Build];
build 2021-08-04 14:36:27
```

1.2.3 Get_Compiler_Version

- Description

  Returns the Library name and formatted version: “<space>v.minor.major”.

- Usage

  function Get_Version return String

- Example

```
Log.Msg ["Compiler version: " & v22.Get_Compiler_Version];
gnat gcc v13.2.0
```

1.2.4 Get_Log_Dir

- Description

  Returns the log directory.

- Usage

  function Get_Log_Dir return String

- Example

```
Log.Msg ["v22.Get_Log_Dir];
/var/log/
```

1.2.5 Get_Tmp_Dir

- Description

  Returns the temporary files directory.

- Usage

  function Get_Tmp_Dir return String

- Example

```
Log.Msg ["v22.Get_Tmp_Dir];
/tmp/
```

1.2.6 Get_Version

- Description

  Returns the Library name and formatted version: “<space>v.minor.major”.

- Usage

  function Get_Version return String

- Example

```
```

```
Log.Msg ["Library version: " & v22.Get_Version];

v22 v0.6
```

1.2.7 Raise_Exception

- Description

  Raise an exception for reporting test and <program_Name.err> file creation.

  In addition to the usual trace, a v22 exception give some extra information like: exception time, program uptime, program & library names & versions, start & home directories and Ada and all languages memory allocation, current & maximum (peak) values.

- Usage

```
procedure Raise_Exception
```

- Example

```
Raise_Exception;

20230921-123354 - INIT    - MSG - Exception test trigered by a raise exception

--------------------------------------------------------------------------------

Exception time         : 20230921-123354
Program uptime         :  0h0 0m0 0s
Program build DT stamp : build 2023-09-21 12:33:15
Program name & version : testgui v0.1
Library name & version : v22 v0.1
Start directory        : /home/sr/Sowebio/informatique/dev/gpl/github/v22/tests
Home directory         : /home/sr
Ada mem. alloc. [bytes]: Ada Cur: [ 4132 ] Max: [ 4708 ]
All mem. alloc. [bytes]: All Cur: [ 15204352 ] Max: [ 15204352 ]

raised V22.RAISE_EXCEPTION.V22_EXCEPTION_TEST : v22.adb:81
[./testgui]
0x496f9e v22__raise_exception at v22.adb:81
0x41c1b1 _ada_testgui at testgui.adb:830
0x488d8f main at b__testgui.adb:1173
[/lib/x86_64-linux-gnu/libc.so.6]
0x7fb0fee29d8e
0x7fb0fee29e3e
[./testgui]
0x415973 _start at ???
0xfffffffffffffffe

--------------------------------------------------------------------------------

20230921-123354 - INIT    - MSG - testgui > Exception detected, finalize application

20230921-123354 - SQL T1  - MSG - Closing SQLITE database: v22_testapi_1
20230921-123354 - SQL T1  - MSG - Closing SQLITE database: v22_testapi_2
```

1.3 Cfg – Configuration file

Handle application configuration file. One configuration file is allowed per application.

1.3.1 Close

- Description

  Close application configuration file. For sanity only, as each setting is instantly flushed to disk.

- Usage

```
procedure Close
```

- Example

```

```

1.3.2 Comment

- Description

  Insert a comment Text to the next line.

- Usage

```
procedure Comment (Text : String)
```

- Example

```

```

1.3.3 Delete

- Description

  Delete parameter in section. If no other parameter in this section, delete section too. Avoid reserved chars *[ ] = #* inside parameters.

- Usage

```
procedure Delete (Section : String; Parameter : String)
```

- Example

```

```

1.3.4 Get

- Description

  Return parameter in section or empty string if not found. Avoid reserved chars *[ ] = #* inside parameters.

- Usage

```
function Get (Section : String; Parameter : String) return String
```

- Example

```

```

1.3.5 Get_Name

- Description

  Returns full qualified (with path) application configuration file name.

- Usage

```
function Get_Name return String
```

- Example

```

```

1.3.6 New_Line

- Description

  Insert a blank line after to the next line.

- Usage

```
procedure New_Line
```

- Example

```

```

1.3.7 Open

- Description

```
Open and load if exist application configuration file. Create blank if
non existent. Default configuration file name is full qualified program
name followed by .cfg extension and created in the program start direc-
tory. This default file name may be changed by Set_Name procedure.
```

- Usage

```
function Open (Cfg_File_Read_In : String := "") return Boolean
```

- Example

```

```

1.3.8 Set

- Description

  Create or replace an existing parameter in a section. If this latter does not exist, also creating it. New setting is persistent even program quits unexpectedly after. Avoid reserved chars *[ ] = #* inside parameters. If reserved chars are passed, the procedure does nothing. An optional trailing comment can also be added.

- Usage

```
procedure Set (Section : String;
               Parameter : String;
               Value : String;
               Comment : String := "")
```

- Example

```

```

1.3.9 Set_Name

- Description

  Set application configuration file name, relative or full qualified.

- Usage

```
procedure Set_Name (Cfg_File_Read_In : String)
```

- Example

```

```

1.4 Fls – Files management

1.4.1 Backup_File

- Description

  Rename file with .bak.n suffix. Iterate n=0..9 searching a free n bak file. If n is free then write .bak.n, if n=9, delete .bak.0

- Usage

```
procedure Backup_File (File_To_Backup : String);
```

- Example

```

```

1.4.2 Copy_File

- Description

  Copy a Source_Name file to a Target_Name file destination.
  Copy_Form is “preserve=all_attributes,mode=overwrite” (full attributes preservation and overwrite file if exists).

- Usage

```
procedure Copy_File (Source_Name, Target_Name : String)
```

- Example

```

```

1.4.3 Create_Directory_Tree

- Description

  Create a directory tree Dir_Tree. Each non-existent directory named by Dir_Tree is created (possibly including other intermediate directories). Return False if operation is unsuccessful (i.e. if base directory tree is inconsistent or still don't exist after the creating attempt). Return True if directory tree already exists or has just been created.

  Extra inner slashes are processed i.e. a directory like /home/sr/opt/ytr.lkj/////kjghgh will be valid. and will create, from /home/sr/opt :
  - Directory ytr.lkj
  - And then inner directory kjghgh

- Usage

```
function Create_Directory_Tree (Dir_Tree : String) return Boolean
```

- Example

```

```

1.4.4 Delete_Directory_Tree

- Description

  Delete a directory tree Dir_Tree. The directory and all of its contents (possibly including other directories) are deleted. Return True if Dir_Tree is successfully deleted or was already deleted. Return False if operation is unsuccessful (i.e. if base directory tree was non existent or still exists after the deleting attempt).

  Dir_Tree must be fully qualified, i.e. starting with a slash (/).

  This function prevents deletion of the following root directories: bin, boot, dev, etc, home, lib, lib32, lib64, libx32, lost+found, media, mnt, opt, proc, root, run, sbin, srv,

sys, tmp, usr, var. Pay close attention, you can't delete /etc but you are allowed to delete /etc/network !

/! With programs ran with root rights, this routine should be used with infinite caution.

/! This function uses Ada.Directories.Delete_Tree, which raises an exception if the directory tree to delete contains a *broken* symbolic link (a file like any other). This latter is seen as *non-existent* and, when the parent directory is deleted, an exception occurs : raised ADA.IO_EXCEPTIONS.USE_ERROR : directory tree rooted at <directory tree> could not be deleted (because *not empty*). Funny, but not so much. Pure C code problem in Ada RTS. Stacked C calls in Russian puppet mode until a logical problem arises.

- Usage

```
function Delete_Directory_Tree (Dir_Tree : String) return Boolean

procedure Delete_Directory_Tree (Dir_Tree : String)
```

- Example

```

```

1.4.5 Delete_File

- Description

Delete a Name file only if this latter exists. No exception will be raised if the file to delete does not exists.

- Usage

```
procedure Delete_File (Name : String)
```

- Example

```

```

1.4.6 Delete_Lines

- Description

Search and remove file lines matching Pattern in File_Name.

- Usage

```
procedure Delete_Lines (File_Name, Pattern : String)
```

- Example

```

```

1.4.7 Download_File

- Description

  Download a file from Url to Dlfile. Do nothing if Dlfile already exists with its size equals Dlsize. Name is purely informational and used to named file in text messages.

  Return True is Dlfile present at the right size, False otherwise.

- Usage

```
function Download_File (Url : Sting;
                        Dlfile : VString;
                        Name : String;
                        Dlsize : Integer := 0) return Boolean;
```

- Example

```
```

1.4.8 Exists

- Description

  Returns True if file or directory Name exists.

- Usage

```
function Exists (Name : String) return Boolean
```

- Example

```
```

1.4.9 Extract_Directory

- Description

  Returns directory from Name.

- Usage

```
function Extract_Directory (Name : String) return String
```

- Example

```
Tio.Put_Line (Extract_Directory ("/etc/ssh/sshd_config"))

/etc/ssh
```

1.4.10 Extract_Name

- Description

  Returns filename from Name.

- Usage

```
function Extract_Filename (Name : String) return VString
```

- Example

```
Tio.Put_Line (Fls.Extract_Filename ("/etc/ssh/sshd_config")) then
sshd_config
```

1.4.11 File_Size

- Description

  Return size of Name file.

- Usage

```
function File_Size (Name : String) return Integer
```

- Example

```

```

1.4.12 Get_Directory

- Description

  Returns current directory.

- Usage

```
function Current_Directory return String
```

- Example

```

```

1.4.13 Is_Root_Directory

- Description

  This function checks the following root directories: bin, boot, dev, etc, home, lib, lib32, lib64, libx32, lost+found, media, mnt, opt, proc, root, run, sbin, srv, sys, tmp, usr, var. Returns True if Dir_Tree is a root directory.

Dir_Tree must be fully qualified, ie starting with a slash (/).

- Usage

```
function Is_Root_Directory (Dir_Tree : String) return Boolean
```

- Example

```
Tio.Put_Line (Fls.Is_Root_Directory ("/etc"));
True

Tio.Put_Line (Fls.Is_Root_Directory ("/etc/network"));
False
```

1.4.14 Move_File

- Description

Move a Source_Name file to a Target_Name file destination. Copy_Form is "preserve=all_attributes,mode=overwrite" (full attributes preservation and overwrite file if exists).

- Usage

```
procedure Move_File (Source_Name, Target_Name : String)
```

- Example

```

```

1.4.15 Rename

- Description

Rename an Old_Name file or directory to a New_Name file or directory. If exists a file New_File, it will be overwritten.

- Usage

```
procedure Rename (Old_Name, New_Name : String)
```

- Example

```

```

1.4.16 Search_Lines

- Description

Search at least a line matching Pattern in File_Name and return true if found.

- Usage

```
function Search_Lines (File_Name, Pattern : String) return Boolean
```

- Example

```

```

1.4.17 Set_Directory

- Description

  Change to a directory Directory. Create Directory if this latter does not exist, return False if operation failed.

- Usage

```
function Set_Directory (Directory : String) return Boolean
```

- Example

```

```

1.5 Gui – Gnoga User Interface

This is the v22 Graphic User Interface, called Gnoga instead of Graphic in homage to the outstanding work of David Botton, not forgetting Dmitri A. Kazakov (Gnoga uses his Simple Components package and Dmitry helped implement the Socket Web interface) and Pascal Pignard, Gnoga's current maintainer, who also integrated his UXStrings library that made Gnoga UTF-8 compatible.

1.5.1 Close_Dialog

- Description

  Close current jQuery dialog, removes it from HTML.

- Usage

```
procedure Close_Dialog (Object : in out GGB.Base_Type'Class)
```

- Example

```

```

- Description

  Removes any content inside content parent.

- Usage

```
procedure Content_Clear (Object : in out GGB.Base_Type'Class)
```

- Example

```
```

1.5.2 Connection_Data_Clear

- Description

  Clear all Connection_Data entries. Should be used when user disconnects.

  Connection_Data is a free to use dictionary unique to each connection.

- Usage

```
procedure Connection_Data_Clear (Object : in out GGB.Base_Type'Class)
```

- Example

```
```

1.5.3 Connection_Data_Delete

- Description

  Delete a Connection_Data entry. Key not found is silently ignored.

  Connection_Data is a free to use dictionary unique to each connection.

- Usage

```
procedure Connection_Data_Delete (Object : in out GGB.Base_Type'Class;
Key : String)
```

- Example

```
```

1.5.4 Connection_Data_Get

- Description

  Retrieve a Connection_Data entry. Returns an empty string if key does not exist.

  Connection_Data is a free to use dictionary unique to each connection.

- Usage

```
function Connection_Data_Get (Object : in out GGB.Base_Type'Class; Key :
String) return  String;
```

- Example

```
```

1.5.5 Connection_Data_List

- Description

  List all Connection_Data entries on debug log.

  Connection_Data is a free to use dictionary unique to each connection.

- Usage

```
procedure Connection_Data_Delete (Object : in out GGB.Base_Type'Class)
```

- Example

```
```

1.5.6 Connection_Data_Set

- Description

  Create a new Connection_Data entry. If Key already exists, the value is replaced.

  Connection_Data is a free to use dictionary unique to each connection.

- Usage

```
procedure Connection_Data_Set (Object : in out GGB.Base_Type'Class; Key :
String; Value : String)
```

- Example

```
```

1.5.7 Content_Clear_HTML

- Description

  Clear HTML in content parent.

- Usage

```
procedure Content_Clear_HTML (Object : in out GGB.Base_Type'Class)
```

- Example

```

Content_Clear_Text
```

- Description

  Clear text in content parent. Should be used when the only content of the menu is this text. Otherwise, should use Content_Parent instead.

- Usage

```
procedure Content_Clear_Text (Object : in out GGB.Base_Type'Class)
```

- Example

```

```

1.5.8 Content_Clear_Title

- Description

  Clear title above content.

- Usage

```
procedure Content_Clear_Title (Object : in out GGB.Base_Type'Class)
```

- Example

```

```

1.5.9 Content_Group_Add_Button

- Description

  Add a button with a callback. Can be use as a submit button.

- Usage

```
procedure Content_Group_Add_Button (Object : in out GGB.Base_Type'Class;
                                    Text : String;
                                    On_Click : GGB.Action_Event;
                                    Parent_Key : String)
```

- Example

```

```

1.5.10 Content_Group_Add_Space

- Description

  Add space between rows in form group.

- Usage

```
procedure Content_Group_Add_Space (Object : in out GGB.Base_Type'Class;
                                   Parent_Key : String;
                                   Height : Integer := 8)
```

- Example

```

```

1.5.11 Content_Group_Add_Title

- Description

  Add a title to a form group.

  Title from arguments of Content_Group_Create is already on screen and has a higher emphasis.

- Usage

```
procedure Content_Group_Add_Title (Object : in out GGB.Base_Type'Class;
                                   Title : String;
                                   Parent_Key : String)
```

- Example

```

```

1.5.12 Content_Group_Check_Box_Add

- Description

  Add a check box in a group.

- Usage

```
procedure Content_Group_Check_Box_Add (Object : in out
GGB.Base_Type'Class; Name : String; Parent_Key : String;
On_Change : GGB.Action_Event := null)
```

- Example

```

```

1.5.13 Content_Group_Check_Box_Checked

- Description

  Set a check box state in a group.

- Usage

```
procedure Content_Group_Check_Box_Checked [Object : in out
GGB.Base_Type'Class; Name : String; Is_Checked : Boolean]
```

- Example

```

```

1.5.14 Content_Group_Check_Box_Is_Checked

- Description

  Get a check box state in a group.

- Usage

```
function Content_Group_Check_Box_Is_Checked [Object : in out
GGB.Base_Type'Class; Name : String] return Boolean
```

- Example

```

```

1.5.15 Content_Group_Create

- Description

  Create a container for forms. This element is displayed in the content container.

  Does not need any clearing apart from Content_Clear to delete this element.

- Usage

```
procedure Content_Group_Create [Object : in out GGB.Base_Type'Class;
                                Title : String]
```

- Example

```

```

1.5.16 Content_Group_Date_Add

- Description

  Add a date in a group.

- Usage

```
procedure Content_Group_Date_Add [Object : in out GGB.Base_Type'Class;
Name : String; Parent_Key : String; On_Change : GGB.Action_Event := null]
```

- Example

```

```

1.5.17 Content_Group_Date_Get

- Description

  Get a date in a group. Format is YYYY-MM-DD.

- Usage

```
function Content_Group_Date_Get (Object : in out GGB.Base_Type'Class;
Name : String) return String
```

- Example

```

```

1.5.18 Content_Group_Date_Set

- Description

  Set a date in a group. Format is YYYY-MM-DD.

- Usage

```
procedure Content_Group_Date_Set (Object : in out GGB.Base_Type'Class;
Name : String; Date : String)
```

- Example

```

```

1.5.19 Content_Group_Drop_Down_Menu_Add

- Description

  Add a drop-down menu, with customizable default item, in a group.

- Usage

```
procedure Content_Group_Drop_Down_Menu_Add (Object : in out
GGB.Base_Type'Class; Name : String; Parent_Key : String; On_Change : GG-
B.Action_Event := null)
```

- Example

```

```

1.5.20 Content_Group_Drop_Down_Menu_Add_Option

- Description

  Add an item to a drop-down menu. If Enabled is True, this item will be the displayed by default.

- Usage

```
procedure Content_Group_Drop_Down_Menu_Add_Option [
                                    Object : in out GGB.Base_Type'Class;
                                    Name : String;
                                    Option : String;
                                    Enabled : Boolean := False]
```

- Example

```

```

1.5.21 Content_Group_Drop_Down_Menu_Add_Option_From_DB_By_Key

- Description

  Registers all Column_Key and Column_Display lines from Table and
  select by default (for further display) the one where Column_Key = Match_Value.

  If No_Display_Raw_One is True and Column_Key equal to Id (primary key) then the first raw (Id=1) will be ignored (thus not displayed).

- Usage

```
procedure Content_Group_Drop_Down_Menu_Add_Option_From_DB_By_Key [
                                Object : in out GGB.Base_Type'Class;
                                Key : String;
                                DB : in out GSD.Connection'Class;
                                Table : String;
                                Column_Key : String;
                                Column_Display : String;
                                Match_Value : String := "";
                                No_Display_Raw_One : Boolean := False];
```

- Example

```

```

1.5.22 Content_Group_Drop_Down_Menu_Empty_Options

- Description

  Clear items in a drop-down menu.

- Usage

```
procedure Content_Group_Drop_Down_Menu_Empty_Options (Object : in out GG-
B.Base_Type'Class; Name : String)
```

- Example

```

```

1.5.23 Content_Group_Drop_Down_Menu_Get

- Description

  Get the selected item in a drop-down menu.

- Usage

```
function Content_Group_Drop_Down_Menu_Get (Object : in out
GGB.Base_Type'Class; Name : String) return String
```

- Example

```

```

1.5.24 Content_Group_Drop_Down_Menu_Get_Key_From_DB

- Description

  Returns the Column_Key from Table where Column_Display = Selected value from Key drop down menu.

- Usage

```
function Content_Group_Drop_Down_Menu_Get_Key_From_DB (
                                Object : in out GGB.Base_Type'Class;
                                Key : String;
                                DB : in out GSD.Connection'Class;
                                Table : String;
                                Column_Key : String;
                                Column_Display : String) return String
```

- Example

```

```

1.5.25 Content_Group_Drop_Down_Menu_Set_Selected

- Description

  Set item selected or deselected in a drop-down menu.

- Usage

```
procedure Content_Group_Drop_Down_Menu_Set_Selected (Object : in out GGB.Base_Type'Class; Name : String; Index : in Positive; Value : in Boolean := True)
```

- Example

```

```

1.5.26 Content_Group_Email_Add

- Description

  Add an email in a group.

- Usage

```
procedure Content_Group_Email_Add (Object : in out GGB.Base_Type'Class; Name : String; Parent_Key : String; On_Change : GGB.Action_Event := null)
```

- Example

```

```

1.5.27 Content_Group_Email_Get

- Description

  Get an email in a group.

- Usage

```
function Content_Group_Email_Get (Object : in out GGB.Base_Type'Class; Name : String) return String
```

- Example

```

```

1.5.28 Content_Group_Email_Set

- Description

  Set an email in a group.

- Usage

```
procedure Content_Group_Email_Set (Object : in out GGB.Base_Type'Class; Name : String; Email : String)
```

- Example

```

```

1.5.29 Content_Group_Item_Lock

- Description

  Lock a form in a form group so the user can't write anything.

- Usage

```
procedure Content_Group_Item_Lock (Object : in out GGB.Base_Type'Class;
                                   Name : String)
```

- Example

```

```

1.5.30 Content_Group_Item_Unlock

- Description

  Unlock a form in a form group.

- Usage

```
procedure Content_Group_Item_Unlock (Object : in out GGB.Base_Type'Class;
                                     Name : String)
```

- Example

```

```

1.5.31 Content_Group_Item_Place_Holder

- Description

  Set a place-holder for field-type forms.

- Usage

```
procedure Content_Group_Item_Place_Holder (
                                     Object : in out GGB.Base_Type'Class;
                                     Name : String;
                                     Place_Holder : String)
```

- Example

```

```

1.5.32 Content_Group_Number_Add

- Description

  Add a number in a group.

  To avoid “exception Error jQuery_Execute converting to Integer [forced to 0]”, all uninitialized Gui.Content_Group_Number_Add must be initialized to 0 with Gui.Content_Group_Number_Set.

- Usage

```
procedure Content_Group_Number_Add [Object : in out GGB.Base_Type'Class;
Name : String; Parent_Key : String; On_Change : GGB.Action_Event := null]
```

- Example

```

```

1.5.33 Content_Group_Number_Get

- Description

  Get a number in a group.

- Usage

```
function Content_Group_Number_Get [Object : in out GGB.Base_Type'Class;
Name : String] return Integer
```

- Example

```

```

1.5.34 Content_Group_Number_Set

- Description

  Set a number in a group.

- Usage

```
procedure Content_Group_Number_Set [Object : in out GGB.Base_Type'Class;
Name : String; Value : Integer]
```

- Example

```

```

1.5.35 Content_Group_Password_Add

- Description

  Add a password in a group.

- Usage

```
procedure Content_Group_Password_Add (Object : in out GGB.Base_Type'Class; Name : String; Parent_Key : String; On_Change : GGB.Action_Event := null)
```

- Example

```
```

1.5.36 Content_Group_Password_Get

- Description

  Get a password in a group.

- Usage

```
function Content_Group_Password_Get (Object : in out GGB.Base_Type'Class; Name : String) return String
```

- Example

```
```

1.5.37 Content_Group_Password_Set

- Description

  Set a password in a group to simulate an existing password in update form.

  It is advisable to spend a fairly long fake password, like 20 characters.

- Usage

```
procedure Content_Group_Password_Set (
                                      Object : in out GGB.Base_Type'Class;
                                      Name : String;
                                      Password : String)
```

- Example

```
Gui.Content_Group_Password_Set (Object, "Password", 20 * "-");
```

1.5.38 Content_Group_Phone_Add

- Description

  Add a phone number in a group.

- Usage

```
procedure Content_Group_Phone_Add (Object : in out GGB.Base_Type'Class;
Name : String; Parent_Key : String; On_Change : GGB.Action_Event := null)
```

- Example

```

```

1.5.39 Content_Group_Phone_Get

- Description

  Get a phone number in a group.

- Usage

```
function Content_Group_Phone_Get (Object : in out GGB.Base_Type'Class;
Name : String) return  String
```

- Example

```

```

1.5.40 Content_Group_Phone_Set

- Description

  Set a phone number in a group.

- Usage

```
procedure Content_Group_Phone_Set (Object : in out GGB.Base_Type'Class;
Name : String; Phone : String)
```

- Example

```

```

1.5.41 Content_Group_Text_Add

- Description

  Add text entry in a group.

- Usage

```
procedure Content_Group_Text_Add (Object : in out GGB.Base_Type'Class;
Name : String; Parent_Key : String; On_Change : GGB.Action_Event := null)
```

- Example

```

```

1.5.42 Content_Group_Text_Get

- Description

  Get text entry in a group.

- Usage

```
function Content_Group_Text_Get (Object : in out GGB.Base_Type'Class;
Name : String) return String;
```

- Example

```

```

1.5.43 Content_Group_Text_Set

- Description

  Set text entry in a group.

- Usage

```
procedure Content_Group_Text_Set (Object : in out GGB.Base_Type'Class;
Name : String; Text : String)
```

- Example

```

```

1.5.44 Content_Group_Text_Area_Add

- Description

  Add text area with customizable height in a group.

- Usage

```
procedure Content_Group_Text_Area_Add (Object : in out
GGB.Base_Type'Class; Name : String; Parent_Key : String; On_Change : GG-
B.Action_Event := null)
```

- Example

```
```

1.5.45 Content_Group_Text_Area_Get

- Description

  Get text area content in a group.

- Usage

```
function Content_Group_Text_Area_Get (Object : in out GGB.Base_Type'Class; Name : String) return String
```

- Example

```
```

1.5.46 Content_Group_Text_Area_Set

- Description

  Set text area content in a group.

- Usage

```
procedure Content_Group_Text_Area_Set (Object : in out GGB.Base_Type'Class; Name : String; Text : String)
```

- Example

```
```

1.5.47 Content_Group_Warning_Add

- Description

  Add a custom (non form) warning area in a group to display a warning message.

  Can be used with an empty Text at first, then later using the setter.

- Usage

```
procedure Content_Group_Warning_Add (Object : in out GGB.Base_Type'Class; Text : String; Key : String; Parent_Key : String)
```

- Example

```
```

1.5.48 Content_Group_Warning_Set

- Description

  Display a warning message in a group.

- Usage

```
procedure Content_Group_Warning_Set (Object : in out GGB.Base_Type'Class;
Key : String; Text : String)
```

- Example

```

```

1.5.49 Content_List_Add_Column

- Description

  Add a column in the table.

  Must be called as much as needed before using Content_List_Add_Item.

  See Content List layout example.

- Usage

```
procedure Content_List_Add_Column (Object : in out GGB.Base_Type'Class;
Variable : String; Parent_Key : String
```

- Example

```

```

1.5.50 Content_List_Add_Item

- Description

  Returns the item index in list.

  Columns must be added before using Content_List_Add_Column.

  Style controls the row text alignment (left, center, right). Center is the default.

  See Content List layout example.

- Usage

```
function Content_List_Add_Item (Object : in out GGB.Base_Type'Class;
                                Parent_Key : String;
                                Row_Index : Integer;
                                Style : String := "") return Integer;
```

- Example

```
```

1.5.51 Content_List_Add_Text

- Description

  This function place a new column in the table at a given row.

  The corresponding item must be created before using Content_List_Add_Item.

  Style controls the column text alignment (left, center, right). Center is the default.

  See Content List layout example.

- Usage

```
procedure Content_List_Add_Text (Object : in out GGB.Base_Type'Class;
                                 Value : String;
                                 Index : Integer;
                                 Parent_Key : String
                                 Style : String := "")
```

- Example

```
```

1.5.52 Content_List_Create

- Description

  Create a table for a list of items. This element is displayed in the content container.

  Does not need any clearing apart from Content_Clear to delete this element.

  Elements can be selected and the index obtained using Content_List_Selected_Row.

  See Content List layout example.

- Usage

```
procedure Content_List_Create (Object : in out GGB.Base_Type'Class; Title
: String)
```

- Example

```
```

1.5.53 Content_List_Selected_Row

- Description

  Return the selected row index.

  See Content List layout example.

- Usage

```
function Content_List_Selected_Row (Object : in out GGB.Base_Type'Class;
Parent_Key : String) return Integer
```

- Example

```

```

1.5.54 Content_Load_HTML

- Description

  Displays a HTML encoded file.

- Usage

```
procedure Content_Load_HTML (Object : in out GGB.Base_Type'Class; Text :
String)
```

- Example

```

```

1.5.55 Content_Parent

- Description

  Return the element below menu title, containing all content.

- Usage

```
function Content_Parent (Object : in out GGB.Base_Type'Class) return
GGV.View_Access
```

- Example

```

```

1.5.56 Content_Put_HTML

- Description

  Displays a HTML encoded string.

- Usage

```
procedure Content_Put_HTML (Object : in out GGB.Base_Type'Class; Text :
String)
```

- Example

```

```

1.5.57 Content_Put_Text

- Description

  Put text in content parent. Should be used when the only content of the menu is this text. Otherwise, should use Content_Parent instead.

- Usage

```
procedure Content_Put_Text (Object : in out GGB.Base_Type'Class; Text :
String)
```

- Example

```

```

1.5.58 Content_Put_Title

- Description

  Set title above content.

- Usage

```
procedure Content_Put_Title (Object : in out GGB.Base_Type'Class; Title :
String)
```

- Example

```

```

1.5.59 Dialog_Buttons

- Description

  Display a one or two Buttons dialog, for user to quit a screen, user choices like Quit/Update or Cancel/Validate, etc.

- Usage

```
procedure Dialog_Buttons (
                          Object : in out Gnoga.Gui.Base.Base_Type'Class;
                          Key : String;
```

```
                              Left_Text : String;
                              Left_Handler : GGB.Action_Event := null;
                              Right_Text : String;
                              Right_Handler : GGB.Action_Event := null]
```

- Example

```
Gui.Dialog_Buttons (Object, User_Form_Title, "Quit", On_Quit'Unrestricted_Access);
```

1.5.60 Dialog_Popup

- Description

  Create a jQuery dialog, with two potential buttons. Buttons are displayed if their corresponding handler is not null. Confirm button is focused.

  It is advised to use Close_Dialog in On_Confirm and On_Cancel handlers.

  See https://api.jqueryui.com/position for Position_My and Position_At information.

- Usage

```
procedure Dialog_Popup (Object : in out GGB.Base_Type'Class;
                        Title : String; Content : String;
                        Left_Text : String := "";
                        Left_Handler : GGB.Action_Event := null;
                        Right_Text : String := "";
                        Right_Handler : GGB.Action_Event := null;
                        Height : Natural := 150;
                        Width : Natural := 300;
                        Position_My : String := "center top+40";
                        Position_At: String := "center top+40")
```

- Example

```

```

1.5.61 Footer_Set_Left_Text

- Description

  Displays text on the left side of the footer.

- Usage

```
procedure Footer_Set_Left_Text (Object : in out GGB.Base_Type'Class; Text
: String := "")
```

- Example

```

```

1.5.62 Footer_Set_Right_Text

- Description

  Displays text on the right side of the footer.

- Usage

```
procedure Footer_Set_Right_Text (Object : in out GGB.Base_Type'Class;
Text : String := "")
```

- Example

```

```

1.5.63 Header_Application_Menu_Add

- Description

  Add menu and cascaded menu to application menu.
  Key must be unique.
  On_Click handler must call Header_Notify_Menu_Click.

- Usage

```
procedure Header_Application_Menu_Add (Key : String; Name : String; Par-
ent_Key :  String; On_Click : GGB.Action_Event)
```

- Example

```

```

1.5.64 Header_Notify_Menu_Click

- Description

  Header module menu callback.

- Usage

```
procedure Header_Notify_Menu_Click (Object : in out GGB.Base_Type'Class;
Key : String)
```

- Example

```

```

1.5.65 Header_Notify_User_Menu_Click

- Description

  Header user menu callback.

- Usage

```
procedure Header_Notify_User_Menu_Click [Object : in out GGB.Base_Type'Class]
```

- Example

```

```

1.5.66 Header_Set_Root

- Description

  Set default root module menu.
  Key must be unique.
  On_Click handler must call Header_Notify_Menu_Click.

- Usage

```
procedure Header_Set_Root (Key : String; Name : String; On_Click : GGB.Action_Event)
```

- Example

```

```

1.5.67 Header_User_Menu_Add

- Description

  Add menu to user menu.
  On_Click handler must call Header_Notify_User_Menu_Click otherwise.
  User navigation menu can't be closed.

- Usage

```
procedure Header_User_Menu_Add (Object : in out GGB.Base_Type'Class; Name : String; On_Click : GGB.Action_Event)
```

- Example

```

```

1.5.68 Launch_Web

- Description

  Open a new tab in browser to URL.

  User's browser might not accept redirection at first, in this case the user needs to enable this.

- Usage

```
Launch_Web (Object : in out GGB.Base_Type'Class; URL : String)
```

- Example

```

```

1.5.69 List

- Description

  List Table columns listed in List_Columns (comma separated) with header table titles List_Header (comma separated) and Condition.

  List Name is the list title.
  List_Key is the table list unique identifier.

  See Gui.Crud Init below for usage and examples.

- Usage

```
procedure List (Object : in out GGB.Base_Type'Class;
                Db_Name : String;
                Db_Table : String;
                List_Name : String;
                List_Key : String;
                List_Header : String;
                List_Columns : String;
                Condition : String := "")
procedure List (Object : in out GGB.Base_Type'Class;
                DB : in out GSD.Connection'Class;
                Db_Table : String;
                List_Name : String;
                List_Key : String;
                List_Header : String;
                List_Columns : String;
                Condition : String := "")
```

- Example

```
List (Object, DBM, "Sys_Users",
                   "Users List",
                   "Users_List_Key",
                   "Identifier, First Name, Name, Phone, Email",
                   "!Login, First_Name, Last_Name, Phone, Email",
                   "ORDER BY Login");
```

1.5.70 Main_Menu

Main_Menu has two modes.

- Mode 1 – Direct mode

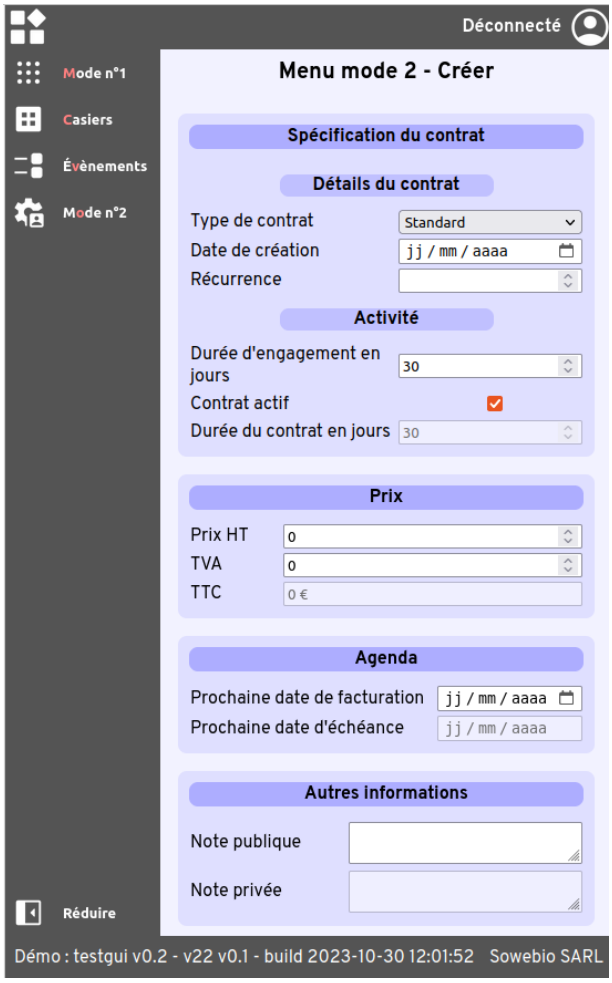
  Using mode one (or direct mode), selecting a menu on the left-hand bar directly calls up a routine. Example:

  A click on “Mode n°1” displays directly the page “Menu mode 1”.

- Mode 2 – sub menu mode

  Using mode two (or sub-menu mode), selecting a menu on the left-hand bar displays a sub menu. Each sub menu calling a routine. Example:

  Then, click on (or type the ‘r’ letter) the sub menu “Créer” displays this demo page:



When using mode two, you must acknowledge the sub-menu by calling the procedure Gui.Main_Menu_Notify_Sub_Element_Click [Object, "**Preferences_Delete**"]; with the sub-key corresponding to the sub-menu. [see testgui code and the call from Gui.-Main_Menu_Add_Sub_Element [Object, "**Preferences_Delete**", "Supprimer", "Preferences", Mgt.Demo_Complex_Form_Delete'Unrestricted_Access];

1.5.71 Main_Menu_Add_Delimiter_Above

- Description

  Add a delimiter above a sub-element.

- Usage

```
procedure Main_Menu_Add_Delimiter_Above [Object : in out GGB.Base_Type'Class; Key : String]
```

- Example

```
```

1.5.72 Main_Menu_Add_Element

- Description

  Add an element to a menu.

  Key must be unique.

- Usage

```
procedure Main_Menu_Add_Element (Object : in out GGB.Base_Type'Class;
                                 Key : String;
                                 Name : String;
                                 Icon_SRC : String;
                                 On_Click : GGB.Action_Event := null)
```

- Example

```

```

1.5.73 Main_Menu_Add_Sub_Element

- Description

  Add an element to a sub menu. If parent is "File", and this element is "Edit", then a unique key should be set as "File_Edit".

  Key must be unique.

  On_Click event procedure must in turn call Main_Menu_Notify_Sub_Element_Click.

- Usage

```
procedure Main_Menu_Add_Sub_Element (Object : in out GGB.Base_Type'Class;
                                     Key : String;
                                     Name : String;
                                     Parent_Key : String;
                                     On_Click : GGB.Action_Event);
```

- Example

```

```

1.5.74 Main_Menu_Clear

- Description

  Clear Main_Menu content.

- Usage

```
procedure Main_Menu_Clear (Object : in out GGB.Base_Type'Class)
```

- Example

```
```

1.5.75 Main_Menu_Disable_Shortcuts

- Description

  Disable Main_Menu shortcuts, for both elements and sub-elements.

- Usage

```
procedure Main_Menu_Disable_Shortcuts (Object : in out GGB.Base_Type'Class)
```

- Example

```
```

1.5.76 Main_Menu_Enable_Shortcuts

- Description

  Enable Main_Menu shortcuts, allowing both elements and sub-elements shortcuts.

- Usage

```
procedure Main_Menu_Enable_Shortcuts (Object : in out GGB.Base_Type'Class)
```

- Example

```
```

1.5.77 Main_Menu_Load

- Description

  Load Main_Menu after elements has been added.

- Usage

```
procedure Main_Menu_Load (Object : in out GGB.Base_Type'Class)
```

- Example

```
```

1.5.78 Main_Menu_ Notify_Sub_Element_Click

- Description

  Callback to place in sub-elements' handlers.

- Usage

```
procedure Main_Menu_Notify_Sub_Element_Click (Object : in out GGB.Base_Type'Class; Key : String)
```

- Example

```

```

1.5.79 Main_Menu_Set_Clickable

- Description

  Set an element (or sub-element) clickable.

- Usage

```
procedure Main_Menu_Set_Clickable (Object : in out GGB.Base_Type'Class; Key : String)
```

- Example

```

```

1.5.80 Main_Menu_Set_Unclickable

- Description

  Set an element (or sub-element) unclickable.

- Usage

```
procedure Main_Menu_Set_Unclickable (Object : in out GGB.Base_Type'Class; Key : String)
```

- Example

```

```

1.5.81 Pop_Up

- Description

  Display a pop-up message with a title defaulted to "Message".

- Usage

```
procedure Pop_Up (Object : in out Gnoga.Gui.Base.Base_Type'Class; Text :
String; Title : String := "Message")
```

- Example

```

```

1.5.82 Print

- Description

  Call system printing on current web view. <<<TODO>>> Don't work, to be fixed. Tends to stop client-server communication.

- Usage

```
procedure Print (Object : in out GGB.Base_Type'Class)
```

- Example

```

```

1.5.83 Put_User_Icon

- Description

  Display the user icon registered by Set_User_Icon.

- Usage

```
procedure Put_User_Icon (Object : in out GGB.Base_Type'Class)
```

- Example

```

```

1.5.84 Set_Application_Icon

- Description

  Set the application icon when clicked displays the modules menu.

- Usage

```
procedure Set_Application_Icon (Icon_File :  String)
```

- Example

```

```

1.5.85 Set_Browser_Icon

- Description

  Set application icon in browser tab. <<<TODO>>> Don't work, to be fixed.

- Usage

```
procedure Set_Browser_Icon (Icon_File : String)
```

- Example

```

```

1.5.86 Set_Browser_Title

- Description

  Set application title in browser tab.

- Usage

```
procedure Set_Browser_Title (Object : in out GGB.Base_Type'Class; Title :
String)
```

- Example

```

```

1.5.87 Set_Debug

- Description

  Set Gnoga debug mode. This mode must be set before calling Setup to take effect.

- Usage

```
procedure Set_Debug (Switch : On_Off)
```

- Example

```

```

1.5.88 Set_Login

- Description

  Application access control.

- Usage

```
procedure Set_Login (Switch : On_Off)
```

- Example

```
```

1.5.89 Set_User_Icon

- Description

  Set the user icon when clicked displays the user menu.

- Usage

```
procedure Set_User_Icon (Icon_File : String)
```

- Example

```
```

1.5.90 Set_User_Name

- Description

  Set user name, displayed next to the user icon.

- Usage

```
procedure Set_User_Name (Object : in out GGB.Base_Type'Class; Name :
String)
```

- Example

```
```

1.5.91 Setup

- Description

  Set application connection parameters
  On_User_Connect is called every time a user calls a web page.

- Usage

```
procedure Setup (On_User_Connect : GGB.Action_Event;
                 Host : String := "";
                 Port : Integer := 8_080;
                 Boot : in String := "boot_jqueryui.html";
                 Title : String := "";
                 Server_Closed_Content : String := "Server closed."]
```

- Example

```
```

```

```

1.5.92 User_Logout

- Description

  Disconnect user. This function assumes Setup was called.

- Usage

```
procedure User_Logout (Object : in out GGB.Base_Type'Class)
```

- Example

```

```

1.6 Gui.Crud – CRUD management

1.6.1 Get

- Description

  Get CRUD parameter from Init procedure.

- Usage

```
function Get (Object : in out Gnoga.Gui.Base.Base_Type'Class; Parameter :
String) return String
```

- Example

```
Gui.Crud.Get (Object, "Db_Key")
```

1.6.2 Init

- Description

  CRUD screen initialization.

  List Name is the list title.
  List_Key is the table list unique identifier.

  List_Header (comma separated) contains the header columns.
  List_Columns (comma separated) contains the list of displayed columns.

  Some tags are automatically generated in the Connection_Data dictionary:
  - “Key”**_n** storing column tagged with "!" value (see below);
  - “Key”**_First** storing column tagged with "!" value for the first line;
  - “Key”**_First_Id** storing first (whole list, not a page) list Id;
  - “Key”**_Id_n** storing Id value for each line;
  - “Key”**_Last** storing column tagged with "!" value for the last line;

- “Key”**_Last_Id** storing last [whole list, not a page] list Id;
- “Key”**_Select** storing column tagged with "!" value for the cliked/selected line.

“Key”_Select is mainly set in On_Delete and On_Update procedures then mainly used in On_Validate_Delete and On_Validate_Update procedures.

- Special tags

  You can tag columns with one or more tags, depending of the context. Example:

```
List_Columns => "Tbl_Dispensers.Name,
                 Tbl_Places.Selection_Number,
                 Tbl_Places.Item_Code,
               M€Tbl_Places.Item_Price,
                 Tbl_Places.Item_Storage_Actual,
               #!Tbl_Places.Id"
```

!	Indexed column key for further seeking the record.
#	Hidden column in list.
*L	Left aligned.
*R	Right aligned.
M$	Money column with $ monetary sign in front of the amount, right aligned.
M€	Money column with € monetary sign behind the amount, right aligned.
M%	Money column with no monetary sign, right aligned.
D/	Date column with DD/MM/YYYY formatting.
D.	Date column with DD.MM.YYYY formatting.
DT/	Date Time column with DD/MM/YYYY HH:MM:SS formatting.
DT.	Date Time column with DD.MM.YYYY HH:MM:SS formatting.

  Note: The default date column format is ISO 8601 YYYY-MM-DD.

- Usage

```
procedure Init (Object : in out Gnoga.Gui.Base.Base_Type'Class;
                Db_Name : String;
                Db_Table : String;
                Db_Key : String;
                List_Name : String;
                List_Key : String;
                List_Header : String;
                List_Columns : String;
                List_Join : String := "";
                List_Limit : Positive := 10)
```

- Example 1

```
Gui.Crud.Init (Object,
               Db_Name => Config.Database_Name,
               Db_Table => "Sys_Users",
               Db_Key => "Users_Key",
               List_Name => "Users List",
               List_Key => "Users_List_Crud",
               List_Header => "Identifier, First Name, Name, Phone, Email",
```

```
                List_Columns => "!Login, First_Name, Last_Name, Phone, Email"]
```

User List

	Identifier	*First name*	*Name*	*Phone*	*Email*
1	admin6	Number	Six		
2	admin7	Number	Seven		
3	**alpha**	Steve	Howe		
4	delta	Jon	Anderson		
5	foxtrot	Chris	Squire		
6	juliet	Bill	Bruford		
7	tango	Pete	Townsend		
8	uniform	Roger	Daltrey		
9	xray	Keith	Moon		
10	zoulou	John	Entwistle		

When a click event is trigged on the **third line** :

```
Key : constant UXString := "Users_List_Key";
Data_Index : constant Integer := Gui.Content_List_Selected_Row (Object, Key);
Data_Value : constant String := Gui.Get_Connection_Data (Object, Key & "_" &
To_String_Unsigned (Data_Index));

Results...

Data_Index: 3
Search key is: Users_List_Key_3
Data_Value: alpha
```

- Example 2

```
With List_Key = Places_Crud

Data_Index : constant String := To_String_Unsigned (Gui.Content_List_Selected_Row (Ob-
ject, Gui.Crud.Get (Object, "List_Key")));

Data_Index = 5 (index in displayed lines of list, 5 = fifth line of the list)

Data_Value : constant String := Gui.Get_Connection_Data (Object, Gui.Crud.Get (Object,
"List_Key") & "_" & Data_Index);

Places_Crud_5 = 59 (db key index, 59 is the value to seek)

Then store 59 in Places_Crud_Select key for further use in search for updating or
deletion.
```

- Example 3 - Complex list with JOIN

```
Gui.Crud.Init (Object,
               Db_Name => Config.Database_Name,
               Db_Table => "Tbl_Dispensers",
               Db_Key => "Name",
               List_Name => "Liste des Distributeurs",
               List_Key => "Dispensers_Crud",
```

```
               List_Header => "Nom, Point de vente",
               List_Columns => "~Tbl_Dispensers.Name, Tbl_Point_Of_Sales.Name",
               List_Join => "INNER JOIN Tbl_Point_Of_Sales ON Tbl_Point_Of_Sales.Id =
Tbl_Dispensers.Point_Of_Sales",
               List_Limit => 10);
```

- Example 4- Event CRUD by Id

```
Gui.List > Key/First: Events_Crud_First/969
Gui.List > Key_Id_n/Value: Events_Crud_Id_1/969
Gui.List > Key_Id_n/Value: Events_Crud_Id_2/968
Gui.List > Key_Id_n/Value: Events_Crud_Id_3/967
Gui.List > Key_Id_n/Value: Events_Crud_Id_4/966
Gui.List > Key_Id_n/Value: Events_Crud_Id_5/965
Gui.List > Key_Id_n/Value: Events_Crud_Id_6/964
Gui.List > Key_Id_n/Value: Events_Crud_Id_7/963
Gui.List > Key_Id_n/Value: Events_Crud_Id_8/962
Gui.List > Key_Id_n/Value: Events_Crud_Id_9/961
Gui.List > Key_Id_n/Value: Events_Crud_Id_10/960
Gui.List > Key_Id_n/Value: Events_Crud_Id_11/959
Gui.List > Key_Id_n/Value: Events_Crud_Id_12/958
Gui.List > Key_Id_n/Value: Events_Crud_Id_13/957
Gui.List > Key_Id_n/Value: Events_Crud_Id_14/956
Gui.List > Key_Id_n/Value: Events_Crud_Id_15/955
Gui.List > Key_Id_n/Value: Events_Crud_Id_16/954
Gui.List > Key_Id_n/Value: Events_Crud_Id_17/953
Gui.List > Key_Id_n/Value: Events_Crud_Id_18/952
Gui.List > Key_Id_n/Value: Events_Crud_Id_19/951
Gui.List > Key_Id_n/Value: Events_Crud_Id_20/950
Gui.List > Key/Last: Events_Crud_Last/950
```

Example 5 - Dispensers by Key [Dispensers name]

```
Gui.List > Key/First: Dispensers_Crud_First/dis 1
Gui.List > Key/Value: Dispensers_Crud_1/dis 1
Gui.List > Key/Value: Dispensers_Crud_2/dis 2
Gui.List > Key/Last: Dispensers_Crud_Last/dis 2
```

1.6.3 List

- Description

  Display a list in CRUD screen.

  CRUD screen management needs to be initialized, typically done in the CRUD main menu. See the Init procedure example above.

- Usage

```
procedure List [Object : in out GGB.Base_Type'Class; Condition :
String := ""]
```

- Example

```
Gui.Crud.List [Object];
```

```
Gui.Crud.List (Object, "WHERE " & Gui.Crud.Get (Object, "Db_Key") & " >= '" & Sql.Escape_String (Data_Value) & "'");
```

1.6.4 List_Length_Get

- Description

  Get list length in CRUD screen.

- Usage

```
function List_Length_Get return Positive
```

- Example

```

```

1.6.5 List_Length_Set

- Description

  Set list length in CRUD screen.

- Usage

```
procedure List_Length_Set (Length : Positive)
```

- Example

```

```

1.6.6 List_Reset

- Description

  Reset list cursor to the list's beginning.

- Usage

```
procedure List_Reset (Object : in out GGB.Base_Type'Class)
```

- Example

```

```

1.6.7 On_Cancel

- Description

  Handle cancel in CRUD screen.

- Usage

```
procedure On_Cancel (Object : in out GGB.Base_Type'Class)
```

- Example

```
Gui.Crud.On_Cancel'Unrestricted_Access
```

1.6.8 On_Down

- Description

  Manage list scroll down in CRUD screen. See List below for further information.

- Usage

```
procedure On_Down (Object : in out GGB.Base_Type'Class)
```

- Example

```
Gui.Crud.On_Down'Unrestricted_Access
```

1.6.9 On_Filter_Off

- Description

  Unfilter the current filtered list.

- Usage

```
procedure On_Filter_Off (Object : in out GGB.Base_Type'Class)
```

- Example

```
Gui.Crud.On_Filter_Off'Unrestricted_Access
```

1.6.10 On_Refresh

- Description

  List refresh.

- Usage

```
procedure On_Refresh (Object : in out GGB.Base_Type'Class)
```

- Example

```
Gui.Crud.On_Down'Unrestricted_Access
```

1.6.11 On_Up

- Description

  Manage list scroll up in CRUD screen. See List_Put below for further information.

- Usage

```
procedure List_On_Up (Object : in out GGB.Base_Type'Class)
```

- Example

```
Gui.Crud.On_Up'Unrestricted_Access
```

1.6.12 Put_Error_Message

- Description

  Display a error message at the bottom of the screen.

- Usage

```
procedure List_On_Up (Object : in out GGB.Base_Type'Class)
```

- Example

```
Gui.Crud.On_Up'Unrestricted_Access
```

1.6.13 Read

- Description

  Read a record in CRUD screens.

Key is assumed to be already space trimmed.

This function doesn't trim Key, as Key must already be space trimmed in the Query, to ensure that there are no space in the written key, and that subsequent searches will be accurate.

- Usage

```
function Read (Object : in out Gnoga.Gui.Base.Base_Type'Class;
               Key : String;
               Query : String) return String
```

- Example

```

```

1.6.14 Search

- Description

  Search on Key in CRUD screen.

- Usage

```
function Search (Object : in out Gnoga.Gui.Base.Base_Type'Class;
                 Key : String;
                 Message_Not_Found : String) return Boolean
```

- Example

```

```

1.6.15 Title

- Description

  List title in CRUD screens.

  Valid Mode is: DB_CREATE, DB_READ, DB_UPDATE, DB_DELETE, DB_SEARCH.

- Usage

```
procedure Title (Object : in out Gnoga.Gui.Base.Base_Type'Class; Mode :
String; Title : String := "")
```

- Example

```

```

1.6.16 Write

- Description

  Write a record in CRUD screens.

  Operation is: DB_CREATE, DB_UPDATE.

- Usage

```
procedure Write (Object : in out Gnoga.Gui.Base.Base_Type'Class;
                 Operation : String;
                 Key : String;
                 Query : String ;
                 Message_Key_Already_Exists : String;
                 Message_Key_Is_Missing : String);
```

- Example

```

```

1.7 Msg – Logging management

To avoid name conflicts with Gnoga, which has a log procedure, the original v20.Log package has been renamed v22.Msg. Then original procedure v20.Msg has been renamed v22.Info as information message.

Msg is a task aware package, using a mutex schema to avoid collision.

1.7.1 Debug

- Description

Log a debug message. 45 characters max before truncation with a maximum line length of 79.

- Usage

```
procedure Debug (Message : Boolean)
procedure Debug (Message : String)
```

- Example

```

```

1.7.2 Debug_Latin_1

- Description

Same as previous Debug procedure but accept Latin_1 Standard.String as input. Log a debug message. 45 characters max before truncation with a maximum line length of 79.

- Usage

```
procedure Debug_Latin_1 (Message : Standard.String)
```

- Example

```

```

1.7.3 Error

- Description

  Log an error message. 45 characters max before truncation with a maximum line length of 79.

- Usage

```
procedure Error (Message : in String)
```

- Example

```

```

1.7.4 Error_Latin_1

- Description

  Same as previous Error procedure but accept Latin_1 Standard.String as input. Log an error message. 45 characters max before truncation with a maximum line length of 79.

- Usage

```
procedure Error_Latin_1 (Message : in Standard.String)
```

- Example

```

```

1.7.5 Get_Dir

- Description

```
Returns log file directory.
```

- Usage

  function Get_Dir return String

- Example

```

```

1.7.6 Info

- Description

  Log an information message. 45 characters max before truncation with a maximum line length of 79.

- Usage

```
procedure Info (Message : Boolean)
procedure Info (Message : On_Off)
procedure Info (Message : ASCII_Character)
procedure Info (Message : String)
procedure Info (Message : Integer)
procedure Info (Message : Long_Integer)
procedure Info (Message : Long_Long_Integer)
procedure Info (Message : Float)
procedure Info (Message : Money)
```

- Example

```

```

1.7.7 Info_Latin_1

- Description

  Same as previous Info procedure but accept Latin_1 Standard.String as input. Log an information message. 45 characters max before truncation with a maximum line length of 79.

- Usage

```
procedure Info_Latin_1 (Message : Standard.String)
```

- Example

```

```

1.7.8 Is_Debug

- Description

```
Return debug status.
```

- Usage

  function Is_Debug return On_Off

- Example

```

```

1.7.9 New_Line

- Description

  Log a blank line.

- Usage

```
procedure New_Line
```

- Example

```

```

1.7.10 Set_Debug

- Description

  Set debug messages status on/[off].

- Usage

```
procedure Set_Debug [Switch : On_Off]
```

- Example

```
Msg.Set_Debug [On]
```

1.7.11 Set_Dir

- Description

  Set log file directory.

- Usage

```
procedure Set_Dir [Dir_In : String]
```

- Example

```

```

1.7.12 Set_Display

- Description

  Log to display on/[off].

- Usage

```
procedure Set_Display [Switch : On_Off]
```

- Example

```
Msg.Set_Display [On]
```

1.7.13 Set_Disk

- Description

  Log to disk on/[off].

- Usage

```
procedure Set_Disk [Switch : On_Off]
```

- Example

```
Msg.Set_Disk [On]
```

1.7.14 Set_Header

- Description

  Line header on/[off].

- Usage

```
procedure Set_Header [Switch : On_Off]
```

- Example

```
Msg.Set_Header [On]
```

1.7.15 Set_Task

- Description

  Set new current log task name. 7 characters max before truncation.

- Usage

```
procedure Set_Task [New_Task : String]
```

- Example

```

```

1.7.16 Title

- Description

  Log a title. 45 characters max before truncation with a maximum line length of 79.

- Usage

```
procedure Title [Message : in String]
```

- Example

```
```

1.8 Net – Network management

1.8.1 Api

- Description

  Communication with hoster or software API. At present, only Ovh hoster and Matomo software are handled.

- Usage

```
   function Api [Api_Provider : Api_Providers;
                 Api_End_Point : String;
                 Api_Consumer_Key : String;
                 Api_Application_Key : String;
                 Api_Application_Secret : String := "";
                 Api_Method : Api_Methods;
                 Api_Query : String;
                 Api_Body : String := ""
                ] return String
```

- Example OVH

```
with GNATCOLL.JSON;
…
package GJ renames GNATCOLL.JSON;
…

   declare
      Ovh_Consumer_Key : constant String := "579***3d5";
      Ovh_Application_Key : constant String := "4be***5b4";
      Ovh_Application_Secret : constant String := "069***b7b";

      Answer : String;
      Api_Query : GJ.JSON_Value := GJ.Create_Object;
   begin

      Answer := Net.Api [Net.Ovh,
                         "eu.api.ovh.com/v1",
                         Ovh_Consumer_Key,
                         Ovh_Application_Key,
                         Ovh_Application_Secret,
                         Net.Get,
                         "domain/zone/ab***em.org"
                        ];
      Msg.Info [Answer];

      --  Populate JSON query
      GJ.Set_Field [Api_Query, "fieldType", "TXT"];
      GJ.Set_Field [Api_Query, "subDomain", "BBBB_Y"];
      GJ.Set_Field [Api_Query, "target", "DDDD_D"];
```

```
      Answer := Net.Api (Net.Ovh,
                         "eu.api.ovh.com/v1",
                         Ovh_Consumer_Key,
                         Ovh_Application_Key,
                         Ovh_Application_Secret,
                         Net.Post,
                         "domain/zone/ab***em.org/record",
                         From_Latin_1 (GJ.Write (Api_Query)));
      Msg.Info (Answer);
```

- Example Matomo

```
with GNATCOLL.JSON;
…
package GJ renames GNATCOLL.JSON;
…

      MatomoConsumerKey : constant String := "45e***142";

      Answer := Net.Api (Net.Matomo,
                         "analytics.soweb.io",
                         MatomoConsumerKey,
                         Net.Post,
                         "  -d idSite=5"
                        );
      Msg.Info (Answer);
```

1.8.2 Command

- Description

  Send remote command to host. Returns True if command successful (remote exit code equal to 0). If used, SE_Output returns remote console output.

  When using with password instead of key, the package "sshpass" must be installed.

- Usage

```
function Command (Target : in String;
                  Command_In : in String;
                  SE_Output : out String) return Boolean

function Command (Target : in String;
                  Command : in String) return Boolean

procedure Command (Target : in String;
                   Command : in String)
```

- Exception

Error_Command	Raised when send command error

- Example

```
List files in a directory:

Net.Command ("root@i51c1.domain.tld", "cd /root; ls -l");
```

```
drwxr-xr-x 7 root     root     4.0K Sep  1 10:45 acme.sh
-rw-r--r-- 1 root     root     3.4K Aug  5 09:28 aide.err
-rw-r--r-- 1 root     root      12K Aug  5 09:53 aide.log
-rw-r--r-- 1 root     root        1 Aug  5 09:28 check.gpr
drwxr-xr-x 2 root     root     4.0K Dec 11 15:02 dmf
-rwxr-xr-x 1 root     root     2.7M Dec 14 11:37 gprbuild
drwxr-xr-x 3 root     root     4.0K Aug  5 09:53 opt
-rw-r--r-- 1 root     root      47M Sep 25 11:37 s015.sql
-rw-r--r-- 1 root     root      134 Aug  7 17:14 test.txt

Complex command example (massive URL change in wordpress DB):

Net.Command ("root@i152c1", +"cd /srv/www/adm152.temp_domain.tld/sar ; php srdb.-
cli.php -h localhost -n dmf_transfert -u dmf -p " & Pwd_DB_Prod & " -s https://
www.old_domain.tld -r https://www.new_domain.tld");
```

1.8.3 Copy_File

- Description

  Copy to distant host. Returns True if copy successful.

  Options allows extra parameters, like -r for recursive copy.

- Usage

```
function Copy_File (Target : in String;
                    File_Tx : in String;
                    Directory_Rx : in String;
                    Options : in String := "") return Boolean;

procedure Copy_File (Target : in String;
                     File_Tx : in String;
                     Directory_Rx : in String;
                     Options : in String := "");
```

- Exception

Error_Copy	Raised when send file error

- Example

```
Copy /home/sr/text.txt to root@i51c1.domain.tld/etc/genesix/test.txt

Net.Copy_File ("root@i51c1.domain.tld","/home/sr/text.txt","/etc/genesix");

Copy recursively from Prg.Start_Dir & "/" & Project_Sources/* to Remote_User & "@" &
Remote_Host:/Remote_Root & "/" & Project_Name/*.

Net.Copy_File (Remote_User & "@" & Remote_Host,
               Prg.Start_Dir & "/" & Project_Sources,
               Remote_Root & "/" & Project_Name,
               "-r");
```

1.8.4 Copy_Rsync

- Description

  Rsync efficient copy to distant host with unlimited directories excluding masks, delimited by semicolons.

  Returns True if copy successful.

- Usage

```
function Copy_Rsync (Target : in String ;
                     Directory_Tx : in String;
                     Directory_Rx : in String ;
                     Excludes_Directories : in String := "") return Boolean;

procedure Copy_Rsync (Target : in String ;
                      Directory_Tx : in String;
                      Directory_Rx : in String ;
                      Excludes_Directories : in String := "");
```

- Exception

Error_Copy	Raised when send file error

- Example

```
Net.Copy_Rsync (Remote_User & "@" & Remote_Host,
                Prg.Start_Dir & "/../" & Project_Libraries & "/",
                Remote_Root & "/" & Project_Libraries,
                "obj:.*:src:tools");
```

1.8.5 Delete_Directory_Tree

- Description

  Delete a directory tree Dir_Tree. The directory and all of its contents (possibly including other directories) are deleted but adding a '*' at the end of the path preserve the last directory of the path (/one/two/ deletes two but /on/two/* preserve two.

  Return True if Dir_Tree is successfully deleted or was already deleted. Return False if operation is unsuccessful (i.e. if base directory tree was non existent or still exists after the deleting attempt).

  Dir_Tree must be fully qualified, ie starting with a slash (/). This function prevents deletion of the following root directories (see Is_Root_Directory for further details). Pay close attention, you can't delete /etc but you are allowed to delete /etc/network !

  Delete a directory tree Dir_Tree. The directory and all of its contents (possibly including other directories) are deleted. Return True if Dir_Tree is successfully deleted

or was already deleted. Return False if operation is unsuccessful (i.e. if base directory tree was non existent or still exists after the deleting attempt).

Dir_Tree must be fully qualified, i.e. starting with a slash (/).

This function prevents deletion of the following root directories: bin, boot, dev, etc, home, lib, lib32, lib64, libx32, lost+found, media, mnt, opt, proc, root, run, sbin, srv, sys, tmp, usr, var. Pay close attention, you can't delete /etc but you are allowed to delete /etc/network !

With programs ran with root rights, this routine should be used with infinite caution.

- Usage

```
function Delete_Directory_Tree (Target : in String;
                                Dir_Tree : String) return Boolean
```

- Example

```
```

1.8.6 Delete_File

- Description

Remove File_To_Delete in remote host Target. Returns True if delete successful.

Delete_File returns True even File_To_Delete did not exist.

- Usage

```
function Delete_File (Target : in String;
                      File_To_Delete : in String) return Boolean;

procedure Delete_File (Target : in String;
                       File_To_Delete : in String);
```

- Example

```
Copy /home/sr/text.txt to root@i51c1.domain.tld/etc/genesix/test.txt

Net.Delete_File ("root@i51c1.domain.tld","/home/sr/text.txt");
```

1.8.7 Directory_Exists

- Description

Returns True if distant directory Name exists.

- Usage

```
function Directory_Exists (Target : in String;
                           Name : String) return Boolean
```

- Example

```
if Net.Directory_Exists (+"host", "Directory_to_test") then
   Tio.Put_Line ("Directory_to_test exists");
end if;
```

1.8.8 File_Exists

- Description

  Returns True if file Name exists.

- Usage

```
function File_Exists (Target : in String;
                      Name : String) return Boolean
```

- Example

```
if Net.File_Exists ("host", "filename_to_test") then
   Tio.Put_Line ("filename_to_test exists");
end if;
```

1.8.9 Get_Exception

- Description

  Returns Exception status.

- Usage

```
function Get_Exception return On_Off
```

- Example

```
Tio.Put_Line ( Get_Exception);

On
```

1.8.10 Get_Network_From_Ip

- Description

  Returns the network part of a /32 classless IP address.

- Usage

```
function Get_Network_From_Ip (Ip : in String) return String
```

- Example

```
Tio.Put_Line ( Net.Get_Network_From_Ip ("120.1.1.1"));
120.1.1
Tio.Put_Line ( Net.Get_Network_From_Ip ("320.1.1.1"));
Empty string
```

1.8.11 Is_Ip_Ok

- Description

  IP validation

- Usage

```
function Is_Ip_Ok (IP : in String) return Boolean;
```

- Example

```
Tio.Put_Line (Is_Ip_Ok ("320.1.1.1"));
False
Tio.Put_Line (Is_Ip_Ok ("120.1.1.1"));
True
```

1.8.12 Is_Ping_Ok

- Description

  Return true if target answer to a ping.

- Usage

```
function Is_Ping_Ok (Target : in String) return Boolean;
```

- Example

```
Net.Is_Ping_Ok ("This_host_exists");
True
Net.Is_Ping_Ok ("This_host_don't_exist");
False
```

1.8.13 Is_Root_Directory

- Description

  This function checks the following root directories: bin, boot, dev, etc, home, lib, lib32, lib64, libx32, lost+found, media, mnt, opt, proc, root, run, sbin, srv, sys, tmp, usr, var. Returns True if Dir_Tree is a root directory.

  Dir_Tree must be fully qualified, ie starting with a slash [/].

- Usage

```
function Is_Root_Directory [Dir_Tree : String] return Boolean
```

- Example

```
Tio.Put_Line [Net.Is_Root_Directory [“/etc”]];
True

Tio.Put_Line [Net.Is_Root_Directory [“/etc/network”]];
False
```

1.8.14 Is_Ssh_Ok

- Description

  Return true if target answer to a ping.

- Usage

```
function Is_Ssh_Ok [Target : in String] return Boolean;
```

- Example

```
Net.Is_Ssh_Ok [+"This_host_exists_with_valid_Ssh_Credentials"];

True

Net.Is_Ssh_Ok [+"This_host_exists_without_valid_Ssh_Credentials"];

False
```

1.8.15 Mount

- Description

  Mount a Target as host.

  If local admin, automatically create local mount point in
  - /mnt/<Target>

  If local [non root] user, automatically create local mount point in
  - /home/<user>/mnt/<Target>

- Usage

```
procedure Mount (Target : String);
```

- Example

```
Local admin case: Mount ("root@i51c1.domain.tld");

Mounts target root@i51c1.domain.tld: /
to /mnt/root@i51c1.domain.tld

Local <user> case (with home at /home/user): Mount ("root@i51c1.domain.tld");

Mounts target sr@i51c1.domain.tld: /
to /home/<user>/mnt/sr@i51c1.domain.tld
```

1.8.16 Mount_Remote

- Description

Mount a Mount_Point targetting Target_To_Mount on Remote_Host with options Mount_Options. All mount options are accepted. Returns true if operation is successful.

- Usage

```
function Mount_Remote (Remote_Host : String;
                       Target_To_Mount : String;
                       Mount_Point : String;
                       Mount_Options : in String := "") return Boolean

procedure Mount_Remote (Remote_Host : String;
                        Target_To_Mount : String;
                        Mount_Point : String;
                        Mount_Options : in String := "");
```

- Example

```
if Net.Mount_Remote ("user@remote_host.org", +"/dev/vg/lvm_volume", +"/tmp/mount-
point", +"-o ro") then
   Tio.Put ("Mount point is mounted read-only");
end if;
```

1.8.17 Send_Mail

- Description

Send a email. Returns true if succeed. Note that this status just means that the mail has been accepted into the mail queue, not that the mail has actually been sent with or without success. This is mail.

No one can reliably test that an email exists. Send_Mail just check From and To email's domains validity and if theses domains have a MX server.

Sender is optional.

Body_Text uses <CR> for new line.

- Dependencies

  bind9-dnsutils package (nslookup)
  sendemail (email sending)

- Usage

```
function Send_Mail (From : String ;
                    To : String ;
                    Subject : String ;
                    Body_Text : String ;
                    Sender : String := "") return Boolean
```

- Example

```
```

1.8.18 Send_SMS

- Description

  Accepts UTF-8 encoding and multiple receivers.

  Returns a SMS sending service dependent result string.

  The sender must be registered against the SMS sending service.

  Sms_Message uses <LF> for new line.

- Dependencies

  curl (api)

- Usage

```
function Send_Sms (Api_Provider : Api_Providers;
                   Api_End_Point : String;
                   Api_Consumer_Key : String;
                   Api_Application_Key : String;
                   Api_Application_Secret : String := "";
                   Sms_Account : String;
                   Sms_Sender : String;
                   Sms_Receivers_List : String;
                   Sms_Message : String := ""
                  ) return String
```

- Example

```
      Result := Net.Send_Sms (Net.OVHcloud,
                              "eu.api.ovh.com/v1",
```

```
                                "579***3d5",
                                "4be***5b4",
                                "069***b7b",
                                "sms-rs16623-2",
                                "V***U.COM",
                                "+336***:+336***",
                                "UTF-8 encoded SMS (éèàùôöœï) with..." & LF &
                                "...line feed !" & LF & LF &
                                "Signed : the plastic duck :)"
                                );

Result    :    {"ids":[572940644,572940647],"tag":"ja0jfkwkqqj38d4r","validReceivers":
["+336***","+336***"],"totalCreditsRemoved":6,"invalidReceivers":[]}
```

1.8.19 Set_Exception

- Description

  Enable Exception processing, which is disabled by default.

- Usage

```
procedure Set_Exception (Switch : On_Off := On)
```

- Example

```
Private_Key := Sql.Read ("Tbl_Cluster", "Key_Private", "WHERE Number = 1");

Set_Exception;
```

1.8.20 Set_Hostname

- Description

  Set Hostname for a Target host. Returns true if command ok.

- Usage

```
function Set_Hostname (Target : String;
                       Hostname : String) return Boolean
```

- Example

```
Private_Key := Sql.Read ("Tbl_Cluster", "Key_Private", "WHERE Number = 1");

if Set_Hostame ("i11c1", "i110c1") then
  Tio.Put ("Hostname is changed");
end if;
```

1.8.21 Set_Key

- Description

  Set SSH private key used to log in distant hosts with commands like Send_Command and Send_File. Key validity is checked. Returns true if Key is properly set.

  A call without parameter delete the key previously set.

- Usage

```
function Set_Key [Key : String := ""] return Boolean;

procedure Set_Key;
```

- Example

```
Private_Key := Sql.Read ["Tbl_Cluster", "Key_Private", "WHERE Number = 1"];

if Set_Key [Private_Key] then
   Tio.Put ["Key is set"];
end if;
```

1.8.22 Set_Message

- Description

  Control message output when using commands like Send_Command and Send_File. Default is console message enable. A call without parameter enable message output.

- Usage

```
procedure Set_Message [Switch : On_Off := On];
```

- Example

```
- Disable console message when using commands like Send_Command and Send_File.

Net.Set_Message [Off];
```

1.8.23 Set_Output

- Description

  Control console output when using commands like Send_Command and Send_File. Default is console output enable. A call without parameter enable console output.

- Usage

```
procedure Set_Output [Switch : On_Off := On];
```

- Example

```
- Disable console output when using commands like Send_Command and Send_File.

Net.Set_Output (Off);
```

1.8.24 Set_Password

- Description

  Set SSH password used to log in distant hosts with commands like Send_Command and Send_File.

  A call without parameter delete the key previously set.

- Usage

```
procedure Set_Password (Password : String := "")

procedure Set_Password;
```

- Example

```

```

1.8.25 Unmount

- Description

  Unmount a mount point on a remote host.

  The local mountpoint directory is deleted.

- Usage

```
procedure Unmount (Target : String);
```

- Exception

Error_Unmount	Raised when unmount error

- Example

```
Local admin case: Net.Unmount ("root@i51c1.domain.tld");

Unmounts /mnt/root@i51c1.domain.tld

Local <user> case (home is /home/user): Net.Unmount ("root@i51c1.domain.tld");

Unmounts /home/<user>/mnt/sr@i51c1.domain.tld
```

1.8.26 Unmount_Remote

- Description

  Unmount a Mount_Point on a Remote_Host. Mount_Point is then deleted. Returns true if the whole operation is successful.

- Usage

```
function Unmount_Remote (Remote_Host : String;
                         Mount_Point : String) return Boolean
procedure Unmount_Remote (Remote_Host : String;
                          Mount_Point : String);
```

- Example

```
if Net.Unmount_Remote ("user@remote_host.org", +"/tmp/mountpoint") then
   Tio.Put ("Mount point is unmounted");
end if;
```

1.9 Pdf – Pdf file management

1.9.1 Get_Scale

- Description

  Return the number of characters which can be displayed in one line depending of the font size.

- Usage

```
function Get_Scale (Object : in out PDF_Out.PDF_Out_File) return Real
```

- Example

```
Get_Scale (My_PDF_File);
```

1.9.2 Get_Size

- Description

  Returns the Str displayed size.

- Usage

```
function Get_Size (Object : in out PDF_Out.PDF_Out_File; Str : String)
return Real
```

- Example

```
Get_Size (My_PDF_File, "What is my size ?");
```

1.9.3 Get_X_Align

- Description

  Return the horizontal coordinate where Str has to be placed to be centered on X coordinate.

- Usage

```
function Get_X_Align (Object : in out PDF_Out.PDF_Out_File; X : Real ; Str : String) return Real
```

- Example

```
Get_X_Align (My_PDF_File, 5.0 * one_cm, "I want to be centered on the 5th cm");
```

1.9.4 Get_X_Center

- Description

  Return the horizontal coordinate where Str has to be placed to be centered on the line.

- Usage

```
function Get_X_Center (Object : in out PDF_Out.PDF_Out_File; Str : String) return Real
```

- Example

```
Get_X_Center (My_PDF_File, "I want to be centered on the page");
```

1.9.5 Get_X_Right

- Description

  Return the horizontal coordinate where Str has to be right justified with a Right margin after Str.

- Usage

```
function Get_X_Right (Object : in out PDF_Out.PDF_Out_File; Right : Real ; Str : String) return Real
```

- Example

```
Get_X_Right (My_PDF_File, 2.0 * one_cm, "I want to left 2cm between me and the right side of the page");
```

1.9.6 Put_Footer

- Description

  Display the last defined footer in the current page.

- Usage

```
procedure Put_Footer (Object : in out PDF_Out.PDF_Out_File)
```

- Example

```
Put_Footer (My_PDF_File);
```

1.9.7 Put_Header

- Description

  Display the last defined header in the current page with a custom title for the page.

- Usage

```
procedure Put_Header (Object : in out PDF_Out.PDF_Out_File; Title : String)
```

- Example

```
Put_Header (My_PDF_File, "My page title");
```

1.9.8 Put_Link

- Description

  Display a clickable link at the given X, Y coordinates.
  Default color is blue.
  Default displayed text is the same as the link.

- Usage

```
type Reference is (Go_To, Mail_To, URI);

procedure Put_Link (Object : in out PDF_Out.PDF_Out_File; X, Y : Real ; Url : String ; Pre : Reference ; Str : String := "";Color : Color_Type := (0.0, 0.0, 0.4))
```

- Example

```
Put_Link (My_PDF_File, 2 * one_cm, 20 * one_cm, "www.soweb.io", URI, "my website");

Put_Link (My_PDF_File, 2 * one_cm, 18 * one_cm, "contact@soweb.io", Mail_To, "contact me");

Put_Link (My_PDF_File, 2 * one_cm, 16 * one_cm, "3", Go_To, "Jump to page 3");
```

1.9.9 Put_Pie_Chart

- Description

  Draw a pie chart.

- Usage

```
procedure Put_Pie_Chart (Object : in out PDF_Out.PDF_Out_File; Center : Point; Radius : Real; Data : Pie_Content)
```

- Example

```
Pie_Data : Pie_Content := [[25.0, [0.5, 0.5, 0.5]], [55.0, [0.8, 0.8, 0.8]], [20.0, [0.3, 0.3, 0.3]]];

Put_Pie_Chart (My_PDF_File, [15.0 * one_cm, 20.0 * one_cm], 10.0, Pie_Data)
```

1.9.10 Put_X_Align

- Description

  Display Str on Y and horizontally centered on X.

- Usage

```
procedure Put_X_Align (Object : in out PDF_Out.PDF_Out_File; X, Y : Real; Str : String)
```

- Example

```
Put_X_Align (My_PDF_File, 5.0 * one_cm, 20.0 * one_cm, "I´ll be horizontally centered on the 5th cm")
```

1.9.11 Put_X_Center

- Description

  Display Str on Y and horizontally centered on the page.

- Usage

```
procedure Put_X_Center (Object : in out PDF_Out.PDF_Out_File; Y : Real;
Str : String)
```

- Example

```
Put_X_Align (My_PDF_File, 20.0 * one_cm, "I'll be horizontally centered on the page")
```

1.9.12 Put_X_Right

- Description

  Display a right justified Str on Y with a Right margin after Str.

- Usage

```
procedure Put_X_Right (Object : in out PDF_Out.PDF_Out_File; Right, Y :
Real; Str : String)
```

- Example

```
Put_X_Right (My_PDF_File, 2.0 * one_cm, 20.0 * one_cm, "There will be 2cm between me
and the right side of the page")
```

1.9.13 Set_Footer

- Description

  Footer setup. Paginate is the maximum page's number.

- Usage

```
procedure Set_Footer (Website, Contact, Item : String := ""; Paginate :
Integer := 0)
```

- Example

```
Set_Footer ("www.soweb.io", "contact@soweb.io", "2025/02/17"; 17)
```

1.9.14 Set_Header

- Description

  Header setup. Logo_Url is a link to the clickable logo (if exists).

➪ All picture in pdf_out is in jpeg format.

- Usage

```
procedure Set_Header (Logo_Path, Line_One, Line_Two : String; Logo_Url : String := "")
```

- Example

```
Set_Header ("logo.jpeg", "this is a pdf for", "www.soweb.io", "www.soweb.io");
```

1.10 Prg – Program management

1.10.1 Check_Password

- Description

  Check password strength. A password must have at least contains uppercase, lowercase, digit and one special char among "-" and "_".

- Usage

```
function Check_Password (Password : String) return Boolean
```

- Example

```

```

1.10.2 Command

- Description

  Constant storing program command (Arg 0).

- Usage

```
Command : constant String
```

- Example

```
Tio_Line (Command);

/home/sr/Seafile/Sowebio/informatique/dev/ada/app/gnx/src/gnx-instance
```

1.10.3 Current_Time_Seconds

- Description

  Returns a duration as seconds since ISO date 197001010. Conforms to Unix time standard. Checked with date +%s. Compliant algorithm until 2070.

  Returns a duration in seconds since current time.

- Usage

```
function Current_Time_Seconds return Natural
```

- Example

```
Log.Msg ["Current time in seconds: " & To_String [Current_Time_Seconds]];
1646227335
```

1.10.4 Date

- Description

  Returns current date as YYYY-MM-DD.

- Usage

```
function Date return String
```

- Example

```

```

1.10.5 Date_Not_Reached

- Description

  Returns true if YYYY-MM-DD DTS is still to come.

  Accept “YYYY-MM-DD” and “YYYY-MM-DD HH:MM:SS” formats but only uses the date part. Checks “YYYY-MM-DD” format before computing.

- Usage

```
function Date_Not_Reached [DTS : String] return Boolean
```

- Example

```

```

1.10.6 Date_Time_Stamp_Reached

- Description

  Returns true if YYYYMMDD-HHMMSS Date_Time_Stamp has passed.

  Checks format before computing.

  You can choose between UTC and Local [Default] time.

- Usage

```
type Time_UTC_Local is (UTC, Local);

function Date_Time_Stamp_Reached (DTS : String ; Zone : Time_UTC_Local := Local) return Boolean
```

- Example

```
if Prg.Date_Time_Stamp_Reached (Task_DTS_Start) then

   iMsg.Info (Prg.Name & " > Task DTS_Start: reached");

end if;
```

1.10.7 Date_Time_Stamp

- Description

  Returns current date time stamp as “YYYYMMDD-HHMMSS”.

  You can choose between UTC and Local (Default) time.

- Usage

```
type Time_UTC_Local is (UTC, Local);

function Date_Time_Stamp (Zone : Time_UTC_Local := Local) return String
```

- Example

```

```

1.10.8 Date_Time_Stamp_From_ISO

- Description

  Returns “YYYYMMDD-HHMMSS” from a date time stamp as “YYYY-MM-DD HH:MM:SS”.

- Usage

```
function Date_Time_Stamp_From_ISO (DTS : String) return String
```

- Example

```

```

1.10.9 Date_Time_Milli_Stamp

- Description

  Returns current datetime stamp with trailing milliseconds as:
  “YYYYMMDD-HHMMSS.NNN”.

  Used in v22.Msg logs.

- Usage

```
function Date_Time_Milli_Stamp return String
```

- Example

```

```

1.10.10 Date_Time_Nano_Stamp

- Description

  Returns current datetime stamp with trailing nanoseconds as:
  “YYYYMMDD-HHMMSS.NNNNNNNNN”.

  Useful for precise timings or unique identifier building with date indication.

- Usage

```
function Date_Time_Nano_Stamp return String
```

- Example

```

```

1.10.11 Date_Time_Stamp_To_Date

- Description

  Returns “YYYY-MM-DD” from a date time stamp as “YYYYMMDD-HHMMSS”.

- Usage

```
function Date_Time_Stamp_To_Date (DTS : String) return String
```

- Example

```

```

1.10.12 Date_Time_Stamp_To_ISO

- Description

  Returns “YYYY-MM-DD HH:MM:SS” from a date time stamp as “YYYYMMDD-HHMMSS”.

- Usage

```
function Date_Time_Stamp_To_ISO (DTS : String) return String
```

- Example

```

```

1.10.13 Date_Time_Stamp_From_Local

- Description

  Returns a date time stamp “YYYYMMDD-HHMMSS” from a local “DD?MM?YYYY HH?MM?SS” (? is any character).

- Usage

```
function Date_Time_Stamp_From_Local (DTS : String) return String
```

- Example

```

```

1.10.14 Date_Time_Stamp_To_Local

- Description

  Returns “DD/MM/YYYY HH:MM:SS” from a date time stamp as “YYYYMMDD-HHMMSS” with a customizable date separator.

- Usage

```
function Date_Time_Stamp_To_Local (DTS : String; Sep : String := "/") re-
turn String
```

- Example

```

```

1.10.15 Date_To_ISO

- Description

  Returns “YYYY-MM-DD” from a string date as “DD?MM?YYYY” (? is any character).

- Usage

```
function Date_To_ISO (DTS : String) return String
```

- Example

```
```

1.10.16 Duration_Stamp

- Description

  Returns a duration as HHhMMmSSs since Time. Unlimited hours counter for long uptimes.

- Usage

```
function Duration_Stamp (Time : Ada.Calendar.Time) return String
```

- Example

```
Log.Msg ("Total execution time: " & Prg.Duration_Stamp (Prg.Start_Time));
```

1.10.17 Duration_Stamp_Seconds

- Description

  Returns a duration as seconds since Time.

- Usage

```
function Duration_Stamp_Seconds (Time : Ada.Calendar.Time) return Natural
```

- Example

```
```

1.10.18 Duration_Stamp_Time

- Description

  Returns a formatted HHhMMmSSs String from Time_Seconds.

- Usage

```
function Duration_Stamp_Time (Time_Seconds : Integer) return String
```

- Example

```
Tio.Put_Line ("Total execution time: " & Prg.Duration_Stamp_Time (1646315044));
```

```
13h35m34s
```

1.10.19 Generate_Password

- Description

  Password generation with variable characters space and length. Generates passwords like: x2U5hhIKX7IJt6.

  Password generation with 64 charset ([a-z] + [A-Z] + [0-9] + '_' + '-') and 14 length. Search space size greater than 1,26 x 10^25.

  You can choose one or more sets between:
  - LETTERS_LOWER = abcdefghijklmnopqrstuvwxyz
  - LETTERS_UPPER = To_Upper (LETTERS_LOWER)
  - FIGURES = 0123456789
  - SPECIAL_CHARACTERS = -_

  Without parameters, uses all sets above with a password length of 14 length.

- Usage

```
function Generate_Password (Characters_Space : String := 
                            LETTERS_LOWER &
                            LETTERS_UPPER &
                            FIGURES &
                            SPECIAL_CHARACTERS;
                            Password_Length : Positive := 14)
                            return String
```

- Example

```
Log.Msg (Prg.Generate_Password (FIGURES, 4));

6496
5668
7708
0408
8630
...
```

1.10.20 Get_Handler_Ctrl_C

- Description

  Activate Ctrl-C interrupt handler. If returns On, Ctrl-C is activated and, when pressed, application is properly finalize. If returns Off, Ctrl-C is inhibited and application continue.

- Usage

```
function Get_Handler_Ctrl_C return On_Off;
```

- Example
  Pressing Ctrl-C in console:

```
20230930-120934 - INIT    - MSG - v22 Framework - GUI test program
20230930-120934 - INIT    - MSG - Copyright (C) Sowebio SARL 2020-2023, according to LGPLv3
20230930-120934 - INIT    - MSG - testgui v0.1 - v22 v0.1 - build 2023-09-30 12:08:43

20230930-120934 - INIT    - MSG - TestGui.Init.App > Configuration file testgui.cfg loaded
20230930-120934 - INIT    - MSG - TestGui.On_Connect > Starting Gnoga server

^C ← Ctrl-C Pressed
20230930-120936 - INIT    - MSG - testgui > Ctrl-C detected, program interrupt is inhibited
```

1.10.21 Get_Version

- Description

  Returns formatted program version : “<space>v.minor.major”.

- Usage

```
function Get_Version return String
```

- Example

```
Log.Msg ("Program version:" & Prg.Get_Version);

Program version:  v2.16
```

1.10.22 Get_Version_Major

- Description

  Returns Major version.

- Usage

```
function Get_Version_Major return Natural;
```

- Example

```
Log.Msg (Prg.Get_Version_Major);

2
```

1.10.23 Get_Version_Minor

- Description

  Returns Minor version.

- Usage

```
function Get_Version_Minor return Natural;
```

- Example

```
Log.Msg [Prg.Get_Version_Minor]; P
16
```

1.10.24 Is_User_Not_Root

- Description

  Returns true if program user's not root.

- Usage

```
function Is_User_Not_Root return Boolean
```

- Example

```

```

1.10.25 Name

- Description

  Return program name.

- Usage

```
function Name return String
```

- Example

```
sr@ro8    ~/Seafile/Sowebio/informatique/github/aide/bin > aide
aide
```

1.10.26 Path

- Description

  Return program path.

- Usage

```
function Path return String
```

- Example

```
sr@ro8    ~/Seafile/Sowebio/informatique/github/aide/bin > aide

/home/sr/Seafile/Sowebio/informatique/github/aide/bin
```

1.10.27 Set_Handler_Ctrl_C

- Description

  Activate Ctrl-C interrupt handler. If Switch is On, Ctrl-C is activated and, when pressed, application is properly finalize. If Switch is Off, Ctrl-C is inhibited and application continue.

- Usage

```
procedure Set_Handler_Ctrl_C (Switch : On_Off);
```

- Example
  Pressing Ctrl-C in console:

```
20230921-122701 - INIT    - MSG - v22 Framework - GUI test program
20230921-122701 - INIT    - MSG - Copyright (C) Sowebio SARL 2020-2023, according to LGPLv3
20230921-122701 - INIT    - MSG - testgui v0.1 - v22 v0.1 - build 2023-09-21 12:26:24

20230921-122701 - INIT    - MSG - testgui > testgui.cfg has been loaded
20230921-122701 - INIT    - MSG - Starting Gnoga server
^C ← Ctrl-C Pressed
20230921-122703 - INIT    - MSG - testgui > Ctrl-C detected, finalize application
```

1.10.28 Set_Exit_Status

- Description

  Set errorlevel return code. Each call is cumulative. Four calls with 1, 2, 4 and 8 set 15 ie msb-00001111-lsb. Can be used everywhere in the program without special call at its end.
  Convention : 1 = no or bad command, 128 = runtime exception (8th bit).

- Usage

```
procedure Set_Exit_Status (Code : Natural)
```

- Example

```

```

1.10.29 Set_Version

- Description

  Set program version.

- Usage

```
procedure Set_Version (Major : Natural; Minor : Natural)
```

- Example

```

```

1.10.30 Start_Dir

- Description

  Constant storing current directory at start.

- Usage

```
Start_Dir : constant String
```

- Example

```

```

1.10.31 Start_Time

- Description

  Constant storing Time at program start.

- Usage

```
Start_Time : constant Ada.Calendar.Time
```

- Example

```

```

1.11 Sql – Databases management

1.11.1 Close

- Description

  Close a database or all database(s) when no database object specified.

- Usage

```
procedure Close (DB : in out GSD.Connection'Class)

procedure Close
```

- Example

```
Sql.Close;

20230911-144939 - SQL T1  - MSG - Closing SQLITE database: v22_testapi_1
20230911-144939 - SQL T1  - MSG - Closing SQLITE database: v22_testapi_2
```

1.11.2 Column_Exists

- Description

  Return true if Column_Name exists in Table_Name.
  Return False if Column_Name or Table_Name does not exist.

- Usage

```
function Column_Exists (DB : in out GSD.Connection'Class;
                        Table_Name : String;
                        Column_Name : String) return Boolean
```

- Example

```
Tio.Put ("Column_Exists: ");
Tio.Put_Line (Column_Exists (DB, "test_table", "Existing_Column")); -- Existing column

Column_Exists: True

Tio.Put ("Column_Exists: ");
Tio.Put_Line (Column_Exists (DB, "test_table", "azeazeaze")); -- Non existing column

Column_Exists: False
```

1.11.3 Delete

- Description

  Delete a row in Table_Name specifying a Where_Condition.

- Usage

```
procedure Delete (Database_Name : String;
                  Table_Name : String;
                  Where_Condition : String := "")

procedure Delete (DB : in out GSD.Connection'Class;
                  Table_Name : String;
                  Where_Condition : String)
```

- Example

```
--  Delete row for Number = 1234 in table Cluster

Sql.Delete ("Cluster", "Number = 1234");
```

1.11.4 Escape_String

- Description

  Escapes incompatible or dangerous characters in a query, impacting SQL query and/or SQL dump.

Escape_String should be used as close as possible to the conditions or situations where it is relevant. The rule is to use Escape_String only in SQL queries, in order to avoid duplicate escapes.

- Usage

```
procedure Delete (DB : in out GSD.Connection'Class;
                  Table_Name : String;
                  Where_Condition : String)

Esc. Seq. Character Represented by Sequence
   \0        ASCII NUL character
   \b        Backspace character
   \t        Tab character
   \n        Newline (linefeed) character
   \r        Carriage return character
   \"        Double quote (") character
   \Z        ASCII 26 Control-Z character
   \'        Single quote (') character
   \%        Percentage character
   \\        Backslash character
   \_        Underscore character
```

- Example

```
Query_Result := Sql.Read (Db_Name, Db_Table, Db_Key_Main,
     "WHERE " & Db_Key_Main & " <= '" & Sql.Escape_String (First_Line_Value) & "' " &
     "ORDER BY " & Db_Key_Main & " DESC LIMIT " & Db_List_Length);
```

1.11.5 Get_Config

- Description

  Get configuration Value from Parameter stored in Sys_Config table.
  The Sys_Config table must be already created.
  Returns an empty string if the Sys_Config table or parameter does not exist

- Usage

```
function Get_Config (DB : in out GSD.Connection'Class;
                     Parameter : String) return String
```

- Example

```
-- Get parameter's value 'Schema_Version' (previously set to '0.1')
Sql.Get_Config ("Schema_Version");
```

```
0.1
```

1.11.6 Get_Database_Brand

- Description

  Returns the Database_Type of DB.

- Usage

```
function Get_Database_Brand (DB : in out GSD.Connection'Class)
                             return Database_Brand

type Database_Brand is (None, MySQL, SQLite, Unknown)
```

- Example

```

```

1.11.7 Get_Database_Main

- Description

  Get the main application database. The main database owns Sys_Config, Sys_Schema and Sys_Users system tables.

- Usage

```
function Get_Database_Main return String
```

- Example

```

```

1.11.8 Get_Version

- Description

  Returns MySQL or SQLite database version.

- Usage

```
function Get_Version (DB : in out GSD.Connection'Class) return String
```

- Example

```
Tio.Put_Line (DB, Sql.Get_Version (MySQL_Handle));

MySQL: v10.3.39-MariaDB-0+deb10u1

Tio.Put_Line (DB, Sql.Get_Version (SQLiTE_Handle));
```

```
SQLite: v3.43.0
```

1.11.9 Index_Exists

- Description

  Return true if Index_Name exists for Table_Name.
  Return False if Index_Name or Table_Name does not exist.
  Names are case insensitive for MySQL and case sensitive for SQLite.

- Usage

```
function Index_Exists (DB : in out GSD.Connection'Class;
                       Table_Name : String;
                       Index_Name : String) return Boolean
```

- Example

```
Tio.Put("Index_Exists: ");
Tio.Put_Line (Sql.Index_Exists (+"key")); -- Existing index

Tio.Put("Index_Exists: ");
Tio.Put_Line (Sql.Column_Exists (+"key1")); -- Non existing index

…

Index_Exists: True
Index_Exists: False
```

1.11.10 Insert

- Description

  Create a row in Table_Name with Columns_Values.

  The special character ^ (or constant CD as Column delimiter) is used to separate column/value pairs and the special character ~ (or constant ND as Name/value delimiter) is used to distinguish the name of a column from its value. See example below.

  A non existent Table or Column don't raise exception but an error is logged.

  Text fields that are too long to be saved in a VARCHAR no longer trigger an exception and are truncated at the maximum length of the VARCHAR. Truncations are recorded in the technical log (screen and file).

- Usage

```
procedure Insert (Database_Name : String;
                  Table_Name : String;
                  Columns_Values : String)

procedure Insert (DB : in out GSD.Connection'Class;
                  Table_Name : String;
```

```
                    Columns_Values : String]
```

- Example

```
-- Fill Number with 1234 and Domain with genesix.org in table Cluster
Sql.Insert (DB, "Cluster", "Number~1234" & "^" & "Domain~genesix.org")
Sql.Insert (DB, "Cluster", "Number" & ND & "1234" & CD & "Domain~genesix.org")
```

1.11.11 Last_RowID

- Description

  Returns last existing RowID in Table_Name or 0 if Table_Name does not exist.

- Usage

```
function Last_RowID (DB : in out GSD.Connection'Class;
                     Table_Name : String) return Integer
```

- Example

```
Tio.Put_Line (Sql.Row_Count (DB, "Table_test"));
12
```

1.11.12 Open

- Description

  Open or create a database, passing a schema version for automatic schema update.

  If schema version is upper than the version stored in Database, returns Open_Need_Update as a database schema update is needed.

  URI conforms to RFC 3986 URI. See examples below:

```
MySQL: db:db_name?host=192.168.0.243&port=3306&user=user_name&password=user_password
SQLite:  file:data.db or file:data.db?mode=ro&cache=private
see https://www.sqlite.org/c3ref/open.html for more information about sqlite options.
```

  DB_Version is major.minor format.

  Returns Database_Status.

  Schema_Load listing and Schema_Update are therefore only launched when necessary.

When you first run the program to generate tables in the empty database, pay attention to specify a version number. If this is not specified, the database will be initialized to version 0.1.

To update the database schema in table Cluster with a new column named 'Thingy':

- Increment Sql.Open version parameter from 1.0 to 1.1;
- Add at the appropriate place in the source: Sql.Schema_Load [Sql.Column_Name, "Thingy", "TEXT"].

See example below.

- Usage

```
function Open [DBM : in out GSD.MySQL.Connection;
               URI : String := "";
               Version : String := "";
               Rank : Database_Rank := Main] return Database_Status;

function Open [DBS : in out GSD.SQLite.Connection;
               URI : String := "";
               Version : String := "";
               Rank : Database_Rank := Main] return Database_Status;

type Database_Rank is [Main, Secondary];

type Database_Status is [None,
                         Open_Failed,
                         Open_Failed_Parameter_Invalid,
                         Open_Success,
                         Open_Need_Update];
```

- Example - Before schema update

Version is 1.0. If database does not exist, it will be created.

```
if Sql.Open [DBS, "file:sqlite_test.db", "1.0"] = Open_Need_Update then

    -- Cluster table ------------------------------------------------------
    Sql.Schema_Load [Sql.Table_Name,         "Tbl_Cluster", Comment => "Clusters table"];
    Sql.Schema_Load [Sql.Column_Name,        "Number", "INTEGER", "Cluster number [1] 1...240"];
    Sql.Schema_Load [Sql.Column_Constraint, "Number", "UNIQUE"];
    Sql.Schema_Load [Sql.Table_Constraint,   "Number", "PRIMARY KEY"];

    Sql.Schema_Load [Sql.Column_Name,        "Key_Name",             "TEXT",    "Cluster key name"];
    Sql.Schema_Load [Sql.Column_Name,        "Key_Private",          "BLOB",    "Cluster private key"];
    Sql.Schema_Load [Sql.Column_Name,        "Key_Public",           "BLOB",    "Cluster public key"];
    Sql.Schema_Load [Sql.Column_Name,        "Supervisor_Instance", "INTEGER", "Supervisor instance"];
    Sql.Schema_Load [Sql.Column_Constraint, "Supervisor_Instance", "REFERENCES Tbl_Instance[Number]"];
    Sql.Schema_Load [Sql.Column_Name,        "Comment",              "TEXT",    "Comment"];

    Sql.Schema_Load [Sql.Index_Name,         "Idx_Cluster_Number", "Number"];
    Sql.Schema_Load [Sql.Index_Constraint,   "Idx_Cluster_Number", "UNIQUE"];

    Sql.Schema_Update [DBS];

end if;

Results:

Database sqlite_test needs creation or update
Create Table: Sys_Config - Create Column: Parameter TEXT UNIQUE PRIMARY KEY
CREATE TABLE Sys_Config [Parameter TEXT UNIQUE PRIMARY KEY]
Table: Sys_Config - Create Column: Value TEXT
Table: Sys_Config - Creating Index: Idx_Config_Parameter Parameter
Create Table: Sys_Schema - Create Column: Table_Name TEXT
```

```
CREATE TABLE Sys_Schema (Table_Name TEXT  )
Table: Sys_Schema - Create Column: Column_Name TEXT
Table: Sys_Schema - Create Column: Column_Type TEXT
Table: Sys_Schema - Create Column: Column_Constraint TEXT
Table: Sys_Schema - Create Column: Version TEXT
Table: Sys_Schema - Create Column: Comment TEXT
Create Table: Tbl_Cluster - Create Column: Number INTEGER UNIQUE PRIMARY KEY
CREATE TABLE Tbl_Cluster (Number INTEGER UNIQUE PRIMARY KEY)
Table: Tbl_Cluster - Create Column: Key_Name TEXT
Table: Tbl_Cluster - Create Column: Key_Private BLOB
Table: Tbl_Cluster - Create Column: Key_Public BLOB
Table: Tbl_Cluster - Create Column: Supervisor_Instance INTEGER REFERENCES Tbl_Instance(number)
Table: Tbl_Cluster - Create Column: Comment TEXT
Table: Tbl_Cluster - Creating Index: Idx_Cluster_Number Number UNIQUE
```

Tbl_Cluster	Number	INTEGER	UNIQUE	+1.0	Cluster number (1) 1...240
Tbl_Cluster	Key_Name	TEXT		+1.0	Cluster key name
Tbl_Cluster	Key_Private	BLOB		+1.0	Cluster private key
Tbl_Cluster	Key_Public	BLOB		+1.0	Cluster public key
Tbl_Cluster	Supervisor_Instance	INTEGER	REFERENCES Tbl_Instance(Number)	+1.0	Supervisor instance
Tbl_Cluster	Comment	TEXT		+1.0	Comment

- Example - A schema update is needed

  Version is upgraded to 1.1 and we add the Thingy1 and Thingy2 fields, in bold below:

```
if Sql.Open (DBS, "file:sqlite_test.db", "1.1") = Open_Need_Update then

    -- Cluster table ------------------------------------------------
    Sql.Schema_Load (Sql.Table_Name,         "Tbl_Cluster", Comment => "Clusters table");
    Sql.Schema_Load (Sql.Column_Name,        "Number", "INTEGER", "Cluster number (1) 1...240");
    Sql.Schema_Load (Sql.Column_Constraint, "Number", "UNIQUE");
    Sql.Schema_Load (Sql.Table_Constraint,   "Number", "PRIMARY KEY");

    Sql.Schema_Load (Sql.Column_Name,        "Key_Name",            "TEXT",    "Cluster key name");
    Sql.Schema_Load (Sql.Column_Name,        "Key_Private",         "BLOB",    "Cluster private key");
    Sql.Schema_Load (Sql.Column_Name,        "Key_Public",          "BLOB",    "Cluster public key");
    Sql.Schema_Load (Sql.Column_Name,        "Supervisor_Instance", "INTEGER", "Supervisor instance");
    Sql.Schema_Load (Sql.Column_Constraint, "Supervisor_Instance", "REFERENCES Tbl_Instance(Number)");
    Sql.Schema_Load (Sql.Column_Name,        "Comment",             "TEXT",    "Comment");
    Sql.Schema_Load (Sql.Column_Name,        "Thingy1",             "TEXT",    "Thingy1 created at v1.1");
    Sql.Schema_Load (Sql.Column_Name,        "Thingy2",             "TEXT",    "Thingy2 created at v1.1");

    Sql.Schema_Load (Sql.Index_Name,         "Idx_Cluster_Number", "Number");
    Sql.Schema_Load (Sql.Index_Constraint,   "Idx_Cluster_Number", "UNIQUE");

    Sql.Schema_Update (DBS);

end if;

Results:

Database sqlite_test needs creation or update
Table: Tbl_Cluster - Create Column: Thingy1 TEXT
Table: Tbl_Cluster - Create Column: Thingy2 TEXT
```

Tbl_Cluster	Key_Name	TEXT		+1.1	Cluster key name
Tbl_Cluster	Key_Private	BLOB		+1.1	Cluster private key
Tbl_Cluster	Key_Public	BLOB		+1.1	Cluster public key
Tbl_Cluster	Supervisor_Instance	INTEGER	REFERENCES Tbl_Instance(Number)	+1.1	Supervisor instance
Tbl_Cluster	Comment	TEXT		+1.1	Comment
Tbl_Cluster	Thingy1	TEXT		+1.1	Thingy1 created at v1.1
Tbl_Cluster	Thingy2	TEXT		+1.1	Thingy2 created at v1.1

- Notes

  Although implementation reserves have been provided for, the table or column deletion function has not yet been implemented.

1.11.13 Ping.Start

- Description

  Ping.Start launch task to periodically reset timeout (each hour), avoiding the infamous MySQL error "server has gone away". This does not apply to SQLite databases.

- Usage

```
task Ping is entry Start; end Ping;
```

- Example

```
Put this line at the beginning of application code:

Sql.Ping.Start;
```

1.11.14 Properties

- Description

  Returns database properties record.

- Usage

```
function Properties (DB_Name : String) return Database_Line

type Database_Line is record
     Brand : Database_Brand := None;
     Status : Database_Status := None;
     URI : String;
     Name : String;
     Host : String;
     Port : Natural;
     User : String;
     Password : String;
     File : String;
     Version : String;
     DBM : GSD.MySQL.Connection;
     DBS : GSD.SQLite.Connection;
end record;
```

- Example

```

```

1.11.15 Read

- Description

  Returns an extraction from Table_Name with comma delimited Columns and standard SQL Condition (like WHERE, ORDER BY, LIMIT).

  The extraction is formatted with standard v22 CD constant as Column delimiter and RD constant as Row delimiter.

  Return an empty string if condition is not met.

- Usage

```
function Read (Database_Name : String;
               Table_Name : String;
               Columns : String;
               Condition : String := "") return String

function Read (DB : in out GSD.Connection'Class;
               Table_Name : String;
               Columns : String;
               Condition : String := "") return String
```

- Example

```
Sql.Read (DB, "Cluster", "Number, Domain");
Sql.Read (DB, "Cluster", "Number, Domain", "WHERE Number = 1234");

-- Used in combination with Field_Display

Field_Display (Sql.Read ("Cluster", "Number,Domain"), CD, RD, "Cluster number, Domain
name"));

Cluster number  Domain name
---------------------------
1               domain1
2               domain2
3               domain3
4               domain4
1234            genesix2.org
```

1.11.16 Row_Count

- Description

  Returns counted rows in Table_Name with Options.

  Option:
  - '*' is all rows, included null-ed;
  - 'DISTINCT Column name' counts not null-ed and distinct rows.

- Usage

```
function Row_Count (DB : in out GSD.Connection'Class;
                    Table_Name : String;
```

```
                        Option : String := "*"] return Integer
```

- Example

```
Tio.Put_Line [DB, Sql.Row_Count [DB, "Table_test"]];

12
```

1.11.17 Schema_Load

- Description

  Load a schema. Commands will be executed by Schema_Update in code source order.

- Usage

```
procedure Schema_Load [Command : in Schema_Command := Null_Command;
                       Name : in String := "";
                       Attribute : in String := "";
                       Comment : in String := ""]
```

- Example

```
Sql.Schema_Load [Sql.Table_Name,         "Cluster"];
Sql.Schema_Load [Sql.Column_Name,        "Number",   "INTEGER"];
Sql.Schema_Load [Sql.Column_Constraint,  "Number",   "UNIQUE"];
Sql.Schema_Load [Sql.Table_Constraint,   "Number",   "PRIMARY KEY"];
Sql.Schema_Load [Sql.Column_Name,        "Domain",   "TEXT"];
Sql.Schema_Load [Sql.Column_Name,        "Email",    "TEXT"];
Sql.Schema_Load [Sql.Column_Name,        "Manager",  "INTEGER"];
Sql.Schema_Load [Sql.Index_Name,         "Idx",      "Number"];
Sql.Schema_Load [Sql.Index_Constraint,   "Idx",      "UNIQUE"];
```

1.11.18 Schema_Update

- Description

  Create and delete tables, table constraints, columns, columns constraints, index, index constraints on database schema after loading schema by Schema_Load.

  <<<TODO>>> : implement delete and backup DB before update or delete.

- Usage

```
procedure Schema_Update [DB : in out GSD.Connection'Class]
```

- Example

```
if Sql.Schema_Need_Update ["Sqlite_Update_Test", 0, 1] then

    Sql.Schema_Load [Sql.Table_Name,         "Cluster"];
    Sql.Schema_Load [Sql.Column_Name,        "Number",   "INTEGER"];
    Sql.Schema_Load [Sql.Column_Constraint,  "Number",   "UNIQUE"];
    Sql.Schema_Load [Sql.Table_Constraint,   "Number",   "PRIMARY KEY"];
```

```
    Sql.Schema_Load (Sql.Column_Name,        "Domain",   "TEXT");
    Sql.Schema_Load (Sql.Column_Name,        "Email",    "TEXT");
    Sql.Schema_Load (Sql.Column_Name,        "Manager",  "INTEGER");
    Sql.Schema_Load (Sql.Index_Name,         "Idx",      "Number");
    Sql.Schema_Load (Sql.Index_Constraint,   "Idx",      "UNIQUE");

    Sql.Schema_Update;

end if;
```

1.11.19 Search

- Description

  Return True if Condition verified.

- Usage

```
function Search (Database_Name : String;
                 Table_Name : String;
                 Condition : String) return Boolean

function Search (DB : in out GSD.Connection'Class;
                 Table_Name : String;
                 Condition : String) return Boolean
```

- Example

```
if Sql.Search (DB, "Cluster","Number = 1234") then
   Tio.Put_Line ("Search 'Number = 1234': Found");
end if;
if not Sql.Search (DB, "Cluster","Number = 9999") then
   Tio.Put_Line ("Search 'Number = 9999': Not found");
end if;

Search 'Number = 1234': Found
Search 'Number = 9999': Not found

if Sql.Search (DB, "Cluster","Login = 'sr' ") then
   Tio.Put_Line ("Search 'Login = sr': Found");
end if;

Search 'Login = sr': Found
```

1.11.20 Set_Config

- Description

  Store configuration Parameter and Value to Sys_Config table.
  The new Value will replaced the eventually existing one.
  The Sys_Config table will be created if needed.

- Usage

```
procedure Set_Config (DB : in out GSD.Connection'Class;
                      Parameter : String;
                      Value : String)
```

- Example

```
-- Set ‘0.1´ value in parameter ‘Schema_Version´
Sql.Set_Config (DB, “Schema_Version”, “0.1");
```

1.11.21 Set_Database_Main

- Description

  Set the main application database. The main database owns Sys_Config, Sys_Schema and Sys_Users system tables.

  The default main database name is the name of the program. I.e if the binary program’s name is “testgui”, the default main database name is set to “testgui”.

- Usage

```
procedure Set_Database_Main (Database_Name : String)
```

- Example

```

```

1.11.22 Table_Exists

- Description

  Return true if Table_Name exists.

- Usage

```
function Table_Exists (DB : in out GSD.Connection'Class;
                       Table_Name : String) return Boolean
```

- Example

```
Tio.Put("Table_Exists: ");
Tio.Put_Line (Table_Exists (DB, "test_table")); -- Existing table

Table_Exists: True

Tio.Put("Table_Exists: ");
Tio.Put_Line (Table_Exists (DB, "test_table1")); -- Non existing table

Table_Exists: False
```

1.11.23 Update

- Description

  Update one or more row in Table_Name with Columns_Values specifying a Where_Condition.

  The special character ^ [or constant ND as Name/value delimiter] is used to separate column/value pairs and the special character ~ [or constant CD as Column delimiter] is used to distinguish the name of a column from its value. See example below.

  A non existent Table or Column don't raise exception but an error is logged.

  Text fields that are too long to be saved in a VARCHAR don't trigger an exception and are truncated at the maximum length of the VARCHAR. Truncations are recorded in the technical log [screen and file].

  The SQL clause WHERE is automatically added if Condition is not empty and the SQL clause WHERE is absent.

- Usage

```
procedure Update (Database_Name : String;
                  Table_Name : String;
                  Columns_Values : String;
                  Condition : String)

procedure Update (DB : in out GSD.Connection'Class;
                  Table_Name : String;
                  Columns_Values : String;
                  Condition : String)
```

- Example

```
--  Update Domain column with genesix2.org value for Number = 1234 in table Cluster

Sql.Update (DB, "Cluster", "Domain~genesix2.org", "Number = 1234");
```

1.12 Sys – System management

1.12.1 Command_Path

- Description

  Return full qualified command path or an empty string if not found.

- Usage

```
function Command_Path (Command_Name : String) return VString;
```

- Example

```
Sys.Command_Path ["mc"];

/usr/bin/mc
```

1.12.2 CRC16_Initialize

- Description

  Initialize CRC16 computing with one of theses modes:
  - CCITT_AUG;
  - BUYPASS_VERIPHONE CRC16.

- Usage

```
function Command_Path [Command_Name : String] return VString;
```

- Example

```
Sys.CRC16_Initialize [Sys.CRC16_BUYPASS_VERIPHONE];
```

1.12.3 CRC16_Compute

- Description

  Return the CRC16 of String_Hex.

- Usage

```
function CRC16_Compute [String_Hex : String] return Unsigned_16;
```

- Example

```
With Payload : constant String := [104 bytes long]
"2312031913270025196200070510001D0000000000000510001E0000000000000510001F0000000000000
5100020000000000000";
CRC16_CCITT_AUG mode : result is 56C1h
CRC16_BUYPASS_VERIPHONE : result is BA56h

CRC : Sys.CRC16_Type;
Sys.CRC16_Initialize [Sys.CRC16_BUYPASS_VERIPHONE];
CRC := Sys.CRC16_Compute [Payload];
Tio.Put [" Crc: "];
Tio.Put [To_Hex [Sys.To_Unsigned_16_High_Byte [CRC]]];
Tio.Put [To_Hex [Sys.To_Unsigned_16_Low_Byte [CRC]]];
```

1.12.4 Get_Alloc_Ada

- Description

  Return current and max allocations done from Ada excluding others languages. Format of returned string : Ada Cur: [868] Max: [1600].

- Usage

```
function Get_Alloc_Ada return String;
```

- Example

```
Prg.Get_Alloc_Ada;

Ada Cur:  [ 868 ] Max:  [ 1600 ]
```

1.12.5 Get_Alloc_All

- Description

  Return current and max allocations done from all languages including Ada. Format of returned string: Ada Cur: [868] Max: [1600]. This uses system calls to find out the program's resident size (RSS) information, both the peak and the current size.

- Usage

```
function Get_Alloc_All return String;
```

- Example

```
Sys.Get_Alloc_All;

All Cur:  [ 2514944 ] Max:  [ 2514944 ]
```

1.12.6 Get_Env

- Description

  Returns String value of String environment variable Name

- Usage

```
function Get_Env (Name : String) return VString
```

- Example

```

```

1.12.7 Get_Home

- Description

  Returns HOME path without trailing slash.

- Usage

```
function Get_Home return VString
```

- Example

```
Sys.Get_Home - for user ‘sr´

“/home/sr”
```

1.12.8 Get_Memory_Dump

- Description

  Dump information about memory usage. Size is the number of the biggest memory users we want to show. Report indicates which sorting order is used, depending of the following options:
  - Prg.All_Reports;
  - Prg.Memory_Usage;
  - Prg.Allocations_Count;
  - Prg.Sort_Total_Allocs;
  - Prg.Marked_Blocks;

> You must activate memory monitor with Set_Memory_Monitor before using this function.

- Usage

```
procedure Get_Memory_Dump (Size : Positive;
                           Report_View : Report := Memory_Usage)

Prg.Get_Memory_Dump (1);
```

- Example

  Displaying all report options :

```
Prg.Get_Memory_Dump (1);

Traceback elements allocated:  2480
Validity elements allocated:  1

Ada Allocs: 60608 bytes in 1258 chunks
Ada Free: 60008 bytes in 1248 chunks
Ada Current watermark:  600 in 10 chunks
Ada High watermark:  1600

1 biggest memory users at this time:
```

```
Results include bytes and chunks still allocated
Traceback elements allocated:  2480
Validity elements allocated:  1

Prg.Get_Memory_Dump (1, Prg.Allocations_Count);

Traceback elements allocated:  2798
Validity elements allocated:  1

Ada Allocs: 68456 bytes in 1419 chunks
Ada Free: 67588 bytes in 1405 chunks
Ada Current watermark:  868 in 14 chunks
Ada High watermark:  1600

 1 biggest number of live allocations:
Results include bytes and chunks still allocated
   5.5%:   48   bytes   in   1   chunks   at   0x000000000040C509   0x000000000040C33B
0x000000000043B74A     0x000000000043D42F     0x000000000042B7A7     0x0000000000407090
0x000000000040C2BE 0x0000000000474D27 0x00000000004053C8

Prg.Get_Memory_Dump (1, Prg.Sort_Total_Allocs);

Traceback elements allocated:  3106
Validity elements allocated:  1

Ada Allocs: 75816 bytes in 1573 chunks
Ada Free: 74948 bytes in 1559 chunks
Ada Current watermark:  868 in 14 chunks
Ada High watermark:  1600

 1 biggest number of allocations:
Results include total bytes and chunks allocated,
even if no longer allocated - Deallocations are ignored

Prg.Get_Memory_Dump (1, Prg.Marked_Blocks);

Traceback elements allocated:  3414
Validity elements allocated:  1

Ada Allocs: 83192 bytes in 1727 chunks
Ada Free: 82324 bytes in 1713 chunks
Ada Current watermark:  868 in 14 chunks
Ada High watermark:  1600

Special blocks marked by Mark_Traceback
  0.0%:  0 chunks / 1 at 0x000000000040C509 0x000000000040C33B 0x000000000043B74A
0x000000000043DB1E     0x00000000004126A5     0x000000000041AC80     0x000000000041ED3D
0x0000000000405B71 0x000000000040C2BE 0x0000000000474D27 0x00000000004053C8
```

1.12.9 Get_System_Name

- Description

  Returns system name like "Debian" or "Ubuntu" or "System not handled (unprocessed system string returned)".

- Usage

```
function Get_System_Name return String;
```

- Example

```
Sys.Get_System_Name - for system "Debian 11 GNU/Linux 11"
```

```
“debian”
```

1.12.10 Get_System_Version

- Description

  Returns system version like 10, 11 for Debian or 18.04, 20.04, 22.04 for Ubuntu or "System not handled [unprocessed system string returned]". For Ubuntu systems, sub-version like 18.04**.6** and **LTS** string are omitted.

- Usage

```
function Get_System_Version return String;
```

- Example

```
Sys.Get_System_Version - for system "Debian 11 GNU/Linux 11"
“11”
Sys.Get_System_Version - for system "Ubuntu 22.04 LTS"
“22.04”
```

1.12.11 Install_Packages

- Description

  Install packages for Debian, Ubuntu or derivatives distributions.

  Can be used for local or remote host either.

  Packages_List is comma separated.

- Usage

```
function Install_Packages (Packages_List : String;
                           Host_Name : VString := "") return Boolean;

function Install_Packages (Packages_List : VString;
                           Host_Name : VString := "") return Boolean;
```

- Example

```
if not Sys.Install_Packages ("curl, libtool, libcurl4, libcurl4-openssl-dev, libssl-
dev") then
   Log.Err ("At least one package has not been installed.");
end if;
```

1.12.12 Is_Command

- Description

  Return true if command exists and reachable from path.

- Usage

```
function Is_Command [Package_Name : String] return Boolean
```

- Example

```
if not Sys.Is_Command ["mc"] then
   Log.Err ["Midnight Commander not available."];
end if;
```

1.12.13 Is_Package

- Description

  Return true if Package_Name is installed.

  Can be used for local or remote host either.

- Usage

```
function Is_Package [Package_Name : String; Host_Name : String := ""]
```

- Example

```
if not Sys.Is_Package ["curl"] then
   Log.Err ["Package Curl is missing."];
end if;
```

1.12.14 Is_Packages

- Description

  Return true if all packages in Package_List is installed.

  Can be used for local or remote host either.

  Packages_List is comma separated.

- Usage

```
function Is_Packages [Package_List : String; Host_Name : String := ""]
```

- Example

```
if not Sys.Is_Packages ["curl, postfix"] then
```

```
   Log.Err ("Package Curl is missing.");
end if;
```

1.12.15 Purge_Packages

- Description

  Purge packages for Debian, Ubuntu or derivatives distributions.

- Usage

```
function Purge_Packages (Packages_List : String;
                         Host_Name : String := "") return Boolean;
```

- Example

```
if not Sys.Purge_Packages ("exim4-base, exim4-config, exim-4-daemon-light") then
   Log.Err ("At least one package has not been purged.");
end if;
```

1.12.16 Reset_Memory_Monitor

- Description

  Reset all internal data (i.e. reset all displayed counters. This is in general not needed, unless you want to know what memory is used by specific parts of your application.

  You must activate memory monitor with Set_Memory_Monitor before using this function.

- Usage

```
procedure Reset_Memory_Monitor
```

- Example

```
Sys.Reset_Memory_Monitor;
```

1.12.17 Set_Env

- Description

  Set an environment variable Name.

- Usage

```
procedure Set_Env (Name : String;  Value : String)
```

- Example

```
```

```
```

1.12.18 Set_Memory_Monitor

- Description

  If Activate_Monitor is true, the program will monitor all memory allocations and deallocations, and through the Get_Memory_Dump procedure below be able to report the memory usage. The overhead is almost null when the monitor is disabled.

- Usage

```
procedure Set_Memory_Monitor (Switch : On_Off)
```

- Example

  Activate memory monitor :

```
Prg.Set_Memory_Monitor (On);
```

  Disable memory monitor :

```
Prg.Set_Memory_Monitor (Off);
```

1.12.19 Shell_Execute

- Description

  Executes shell command. Return the exit code if passed from the executed command. Without Output parameter, the command console output is displayed by default but can be redirected. If Output is used, then the executed command output is return in this parameter.

- Usage

```
procedure Shell_Execute (Command : String)

procedure Shell_Execute (Command : String;
                         Result : out Integer)

procedure Shell_Execute (Command : String;
                         Result : out Integer;
                         Output : out String)
```

- Example

```
---------------------
declare
   SE_Result : Integer := 0;
begin
   Sys.Shell_Execute ("find test.cfg", SE_Result);
   Tio.Put_Line(SE_Result);
```

```
   Tio.Line;
end;

0 <- found

---------------------
declare
   SE_Result : Integer := 0;
begin
   Sys.Shell_Execute ["find i.dont.exist", SE_Result];
   Tio.Put_Line[SE_Result];
   Tio.New_Line;
end;

1 <- not found

---------------------
declare
   SE_Result : Integer := 0;
   SE_Output : VString := "";
begin
   Sys.Shell_Execute ["cat test.cfg", SE_Result, SE_Output];
   if SE_Result = 0 then
      Tio.Put_Line [SE_Output];
      Tio.Line;
   end if;
end;

[Section_1]
Parameter_11 = Value_11
[Section_2]
Parameter_21 = Value_21
[Section_3]
Parameter_31 = Value_31

...which is the content of test.cfg.
```

1.12.20 To_Unsigned_16

- Description

  Assemble an Unsigned_16 from two High and Low bytes.
  Return an Unsigned_16 from a String type of one character.
  Return an Unsigned_16 from a Character type.

- Usage

```
function To_Unsigned_16 [High, Low : Unsigned_8] return Unsigned_16

function To_Unsigned_16 [String_In : String] return Unsigned_16

function To_Unsigned_16 [Char_In : Character] return Unsigned_16
```

- Example

```

```

1.12.21 To_Unsigned_16_High_Byte

- Description

  Return the High Byte of a Unsigned_16.

- Usage

```
function To_Unsigned_16_High_Byte (Word : Unsigned_16) return Unsigned_8
```

- Example

```

```

1.12.22 To_Unsigned_16_Low_Byte

- Description

  Return the Low Byte of a Unsigned_16.

- Usage

```
function To_Unsigned_16_Low_Byte (Word : Unsigned_16) return Unsigned_8
```

- Example

```

```

1.12.23 To_Unsigned_8

- Description

  Return an Unsigned_8 from a String type of one character.
  Return an Unsigned_8 from a Character type.

- Usage

```
function To_Unsigned_8 (String_In : String) return Unsigned_8

function To_Unsigned_8 (Char_In : Character) return Unsigned_8
```

- Example

```

```

1.13 Tio – Text console management

```
Max_Row  : constant Natural := 24;
Max_Column : constant Natural := 79;

subtype Row is Natural range 0..Max_Row;
```

```
subtype Column is Natural range 0..Max_Column;
```

1.13.1 Animated_Delay

- Description

  Animated delay in seconds with markers each 5 and 10 seconds.

- Usage

```
procedure Animated_Delay (Delay_Seconds : Positive);
```

- Example

```
Animated_Delay (27);

....!....|....!....|../ < animated wheel with /-\|/-| characters
.1s !5s  |10s

When finished

....!....|....!....|....!..
```

1.13.2 Beep

- Description

  Send a beep through the console.

- Usage

```
procedure Beep
```

- Example

```

```

1.13.3 Bell

- Description

  If running with user rights (no root) and player *paplay* and *freedesktop soundfile complete.oga* exists, send a bell sound, otherwise call Beep.

```
Pulseaudio paplay & freedesktop sounds come as standard on Ubuntu (and
derivatives) 18.04 LTS and 22.04 LTS. Running paplay with root rights
fails with error: pa_context_connect() failed: Connection refused.
```

- Usage

```
procedure Bell
```

- Example

```
```

1.13.4 Clear_Screen

- Description

  Clear the screen.

- Usage

```
procedure Clear_Screen
```

- Example

```
-- Clear the screen

Clear_Screen;
```

1.13.5 Confirm_Twice

- Description

  Double check by user before action. Returns True if user has validate. To confirm the action, the user must press at first ‘y’ or ‘Y’ and then confirm with ‘c’ or ‘C’.

  User_Prompt_1 automatically ends with " 'y/n' ?".
  User_Prompt_2 automatically ends with " by pressing 'c'".

- Usage

```
function Confirm_Twice (User_Prompt_1 :
                        String ; User_Prompt_2 : String) return Boolean;
```

- Example

```
if Tio.Confirm_Twice (Name & ".Create: Do you want to create the " & Name,
                      Name & ".Create: Confirm creation of " & Name) then
        … Some operations …
   Tio.Put_Line (Name & ".Create: " & Name & " " & Value & " created");
end if;
```

1.13.6 Cursor_Line_Backward

- Description

  Move the cursor backward X rows.

- Usage

```
procedure Cursor_Line_Backward (X : Row)
```

- Example

```
```

1.13.7 Cursor_Line_Erase

- Description

  Erase the current line from the current cursor position to the end of the line.

- Usage

```
procedure Cursor_Line_Erase (X : Row)
```

- Example

```
```

1.13.8 Cursor_Line_Forward

- Description

  Move the cursor forward X rows.

- Usage

```
procedure Cursor_Line_Forward (X : Row)
```

- Example

```
```

1.13.9 Cursor_Move

- Description

  Move the cursor at the specified X,Y coordinates.

- Usage

```
procedure Cursor_Move (X : Row; Y : Column)
```

- Example

```
```

1.13.10 Cursor_Restore

- Description

  Restore the previous saved cursor position.

- Usage

  procedure Cursor_Restore

- Example

```
```

1.13.11 Cursor_Save

- Description

  Save the current cursor position.

- Usage

```
procedure Cursor_save
```

- Example

```
```

1.13.12 New_Line

- Description

  Create a new blank line, or more than one when Spacing is passed.

- Usage

```
procedure New_Line (Spacing : Positive)
```

- Example

```
```

1.13.13 Get_Ansi

- Description

  Get ANSI state for v22 display functions and procedures, including Msg package.

- Usage

```
function Ansi return Boolean;
```

- Example

```
```

1.13.14 Get_Immediate

- Description

  Get a character validated by [Enter]

- Usage

```
procedure Get_Immediate (C : out Character)
```

- Example

```
```

1.13.15 Get_Line

- Description

  Get a string validated by [Enter]

- Usage

```
procedure Get_Line return String
```

- Example

```
```

1.13.16 Get_Password

- Description

  Returns a password blind typed.

- Usage

```
function Get_Password return String
```

- Example

```
Pass := Tio.Get_Password;

Password:
```

1.13.17 Pause

- Description

  Displays *Press any key to continue or [Ctrl-C] to abort...* waiting for user input.

- Usage

```
procedure Pause
```

- Example

```
procedure Test_Pause is
begin
  Pause;
end Test_Pause;
```

1.13.18 Put

- Description

  Print to the console.

- Usage

```
procedure Put (B : Boolean)

procedure Put (B : On_Off)

procedure Put (C : Character)

procedure Put (V : String)

procedure Put (I : Integer)

procedure Put (I : Long_Integer)

procedure Put (I : Long_Long_Integer)

procedure Put (M : Money)
```

- Example

```

```

1.13.19 Put_Line

- Description

  Print to the console then add a new line.

- Usage

```
procedure Put_Line (B : Boolean)

procedure Put_Line (B : On_Off)

procedure Put_Line (C : Character)
```

```
procedure Put_Line (V : String)

procedure Put_Line (I : Integer)

procedure Put_Line (I : Long_Integer)

procedure Put_Line (I : Long_Long_Integer)

procedure Put_Line (M : Money)
```

- Example

```

```

1.13.20 Set_Ansi

- Description

  Set ANSI state for v22 display functions and procedures, including Msg package.

- Usage

```
procedure Set_Ansi (Switch : On_Off);
```

- Example

```

```

1.13.21 Set_Cursor

- Description

  Display or hide the cursor console.

- Usage

```
procedure Set_Cursor (Switch : On_Off)
```

- Example

```
procedure Set_Cursor (Off);
```

1.14 Tio – Text files management

subtype File is Ada.Text_IO.File_Type;
Copy_Form : constant String := "preserve=no_attributes,mode=overwrite";

1.14.1 Append

- Description

  Append on an existing file.

File mode is “Out” (write mode).

- Usage

```
procedure Append (Handle : in out File; Name : String)
```

- Example

```
Tio.Append (File_Tmp_Handle, "./toto");
while not Tio.End_Of_File (File_Tmp_Handle) loop
  Tio.Put_Line (File_Tmp_Handle, "This is a new line of data");
end loop;
Tio.Close (File_Tmp_Handle);
```

1.14.2 Close

- Description

Close a file.

- Usage

```
procedure Close (Handle : in out File)
```

- Example

```
Tio.Open (File_Tmp_Handle, "./toto");
while not Tio.End_Of_File (File_Tmp_Handle) loop
  Tio.Get_Line (File_Tmp_Handle, Line_Buffer);
end loop;
Tio.Close (File_Tmp_Handle);
```

1.14.3 Create

- Description

Create a file.
File mode is “Out” (write mode).

- Usage

```
procedure Create (Handle : in out File; Name : String)
```

- Example

```
.../...
```

```
   File_Tmp_Handle :  Tio.File;

begin

   Tio.Create (File_Tmp_Handle, "./toto");

   Tio.Put_Line (File_Tmp_Handle, "Write a first line in ./toto");
   Tio.Put_Line (File_Tmp_Handle, "Write a second line in ./toto");

   Tio.Close (File_Tmp_Handle);

…/…
```

1.14.4 End_Of_File

- Description

  Return true if end of file is reached.

- Usage

```
function End_Of_File (Handle : File) return Boolean
```

- Example

```

```

1.14.5 End_Of_Line

- Description

  Return true if end of line is reached.

- Usage

```
function End_Of_Line (Handle : File) return Boolean
```

- Example

```

```

1.14.6 Flush

- Description

  Flush file buffer to disk.

- Usage

```
procedure Flush (Handle : in File)
```

- Example

```

```

1.14.7 Get_Line

- Description

  Get the current line and then move the file pointer to the next line.

- Usage

```
function Get_Line (Handle : File) return String
procedure Get_Line (Handle : File; V : out String)
```

- Example

```
Tio.Create (File_Tmp_Handle, "./toto");
while not Tio.End_Of_File (File_Tmp_Handle) loop
  Tio.Get_Line (File_Tmp_Handle, Line_Buffer);
end loop;
Tio.Close (File_Tmp_Handle);
```

1.14.8 Is_Open

- Description

  Returns true if Handle file is open.

- Usage

```
function Is_Open (Handle : in File) return Boolean
```

- Example

```

```

1.14.9 New_Line

- Description

  Create a new blank line, or more when Spacing is passed.

- Usage

```
procedure  New_Line (Handle : File; Spacing : Positive :=1 )
```

- Example

```

```

1.14.10 Open_Conf

- Description

  Special Open function for config files and others valuable files.

  Ensure that the complete directory tree structure exists before creating file. Creating this directory tree if needed. Creates or Append files if needed.

  Always make backup before Append. If Wipe_Before_Process is True, the file Name is backup-ed before being deleted.

- Usage

```
procedure Open_Conf (Handle : in out File;
                     Name : String;
                     Wipe_Before_Process : Boolean := False;
                     Permissions : VString := "");
```

- Example

```
…/…

   File_Tmp_Handle : Tio.File;

begin

   Tio.Open_Conf (File_Tmp_Handle, +"./toto", True, +"0600");

   Tio.Put_Line (File_Tmp_Handle, "Write a first line in ./toto");
   Tio.Put_Line (File_Tmp_Handle, "Write a second line in ./toto");

   Tio.Close (File_Tmp_Handle);

    …/…
```

1.14.11 Open_Read

- Description

  Open a file. File mode is "In" (read mode).

  ➪ For UTF-8 encoded text files, see Uxf.Open_Read function.

- Usage

```
procedure Open_Read (Handle : in out File; Name : String)
```

- Example

```
…/…

   File_Tmp_Handle : Tio.File;

begin
```

```
Tio.Open_Read (File_Tmp_Handle, "./toto");
while not Tio.End_Of_File (File_Tmp_Handle) loop
   Tio.Get_Line (File_Tmp_Handle, Line_Buffer);
end loop;
Tio.Close (File_Tmp_Handle);
 .../...
```

1.14.12 Put

- Description

  Write to a file.

- Usage

```
procedure Put (Handle : File; C : Character)
procedure Put (Handle : File; S : String)
```

- Example

```

```

1.14.13 Put_Line

- Description

  Write a file and then add a new line.

- Usage

```
procedure  Put_Line (Handle : File; C : Character)
procedure  Put_Line (Handle : File; S : String)
```

- Example

```

```

1.14.14 Read_File

- Description

  Read a text file File_To_Read and returning a String buffer. LF (line feed) are preserved.

- Usage

```
function Read_File (File_Name : String) return String
```

- Example

```
```

1.14.15 Reset

- Description

  Reset the file pointer to the start of the file.

- Usage

```
procedure Reset (Handle : in out File)
```

- Example

```
```

1.14.16 Write_File

- Description

  Write a text file File_To_Write with Content. LF in content are preserved and used as line feed. Read Open_Conf documentation for implementation details.

- Usage

```
procedure Write_File (File_Name : String;
                      Content : String;
                      Permissions : String := "")
```

- Example

```
```

1.15 Uxf – UTF-8 UXStrings text files extensions

subtype File is XStrings-Text_IO.File_Type;
Copy_Form : constant String := "preserve=no_attributes,mode=overwrite";

1.15.1 Append

- Description

  Append on a existing file.
  File mode is “Out” [write mode].

- Usage

```
procedure Append (Handle : in out File; Name : String)
```

- Example

```
Uxf.Append (File_Tmp_Handle, "./toto");
while not Uxf.End_Of_File (File_Tmp_Handle) loop
  Uxf.Put_Line (File_Tmp_Handle, "This is a new line of data");
end loop;
Uxf.Close (File_Tmp_Handle);
```

1.15.2 Close

- Description

  Close a file.

- Usage

```
procedure Close (Handle : in out File)
```

- Example

```
Uxf.Open (File_Tmp_Handle, "./toto");
while not Uxf.End_Of_File (File_Tmp_Handle) loop
  Uxf.Get_Line (File_Tmp_Handle, Line_Buffer);
end loop;
Uxf.Close (File_Tmp_Handle);
```

1.15.3 Create

- Description

  Create a file.
  File mode is "Out" (write mode).

- Usage

```
procedure Create (Handle : in out File; Name : String)
```

- Example

```
.../...
   File_Tmp_Handle : Tio.File;
begin
   Uxf.Create (File_Tmp_Handle, "./toto");
```

```
   Uxf . Put_Line (File_Tmp_Handle, "Write a first line in ./toto");
   Uxf. Put_Line (File_Tmp_Handle, "Write a second line in ./toto");

   Uxf. Close (File_Tmp_Handle);

…/…
```

1.15.4 End_Of_File

- Description

  Return true if end of file is reached.

- Usage

```
function End_Of_File (Handle : in out File) return Boolean
```

- Example

```

```

1.15.5 End_Of_Line

- Description

  Return true if end of line is reached.

- Usage

```
function End_Of_Line (Handle : in out File) return Boolean
```

- Example

```

```

1.15.6 Flush

- Description

  Flush file buffer to disk.

- Usage

```
procedure Flush (Handle : in File)
```

- Example

```

```

1.15.7 Get_Line

- Description

  Get the current line and then move the file pointer to the next line.

- Usage

```
function Get_Line (Handle : File) return String
procedure Get_Line (Handle : File; V : out String)
```

- Example

```
Uxf.Create (File_Tmp_Handle, "./toto");
while not Uxf.End_Of_File (File_Tmp_Handle) loop
  Uxf.Get_Line (File_Tmp_Handle, Line_Buffer);
end loop;
Uxf.Close (File_Tmp_Handle);
```

1.15.8 Is_Open

- Description

  Returns true if Handle file is open.

- Usage

```
function Is_Open (Handle : in File) return Boolean
```

- Example

```

```

1.15.9 New_Line

- Description

  Create a new blank line, or more when Spacing is passed.

- Usage

```
procedure  New_Line (Handle : File; Spacing : Positive := 1)
```

- Example

```

```

1.15.10 Open_Read

- Description

  Open a file. File mode is “In” (read mode).

  Maximum file length is one petabyte (1024 TB).

- Usage

```
type Encoding_Scheme is (ASCII_7, Latin_1, UTF_8, UTF_16BE, UTF_16LE)

procedure Open_Read (Handle : in out File; Name : String; Scheme : Encod-
ing_Scheme := UTF_8)
```

- Example

```
…/…
   File_Tmp_Handle : Uxf.File;
begin
   Uxf.Open_Read (File_Tmp_Handle, “./toto”);
   while not Uxf.End_Of_File (File_Tmp_Handle) loop
      Uxf.Get_Line (File_Tmp_Handle, Line_Buffer);
   end loop;
   Uxf.Close (File_Tmp_Handle);
    …/…
```

1.15.11 Put

- Description

  Write to a file.

- Usage

```
procedure Put (Handle : File; C : Character)

procedure Put (Handle : File; S : String)
```

- Example

```

```

1.15.12 Put_Line

- Description

  Write a file and then add a new line.

- Usage

```
procedure  Put_Line (Handle : File; S : String)
```

- Example

```

```

1.15.13 Reset

- Description

  Reset the file pointer to the start of the file.

- Usage

```
procedure Reset (Handle : in out File)
```

- Example

```

```

1.16 Uxs – UTF-8 UXStrings extensions

Uxs is a package extending UXString.

Some Uxs original routines has been integrated in UXStrings.

Null_String : Null_UXString

1.16.1 Char_Count

- Description

  Count each char in String_To_Process relative to Char_Set_Pattern.

- Usage

```
function Char_Count (String_To_Process : String;
                     Char_Set_Pattern : String) return Integer;
```

- Example

```
Tio.Put_Line (Uxs.Char_Count ("alpha", "ap"));
```

```
3
```

1.16.2 Ends_With

- Description

  Check if String Item ends with another String Pattern.

- Usage

```
function Ends_With (Item : String; Pattern : Character) return Boolean;
```

- Example

```
- Check String with String pattern
if Uxs.Ends_With ("package", "age") then
  Tio.Put_Line ("Match !");
end if;
```

1.16.3 Field_* guidelines

Field_* functions deal with string (String_To_Process) forming lists of fields separated by a delimiting character (Field_Delimiter).

Use only Field_Delimiter characters between 0dec and 127dec, due to some keyboard available characters encoding with 2 chars.

Some recommended Field_Delimiter characters are listed in v20.ads but also above in the v20 documentation : Delimiter characters.

1.16.4 Field_By_Index

- Description

  Return a field indexed by Index_Field and delimited by Field_Delimiter.

- Usage

```
function Field_By_Index (String_Input : String;
                         Index_Field : Integer;
                         Field_Delimiter : String) return String;
```

- Example

```
Tio.Put_Line (Uxs.Field_By_Index ("alpha:bravo:charlie", 2, ":"));

bravo
```

1.16.5 Field_By_Name

- Description

  Return a field from a search string and delimited by Field_Delimiter. Returns an empty String if not found.

- Usage

```
function Field_By_Name (String_Input : String;
                        Field_To_Search : String;
                        Field_Delimiter : String) return String;
```

- Example

```
Tio.Put_Line (Uxs.Field_By_Name ("alpha:bravo:charlie", "rav", ":"));

bravo
```

1.16.6 Field_Count

- Description

  Count fields in String_To_Process and return fields number.

To handle one field case without trailing Field_Delimiter, if String_To_Process not empty and Field_Delimiter not found, Field_Count returns 1.

- Usage

```
function Field_Count (String_To_Process : String;
                      Field_Delimiter : String) return Integer;
```

- Example

```
Tio.Put_Line (uxs.Field_Count ("alpha:bravo:charlie", ":"));

3
```

1.16.7 Field_Included

- Description

  Returns True if all Items_List are included in String_To_Process list, which is delimited by Field_Delimiter.

- Usage

```
function Field_Included (String_To_Process : String;
                         Items_List  : string;
                         Field_Delimiter : String) return Boolean;
```

- Example

```
Tio.Put_Line (Uxs.Field_Count ("alpha,bravo,charlie", "alpha,charlie" , ",")];

True
```

1.16.8 Field_Display

- Description

  Formatted display of a string fields structured in rows and columns. Optional header names are separated by commas.

  Constants declaration abstract in v22 (related to Field_* functions) :

```
ND : constant String := "~"; -- Name/value delimiter
CD : constant String := "^"; -- Column delimiter
RD : constant String := "\"; -- Row delimiter
```

- Usage

```
procedure Field_Display (String_To_Process : String;
                         Column_Delimiter : String;
                         Row_Delimiter : String;
                         Custom_Header : String := "");
```

- Example

  Combined example with Uxs.Field_Display and Sql.Read functions :

```
Uxs.Field_Display (Sql.Read ("Cluster", "Number,Domain"), CD, RD, "Cluster number, Do-
main name");

Cluster number  Domain name
----------------------------
1               domain1
2               domain2
3               domain3
4               domain4
1234            genesix2.org
```

1.16.9 Field_Get_Data

- Description

  Return space trimmed datas from Field_Name identifier in Datas or an empty string if Field_Name not found.

- Usage

```
function Field_Get_Data (Datas : String ;
                         Field_Name : String) return String
```

- Example

```
Datas : GET<20>/24010C75---PAYLOAD---C47C/%20HTTP/1.1<20>HTTP/1.1<0D><0A>
        X-Forwarded-For:<20>172.31.0.2<0D><0A>
        Host:<20>localhost:8001<0D><0A>
        Connection:<20>close<0D><0A>
        <0D><0A>

Tio.Put_Line (Field_Get_Data (Received_Datas, "X-Forwarded-For:"));

172.31.0.2
```

1.16.10 Field_Included

- Description

Returns True if all Items_List are included in String_To_Process list, which is delimited by Field_Delimiter.

- Usage

```
function Field_Included (String_To_Process : String ;
                         Items_List : String ;
                         Field_Delimiter : String) return Boolean
```

- Example

```

```

1.16.11 Field_Search

- Description

Search Field_To_Search in String_Input and return True if found.

Behavior : "a" should not be found in "span", "0" should not be found in "01" and "1" should not be found un "01". “span” and “01” should be found respectively.

- Usage

```
function Field_Search (String_Input : String;
                       Field_To_Search : String;
                       Field_Delimiter : String) return Boolean;
```

- Example

```
Tio.Put_Line (Uxs.Field_Search ("alpha:bravo:charlie", "bravo", ":"));
```

```
True
```

1.16.12 From_DB_To_Date_String

- Description

  Converts a ISO 8601 YYYY-MM-DD string to DD/MM/YYYY string with optional separator replacement.

- Usage

```
function From_DB_To_Date_String (DB_Value : String; Separator : String := "/") return String
```

- Example

```

```

1.16.13 From_DB_To_Money

- Description

  Converts a String (as an image of type Bigint in database) into a Money type.

  Bigint (Long_Long_Integer) is used for accurate storage in database.

- Usage

```
function From_DB_To_Money (DB_Value : String) return Money
```

- Example

```

```

1.16.14 From_DB_To_Money_String

- Description

  Converts a String (as an image of type Bigint in database) into a String (as an image of type Money).

  Bigint (Long_Long_Integer) is used for accurate storage in database.

- Usage

```
function From_DB_To_Money_String (DB_Value : String) return String
```

- Example

```
000    =>     0.00
001    =>     0.01
```

```
 012    =>    0.12
 12312  =>  123.12
-001    =>   -0.01
-012    =>   -0.12
-12312  => -123.12
```

1.16.15 From_DB_To_Money_String_With_Padding_Sign

- Description

  Converts a String (as an image of type Bigint in database) into a String (as an image of type Money).

  Bigint (Long_Long_Integer) is used for accurate storage in database.

  With positive and invisible padding sign to keep vertical alignment.

- Usage

```
function From_DB_To_Money_String_With_Padding_Sign (DB_Value : String)
return String
```

- Example

```
 000    =>    0.00
 001    =>    0.01
 012    =>    0.12
 12312  =>  123.12
-001    =>   -0.01
-012    =>   -0.12
-12312  => -123.12
```

1.16.16 From_Money_To_DB

- Description

  Converts a Money type or a String type (as an image of type Money) into a String compatible with storage as a Bigint in database.

  Bigint (Long_Long_Integer) is used for accurate storage in database.

- Usage

```
function From_Money_To_DB (DB_Value : Money) return String
function From_Money_To_DB (DB_Value : String) return String
```

- Example

```
123         =>  12300
123.        =>  12300
123.1       =>  12301
123.12      =>  12312
123.123456  =>  12312
   .1       =>  001
```

```
   .12     =>  012
   .123456 =>  012
 -0.123456 => -012
  -.123456 => -012
  -.       =>  000
  -        =>  000
```

1.16.17 Is_Numeric

- Description

  Returns True if Item string is numeric (i.e. contains only digits with or without leading signs like space, plus or minus).

- Usage

```
function Is_Numeric (Item : in String; Signs : String := "") return Boolean
```

- Example

```
tio.Put_Line (Is_Numeric ("12AZE12"));

False

tio.Put_Line (Is_Numeric ("1212"));

True

tio.Put_Line (Is_Numeric ("-1212", "-"));

True

tio.Put_Line (Is_Numeric ("+1212", " -+"));

True
```

1.16.18 Padding_Left

- Description

  Replace all Char_In by Char_Out in String_To_Process.

- Usage

```
function Padding_Left (String_To_Process : String ;
                       Padding_Character : String ;
                       Result_Length : Positive) return String
```

- Example

```
Padding_Left ("12", "0", 6);

"000012"
```

1.16.19 Replace_Char

- Description

  Replace all Char_In by Char_Out in String_To_Process.

- Usage

```
function Replace_Char (String_To_Process : String;
                       Char_In : Character;
                       Char_Out : Character) return String;
```

- Example

```
Replace_Char (ABCDEFGH", 'D', 'Z');

"ABCZEFGH"
```

1.16.20 Starts_With

- Description

  Check if String Item starts with another String Pattern.

- Usage

```
function Starts_With (Item : String; Pattern : Character) return Boolean;
```

- Example

```
- Check String with String pattern
if Ends_With ("package",  "pac") then
  Tio.Put_Line ("Match !");
end if;
```

1.16.21 Stript_Chars

- Description

  Stript each char in String_To_Process relative to Char_List.

- Usage

```
function Stript_Chars (String_To_Process : String ; Char_List : String)
return String
```

- Example

```
Tio.Put_Line (Uxs.Stript_Chars ("ABCDEFGH", "BDF"));

"ACEGH"
```

1.16.22 Tail_After_Match

- Description

  Extract a String from Source starting from Pattern+1 position to the end.
  If Pattern not found, return Source unchanged.

- Usage

```
function Tail_After_Match (Source : String;
                           Pattern : Character) return String;

function Tail_After_Match (Source : String;
                           Pattern : String) return String;
```

- Examples

```
Path := "/etc/genesix/gnx-startup";

Tio.Put_Line (Uxs.Tail_After_Match (Path, '/'));
"gnx-startup"

Put_Line (Uxs.Tail_After_Match (Path, "ix"));
"/gnx-startup"

Put_Line (Uxs.Tail_After_Match (Path, "gene"));
"six/gnx-startup"

Tio.Put_Line (Uxs.Tail_After_Match (Path, "etc/genesix/gnx-startu"));
"p"

Tio.Put_Line (Uxs.Tail_After_Match (Path, "/etc/genesix/gnx-startu"));
"p"

Tio.Put_Line (Uxs.Tail_After_Match (Path, "/etc/genesix/gnx-startup"));
empty string

Tio.Put_Line (Uxs.Tail_After_Match (Path, "/etc/genesix/gnx-startupp"));
empty string

Tio.Put_Line (Uxs.Tail_After_Match (Path, "/etc/geneseven"));
empty string
```

1.16.23 To_Hex

- Description

  Convert a Byte or a String to a String hexadecimal output.

- Usage

```
function To_Hex (Byte : Interfaces.Unsigned_8) return String

function To_Hex (String_In : String) return  String
```

- Example

```
tio.Put_Line [Uxs.To_Hex ["ABCDEF"]];

41 42 43 44 45 46
```

1.16.24 To_Hex_Control_Codes

- Description

  Convert any ASCII character value ranging 0..32 to a hexadecimal output but leave others characters unchanged.

- Usage

```
function To_Hex_Control_Codes [String_To_Convert : String] return String;
```

- Example

```
Msg.Info [To_Hex_Control_Codes ["test " & CRLF]];

test<20><0D><0A>
```

1.16.25 To_Hex_From_Val

- Description

  Convert an ASCII String value ranging 0..127 to a String hexadecimal output.

- Usage

```
function Ascii_Value_To_Hex [Input : String] return String
```

- Example

```
tio.Put_Line [Uxs.To_Hex_From_Val ["61"]];

3D
```

1.16.26 To_Integer

- Description

  Convert a String to an Integer.
  Leading and trailing spaces are trimmed before conversion.
  Returns 0 if String is empty or contains a least one non numeric character.

- Usage

```
function To_Integer [V : String] return Integer
```

- Example

```
tio.Put_Line (Uxs.To_Integer ("22"));

22
```

1.16.27 To_Long_Long_Integer

- Description

  Convert a String to a Long_Long_Integer.
  Leading and trailing spaces are trimmed before conversion.
  Returns 0 if String is empty or contains a least one non numeric character.

- Usage

```
function To_Long_Long_Integer (V : String) return Long_Long_Integer
```

- Example

```
tio.Put_Line (Uxs.To_Long_Long_Integer ("22"));

22
```

1.16.28 To_Long_Integer_From_Hex

- Description

  Convert a hexadecimal string to a Long_Integer.
  Leading and trailing spaces are trimmed before conversion.
  Returns 0 if String is empty or contains non numeric character.

- Usage

```
function To_Long_Integer_From_Hex (Hex_Val : String) return Long_Integer
```

- Example

```
Msg.Info ("Result: " & To_String (To_Long_Integer_From_Hex (""))); 0
Msg.Info ("Result: " & To_String (To_Long_Integer_From_Hex ("01GG"))); 0
Msg.Info ("Result: " & To_String (To_Long_Integer_From_Hex ("015E"))); 350
Msg.Info ("Result: " & To_String (To_Long_Integer_From_Hex ("  015E  "))); 350
Msg.Info ("Result: " & To_String (To_Long_Integer_From_Hex ("5BCD"))); 23501
Msg.Info ("Result: " & To_String (To_Long_Integer_From_Hex ("ADAADADA"))); 2913655514
Msg.Info ("Result: " & To_String (To_Long_Integer_From_Hex ("FFFFFFFF"))); 4294967295
Msg.Info ("Result: " & To_String (To_Long_Integer_From_Hex ("3B9AC9FF"))); 999999999
```

1.16.29 To_Money

- Description

  Convert Bigint database storage (Long_Long_Integer) or String to Money type.

- Usage

```
function To_Money (DB_Integer : Long_Long_Integer) return Money

function To_Money (DB_Value : String) return Money
```

- Example

```

```

1.16.30 To_String

- Description

  Convert a Boolean, an On_Off type, an Integer, a Long Integer, a Long Long Integer, a Money type or a Char into String type.

- Usage

```
function To_String (B : Boolean) return String

function To_String (B : On_Off) return String

function To_String (I : Integer) return String

function To_String (I : Long_Integer) return String

function To_String (I : Long_Long_Integer) return String

unction To_String (I : Money) return String

function To_String (C : ASCII_Character) return String
```

- Example

```
tio. Put_Line (Uxs. To_String (22));
“ 22”
```

1.16.31 To_String_Unsigned

- Description

  Convert an Integer into String type removing the sign, i.e ' ' space for plus and '-' for minus.

- Usage

```
function To_String_Unsigned (I : Integer) return String
function To_String_Unsigned (I : Unsigned_16) return String
```

- Example

```
tio.Put_Line (Uxs.To_String_Unsigned (22));
"22"
```

1.16.32 To_Val

- Description

  Convert a String to String ASCII decimal values formatted output.

- Usage

```
function To_Val (String_To_Convert : String) return  String
```

- Example

```
Tio.Put_Line (Uxs.To_Val ("ABCDEF"));
65 66 67 68 69 70
```

1.16.33 Trim_Both

- Description

  Returns an all trimmed spaces String of String Source.

- Usage

```
function Trim_Both  (Source : String) return String
```

- Example

```
Tio.Put_Line (Uxs.Trim_Right ("   AB CD   "));
"AB CD"
```

1.16.34 Trim_Left

- Description

  Returns a trimmed leading spaces String of String Source.

- Usage

```
function Trim_Left  (Source : String) return String
```

- Example

```
Tio.Put_Line (Uxs.Trim_Left (+"   ABCD   "));
"ABCD   "
```

1.16.35 Trim_Right

- Description

  Returns a trimmed trailing spaces VString of VString Source.

- Usage

```
function Trim_Right (Source : String) return String
```

- Example

```
Tio.Put_Line (Uxs.Trim_Right (+"   ABCD   "));
"   ABCD"
```

1.16.36 Trim_Slashes

- Description

  Returns an all trimmed slashes String of String Source.

- Usage

```
function Trim_Slashes (Source : String) return String
```

- Example

```
Tio.Put_Line (Uxs.Trim_Slashes ("/"));
""

Tio.Put_Line (Uxs.Trim_Slashes ("I"));
"I"

Tio.Put_Line (Uxs.Trim_Slashes ("/i"));
"i"

Tio.Put_Line (Uxs.Trim_Slashes ("//////i/////"));
"i"
```

2 Gnoga

The Gnoga API is too large to be fully reproduced here. Please refer to the automatically generated HTML documentation available from GnatStudio menu.

2.1 Server.Database

2.1.1 Types

```
type Connection is limited interface;
type Connection_Access is access all Connection'Class;
```

```
type Recordset is interface;
type Recordset_Access is access all Recordset'Class;
```

```
type Field_Description is record
      Column_Name   : String;
      Data_Type     : String;
      Can_Be_Null   : Boolean;
      Default_Value : String;
end record;

package Field_Description_Arrays is new Ada.Containers.Indefinite_Vectors (Natural,
Field_Description);
subtype Field_Description_Array_Type is Field_Description_Arrays.Vector;
```

2.1.2 Exceptions

```
Connection_Error : Unable to connect to MYSQL Server or Not connected to Server

Database_Error : Unable to switch to specified Database

Table_Error : Unable to locate table or table has no fields

Query_Error : Unable to execute query

Empty_Recordset_Error : The recordset is currently empty

Empty_Row_Error : Attempt to read value from Row before calling Next

End_Of_Recordset : Attempt to go pass the last row in recordset

No_Such_Field : The value for a field name was requested that does not exits

Null_Field : The value for a Null field was requested

Not_Implemented : If a database method is called that is not implemented by the spe-
cific database engined used this exception will be raised.
```

2.1.3 Affected_Rows

- Description

  Returns the number of rows affected by an Execute_Query.

- Usage

  function Affected_Rows (C : Connection) return Natural

- Example

```
```

2.1.4 Close

- Description

  Close current recordset and free resources.

- Usage

  procedure Close (RS : in out Recordset)

- Example

```
```

2.1.5 Disconnect

- Description

  Disconnect from server.

- Usage

  procedure Disconnect (C : in out Connection)

- Example

```
```

2.1.6 Error_Message

- Description

  Returns the last error message that has occurred on this connection.

- Usage

  function Error_Message (C : Connection) return String

- Example

```
```

2.1.7 Escape_String

- Description

  Prepares a string for safe storage in a query.

- Usage

  function Escape_String (C : Connection; S : String) return String

- Example

```
```

2.1.8 Execute_Query

- Description

  Execute an SQL Query with no result set.

- Usage

  procedure Execute_Query (C : in out Connection; SQL : String)

- Example

```
```

2.1.9 Execute_Update

- Description

  Executes and SQL Query and returns the number of affected row.

- Usage

  function Execute_Update (C : in out Connection; SQL : String) return Natural

- Example

```
```

2.1.10 Field_Decimals

- Description

  Returns the decimal portion of a field size if it exists or 0. For example: if the Data_Type = float(10,2) it will return 2.

- Usage

  function Field_Decimals (Field : Field_Description) return Natural

- Example

```
```

2.1.11 Field_Descriptions

- Description

  Return an array of Field_Description records describe the fields of a table.

- Usage

  function Field_Descriptions (C : Connection; Table_Name : String) return Field_Description_Array_Type

- Example

```
```

2.1.12 Field_Name

- Description

  Return name of field.

- Usage

  function Field_Name (RS : Recordset; Field_Number : Natural) return String

- Example

```
```

2.1.13 Field_Options

- Description

  Returns the field options portion of a data type, for example: If the Field.Data_Type = enum('N','Y') then will return 'N','Y' as this is described in the database in the same way as field size, this may be used for a string representation of the size as well. For example varchar(80) will return the string 80. This is also used for descriptions like decimal(10,2), etc.

- Usage

  function Field_Options (Field : Field_Description) return String

- Example

```
```

2.1.14 Field_Options

- Description

  Returns the field size portion of a data type, for example: If the Field.Data_Type = varchar(80) then will return 80. If the Data_Type does not have a size portion will return 0. If the Data_Type is a numeric with decimals, e.g. decimal(10,2) then it will return the non-decimal portion.

- Usage

  function Field_Size (Field : Field_Description) return Natural

- Example

```
```

2.1.15 Field_Type

- Description

  Returns the field type portion of a data type, for example: If the Field.Data_Type = Varchar(80) then will return Varchar.

- Usage

  function Field_Type (Field : Field_Description) return String

- Example

```
```

2.1.16 Field_Value

- Description

  Return value of field, if Handle_Nulls is true, Null values will return as empty Strings.

- Usage

  function Field_Value (RS : Recordset; Field_Number : Natural; Handle_Nulls : Boolean := True) return String
  function Field_Value (RS : Recordset; Field_Name : String; Handle_Nulls : Boolean := True) return String

- Example

```
```

2.1.17 Field_Values

- Description

  Return map of all values for current row, NULL values are set to an empty String.

- Usage

  function Field_Values (RS : Recordset) return Gnoga.Types.Data_Map_Type

- Example

```
```

2.1.18 ID_Field_String

- Description

  Returns the proper type format for the ID field.
  For MySQL = "id INTEGER PRIMARY KEY AUTO_INCREMENT".
  For SQLite = "id INTEGER PRIMARY KEY AUTOINCREMENT".

- Usage

  function ID_Field_String (C : Connection) return String

- Example

```
```

2.1.19 Insert_ID

- Description

  Returns the last value assigned to an auto increment field upon insert.

- Usage

  function Insert_ID (C : Connection) return Natural

- Example

```
```

2.1.20 Is_Null

- Description

  Return True if value of field is null.

- Usage

  function Is_Null (RS : Recordset; Field_Number : Natural) return Boolean
  function Is_Null (RS : Recordset; Field_Name : String) return Boolean

- Example

```
```

2.1.21 Iterate

- Description

  (1) Iterate through all rows in the result set of the query
  (2) Iterate through all rows in the recordset
  (3) Iterate through all rows in the result set of the query
  (4) Iterate through all rows in the recordset

- Usage

  (1) procedure Iterate (C : in out Connection; SQL : in String; Process : not null access procedure (RS : Recordset'Class))

  (2) procedure Iterate (RS : in out Recordset; Process : not null access procedure (RS : Recordset'Class))

  (3) procedure Iterate (C : in out Connection; SQL : String; Process : not null access procedure (Row : Gnoga.Types.Data_Map_Type))

  (4) procedure Iterate (RS : in out Recordset; Process : not null access procedure (Row : Gnoga.Types.Data_Map_Type))

- Example

```
```

2.1.22 List_Of_Tables

- Description

  Return an array of table names

- Usage

  function List_Of_Tables (C : Connection) return Gnoga.Types.Data_Array_Type

- Example

```
```

2.1.23 List_Fields_Of_Table

- Description

  Return an array of field names for table.

- Usage

  function List_Fields_Of_Table (C : Connection; Table_Name : String) return Gnoga.Types.Data_Array_Type

- Example

```
```

2.1.24 Next

- Description

  Go to next row or Go to next row and return true if not End of Recordset.

- Usage

  procedure Next (RS : in out Recordset)
  function Next (RS : in out Recordset) return Boolean

- Example

```
```

2.1.25 Number_Of_Fields

- Description

  Return number of fields (columns) in recordset.

- Usage

  function Number_Of_Fields (RS : Recordset) return Natural

- Example

```
```

2.1.26 Query

- Description

  Execute query that returns Recordset.

- Usage

  function Query (C : Connection; SQL : String) return Recordset'Class

- Example

```
```

2.2 Application, Types, Gui, Server, Client

The Gnoga framework's root package is Gnoga. There are five child packages making up the five areas of Gnoga development.

```
Gnoga.Application and its children are related to initializing and managing the life
cycle of Gnoga applications.

Gnoga.Types contains Gnoga specific types used throughout the framework.

Gnoga.Gui contains the user-interface portions of Gnoga. It is further divided into:
     - Gnoga.Gui.Base - Common base functionality and events to all UI objects
     - Gnoga.Gui.Document - Binding to root element of DOM in a window
     - Gnoga.Gui.Element - General binding to all UI objects
     - Gnoga.Gui.Element.Common - Common UI elements
     - Gnoga.Gui.Element.Form - Form-related UI elements
     - Gnoga.Gui.Ekement.Canvas - Binding to a drawing canvas
     - Gnoga.Gui.Element.Multimedia - Multimedia bindings
     - Gnoga.Gui.Element.SVG - SVG canvas binding
     - Gnoga.Gui.Location - Browser window location control
     - Gnoga.Gui.Navigator - Browser application control
     - Gnoga.Gui.Screen - Desktop screen properties
     - Gnoga.Gui.View - Layout control of UI elements
     - Gnoga.Gui.Window - Control of connection to UI

Gnoga.Server - Server side bindings and features
     - Gnoga.Server - Application settings and directories
     - Gnoga.Server.Connection - Low level control of connection to UI
     - Gnoga.Server.Database - Database bindings (MySQL and SQLite3)
     - Gnoga.Server.Migration - Database schema migration interface
     - Gnoga.Server.Model - Active Record implementation for Database access
     - Gnoga.Server.Template_Parser - Template parsing (Python or simple text)

Gnoga.Client - Non GUI client side bindings
     - Gnoga.Client.Storage - Local storage on client side
     - Gnoga.Client.Bind_Page - Dynamically create Gnoga objects for an HTML page
```

2.3 Hierarchy for GUI Types

```
Gnoga.GUI.Base.Base_Type
|
|__ Gnoga.Gui.Document.Document_Type
|
|__ Gnoga.Gui.Element.Element_Type
   |
   |__ Gnoga.Gui.Element.Canvas.Canvas_Type
```

```
    |                      Context_Type
    |                      |
    |                      |__ Canvas.Context_2d.Context_2d_Type
    |
    |__ Gnoga.Gui.Element.Common.A_Type
    |                            Button_Type
    |                            Div_Type
    |                            P_Type
    |                            IMG_Type
    |                            HR_Type
    |                            BR_Type
    |                            Meter_Type
    |                            Progress_Bar_Type
    |                            Span_Type
    |
    |__ Gnoga.Gui.Element.Form.Form_Type
    |                      |     Form_Element_Type
    |                      |     Data_List_Type
    |                      |     Text_Area_Type
    |                      |     Hidden_Type
    |                      |     Input_Button_Type
    |                      |     Submit_Button_Type
    |                      |     Reset_Button_Type
    |                      |     Check_Box_Type
    |                      |     Radio_Button_Type
    |                      |     Check_Box_Type
    |                      |     Input_Image_Type
    |                      |     Text_Type
    |                      |     Email_Type
    |                      |     Password_Type
    |                      |     URL_Type
    |                      |     Search_Type
    |                      |     Color_Picker_Type
    |                      |     Date_Type
    |                      |     Time_Type
    |                      |     Month_Type
    |                      |     Week_Type
    |                      |     Date_Time_Type
    |                      |     Date_Time_Local_Type
    |                      |     Number_Type
    |                      |     Range_Type
    |                      |     Label_Type
    |                      |     Selection_Type
    |                      |     Option_Type
    |                      |     Option_Group
    |                      |__ Form.Fieldset.Fieldset_Type
    |
    |__ Gnoga.Gui.Element.IFrame.IFrame_Type
    |
    |__ Gnoga.Gui.Element.List.Ordered_List_Type
    |                            Unordered_List_Type
    |                            List_Item_Type
    |                            Definition_List_Type
    |                            Term_Type
    |                            Description_Type
    |
    |__ Gnoga.Gui.Element.Multimedia.Multimedia_Type
    |                                  Audio_Type
    |                                  Video_Type
    |
    |__ Gnoga.Gui.Element.Phrase.Phase_Type (Abbr, Code, Strong, Em, Dfn, Samp,
    |                                        Kbd, Var, Marked, Del, Ins, S, Q,
    |                                        Big, Small, Time, Tt, Cite, I, B,
    |                                        U, Sub, Sup)
    |
    |__ Gnoga.Gui.Element.Section.Section_Type (Address, Article, Aside, Header,
    |                                           Main, Nav, P, Pre, Section,
    |                                           BlockQuote, H1, H2, H3, H4, H5,
    |                                           H6, HGroup)
```

```
   |
   |__ Gnoga.Gui.Element.Style.Style_Block.Style_Type
   |
   |__ Gnoga.Gui.Element.SVG.SVG_Type
   |
   |__ Gnoga.Gui.Element.Table.Table_Type
   |                           Table_Row_Type
   |                           Table_Column_Type
   |                           Table_Heading_Type
   |                           Table_Header_Type
   |                           Table_Body_Type
   |                           Table_Footer_Type
   |                           Table_Group_Type
   |                           Table_Column_Type
   |
   |__ Gnoga.Gui.Location.Location_Type
   |
   |__ Gnoga.Gui.Module - Place holder for 3rd party extensions
   |
   |__ Gnoga.Gui.Navigator
   |
   |__ Gnoga.Gui.Plugin - Place hordler for 3rd part JS bindings
   |           included - Ace_Editor
   |                      Boot_Strap
   |                      |
   |                      |__ Container_Type
   |                          Fluid_Container_Type
   |                          Row_Type
   |                          Jumbotron_Type
   |                          Table_Type
   |                          Form_Group_Type
   |                          Check_Box_Type
   |                          Radio_Button_Type
   |                      jQuery - Additional to Gnoga's use
   |                      jQueryUI - Interactions, Effects, Utilities
   |                      |
   |                      |__ jQueryUI.Widget.Accordian_Type
   |                                         jQueryUI Button
   |                                         Dialog_Type
   |                                         Progress_Bar_Type
   |                                         jQueryUI Menu
   |                                         jQueryUI Select Menu
   |                                         Tabs_Type
   |                                         jQueryUI Tool Tips
   |                      MacGap - Native Mac OS X functionality
   |__ Gnoga.Gui.Screen
   |
   |__ Gnoga.Gui.View.View_Base_Type
   |              |    View_Type
   |              |
   |              |__ View.Card.Card_View_Type
   |              |             Tab_Type
   |              |             Tab_Item_Type
   |              |
   |              |__ View.Console.Console_View_Type
   |              |
   |              |__ View.Docker.Docker_View_Type
   |
   |__ Gnoga.Gui.Window.Window_Type
```

2.4 Property and Method Overview

2.4.1 Base_Type

```
   * Properties
```

```
      - Height
      - Width

   * Frawmework Properties:
      - Buffer_Connection
      - Connection_Data [ro]
      - Connection_ID
      - DOM_Selector [ro]
      - Dynamic
      - ID
      - ID_Type [ro]
      - Parent
      - Unique_ID [ro]
      - Valid [ro]

   * Generic Client Side Access to Properties
      - Property

   * Methods
      - Focus
      - Blur

   * Framework Methods
      - Flush_Buffer

   * Generic Client Side Execution of Methods
      - Execute
```

2.4.2 Element_Type

All of Base_Type Plus:

```
   * Properties - General
      - Access_Key
      - Advisory_Title
      - Class_Name
      - Draggable
      - Editable
      - Inner_HTML
      - Outer_HTML [ro]

   * Properties - Text Content
      - Language_Code
      - Tab_Index
      - Spell_Check
      - Text
      - Text_Direction

   * Properties - Visibility and Layout
      - Hidden
      - Visible
      - Display
      - Clear_Side [wo]
      - Layout_Float [wo]
      - Overflow
      - Overflow_X [wo]
      - Overflow_Y [wo]
      - Resizable
      - Position
      - Verticle_Align [wo]
      - Box_Sizing
      - Z_Index [wo]
      - Margin [wo]
      - Padding [wo]
```

```
* Properties - Position
   - Height (from Base_Type)
   - Width (from Base_Type)
   - Position_Top (ro)
   - Position_Left (ro)
   - Offset_From_Top (ro)
   - Offset_From_Left (ro)
   - Left
   - Right
   - Top
   - Bottom
   - Box_Height
   - Box_Width
   - Minimum_Height
   - Minimum_Width
   - Maximum_Height
   - Maximum_Width
   - Inner_Height
   - Inner_Width
   - Outer_Height (ro)
   - Outer_Width (ro)
   - Outer_Height_To_Margin (ro)
   - Outer_Width_To_Margin (ro)
   - Client_Width (ro)
   - Client_Height (ro)
   - Client_Left (ro)
   - Client_Top (ro)
   - Offset_Width (ro)
   - Offset_Height (ro)
   - Offset_Left (ro)
   - Offset_Top (ro)
   - Scroll_Width (ro)
   - Scroll_Height (ro)
   - Scroll_Left
   - Scroll_Top

* Properties - Style - Color
   - Color
   - Opacity
   - Background_Attachment
   - Background_Color
   - Background_Image
   - Background_Position
   - Background_Origin
   - Background_Repeat
   - Background_Clip
   - Background_Size
   - Border (wo)
   - Border_Radius (wo)
   - Shadow (wo) / Shadow_None
   - Outline (wo)
   - Cursor

* Properties - Style - Text
   - Font (wo)
   - Text_Alignment (wo)

* Framework Properties:
   - Auto_Place
   - First_Child
   - Next_Sibling
   - HTML_Tag (ro)

* Generic Client Side Access to Properties
   - Style
   - Attribute

* Methods
   - Click
```

```
    - Add_Class
    - Remove_Class
    - Toggle_Class
    - Place_Inside_Top_Of
    - Place_Inside_Bottom_Of
    - Place_Before
    - Place_After
    - Remove
```

2.4.3 Events

```
* Object Events
   - On_Resize
   - On_Scroll

* Form Events
   - On_Focus
   - On_Blur
   - On_Change
   - On_Focus_In
   - On_Focus_Out
   - On_Input
   - On_Reset
   - On_Search
   - On_Select
   - On_Submit

* Mouse Events
   - On_Click
   - On_Mouse_Click
   - On_Mouse_Right_Click
   - On_Context_Menu
   - On_Double_Click
   - On_Mouse_Double_Click
   - On_Mouse_Enter
   - On_Mouse_Leave
   - On_Mouse_Over
   - On_Mouse_Out
   - On_Mouse_Down
   - On_Mouse_Up
   - On_Mouse_Move

* Drag and Drop Events
   - On_Drag_Start
   - On_Drag
   - On_Drag_End
   - On_Drag_Enter
   - On_Drag_Leave
   - On_Drop

* Keyboard Events
   - On_Character
   - On_Wide_Character
   - On_Key_Down
   - On_Key_Up
   - On_Key_Press

 * Clipboard Events
   - On_Copy
   - On_Cut
   - On_Paste

 * Generic Events
   - On_Create
   - On_Destroy
   - On_Child_Added
```

```
      - On_Child_Removed
      - On_Message
```

3 UXStrings

3.1 Types

in v22, simply use String type instead of UXStrings.

```
type Encoding_Scheme is (ASCII_7, Latin_1, UTF_8, UTF_16BE, UTF_16LE);
-- Supported encoding schemes

subtype UTF_16_Encoding_Scheme is Encoding_Scheme range UTF_16BE .. UTF_16LE;
-- Supported UTF-16 encoding schemes

subtype ASCII_Character is Ada.Characters.Handling.ISO_646;
subtype ASCII_Character_Array is String with Dynamic_Predicate => (for all Item of
ASCII_Character_Array => Item in ASCII_Character);
-- Characters in ISO/IEC 646

subtype Latin_1_Character is Character;
subtype Latin_1_Character_Array is String;
-- Characters in ISO/IEC 8859-1

subtype BMP_Character is Wide_Character;
subtype BMP_Character_Array is Wide_String;
-- Characters in Unicode Basic Multilingual Plane
-- (Could be also named UCS_2_Character (Universal Coded Character Set)?)

subtype Unicode_Character is Wide_Wide_Character;
subtype Unicode_Character_Array is Wide_Wide_String;
-- Characters in Unicode planes
-- (Could be also named UCS_4_Character?)

subtype UTF_8_Character_Array is Ada.Strings.UTF_Encoding.UTF_String;
subtype UTF_16_Character_Array is Ada.Strings.UTF_Encoding.UTF_String;
-- Array of 8 bits values representing UTF encodings (UTF-8, UTF-16BE, or UTF-16LE)

type UXString is tagged private with
    Constant_Indexing => Element,
    Iterable => (First => First, Next => Next, Has_Element => Has_Element, Element =>
Element), String_Literal    => From_Unicode;
-- Container type of Unicode characters with dynamic size usually named string

Null_UXString : constant UXString;
-- Represent the null string
```

3.2 Uxstrings

3.2.1 Append

- Description

  Update Source to the concatenation of Source and New_Item.

- Usage

```
procedure Append (Source : in out UXString;
                  New_Item : UXString)
```

```
procedure Append (Source : in out UXString;
                  New_Item : Unicode_Character)
```

- Example

```

```

3.2.2 Character_Set_Version

- Description

  Returns an implementation-defined identifier that identifies the version of the character set standard that is used for categorizing characters by the implementation.

- Usage

```
function Character_Set_Version return UXString
```

- Example

```

```

3.2.3 Count

- Description

  Return the number of non overlapping occurrences of Pattern matching Source with respect of Mapping.

  Return the number of occurrences of characters in parameter Set matching Source

- Usage

```
function Count (Source : UXString;
                Pattern: UXString;
                Mapping : Wide_Wide_Character_Mapping := Identity)
return Natural

function Count (Source : UXString;
                Pattern : UXString;
                Mapping : Wide_Wide_Character_Mapping_Function)
return Natural

function Count (Source : UXString;
                Set : Wide_Wide_Character_Set) return Natural
```

- Example

```

```

3.2.4 Delete

- Description

  Return Source whom characters with positions from Low to High are removed.

  Update Source whom characters with positions from Low to High are removed

- Usage

```
function Delete (Source : UXString;
                 From : Positive; Through : Natural) return UXString

procedure Delete (Source : in out UXString;
                  From : Positive; Through : Natural)
```

- Example

```

```

3.2.5 Element

- Description

  Return the Unicode character of Source at Index position

- Usage

  function Element (Source : UXString; Index : Positive) return Unicode_Character;

- Example

```

```

3.2.6 Ending

- Description

  Get or set end of line mode. You should use Line_Mark to apply the ending mode.

- Usage

```
type Line_Ending is (CR_Ending, LF_Ending, CRLF_Ending);

function Ending (File : in File_Type) return Line_Ending;
procedure Ending (File : in File_Access; Value : Line_Ending);
```

- Example

```

```

3.2.7 Ends_With

- Description

  Return True if Source ends with pattern with respect case of sensitivity.

- Usage

```
function Ends_With (Source : UXString; Pattern : UXString; Sensitivity :
Case_Sensitivity := Sensitive) return Boolean;
```

- Example

```
```

3.2.8 Equal_Case_Insentive

- Description

  Returns True if the strings consist of the same sequence of characters after applying locale-independent simple case folding, as defined by documents referenced in the note in Clause 1 of ISO/IEC 10646:2011. Otherwise, returns False.

- Usage

```
function Equal_Case_Insensitive (Left, Right : UXString) return Boolean
```

- Example

```
```

3.2.9 Find_Token

- Description

  Set First to position of the first character inside or outside Set matches Source starting at From position. Set Last to position of the last character inside or outside Set matches Source with respect of Test membership

  Set First to position of the first character inside or outside Set matches Source. Set Last to position of the last character inside or outside Set matches Source with respect of Test membership.

- Usage

```
procedure Find_Token (Source : UXString;
                      Set : Wide_Wide_Character_Set;
                      From : Positive; Test : Membership;
                      First : out Positive; Last   : out Natural)

procedure Find_Token (Source : UXString;
                      Set : Wide_Wide_Character_Set;
                      Test : Membership; First : out Positive;
                      Last : out Natural)
```

- Example

```
```

3.2.10 First

- Description

  Return the position of the first character of Source (actually 1).

- Usage

```
function First (Source : UXString) return Positive;
```

- Example

```
```

3.2.11 From_ASCII

- Description

  Return an UXString from the ASCII character Item.
  Return an UXString from the array of ASCII characters Source.

- Usage

```
function From_ASCII (Item : ASCII_Character) return UXString

function From_ASCII (Source : ASCII_Character_Array) return UXString
```

- Example

```
```

3.2.12 From_BMP

- Description

  Return an UXString from the BMP character parameter Item.
  Return an UXString from the array of BMP characters parameter Source.

- Usage

```
function From_BMP (Item : BMP_Character) return UXString

function From_BMP (Source : BMP_Character_Array) return UXString
```

- Example

```
```

3.2.13 From_Latin_1

- Description

  Return an UXString from the Latin 1 character parameter Item.
  Return an UXString from the array of Latin 1 characters parameter Source.

- Usage

```
function From_Latin_1 (Item : Latin_1_Character) return UXString

function From_Latin_1 (Source : Latin_1_Character_Array) return UXString
```

- Example

```

```

3.2.14 From_Unicode

- Description

  Return an UXString from the Unicode character parameter Item.
  Return an UXString from the array of Unicode characters parameter Source.

- Usage

```
function From_Unicode (Item : Unicode_Character) return UXString

function From_Unicode (Source : Unicode_Character_Array) return UXString
```

- Example

```

```

3.2.15 From_UTF_8

- Description

  Return an UXString from the array of UTF-8 characters parameter Source,
  leading BOM characters are suppressed if any.

- Usage

```
function From_UTF_8 (Source : UTF_8_Character_Array) return UXString
```

- Example

```

```

3.2.16 From_UTF_16

- Description

  Return an UXString from the array of UTF-16 characters parameter Source according to the encoding scheme specified by Input_Scheme, leading BOM characters are suppressed if any.

- Usage

```
function From_UTF_16 (Source : UTF_16_Character_Array;
                      Input_Scheme : UTF_16_Encoding_Scheme)
return UXString
```

- Example

```
```

3.2.17 Get_ASCII

- Description

  Return the ASCII character of Source at Index position, if the character is not in ASCII set then Substitute is returned

- Usage

```
Q_L : Latin_1_Character renames Ada.Characters.Latin_1.Question

function Get_ASCII (Source : UXString;
                    Index : Positive;
                    Substitute : in ASCII_Character := Q_L)
return ASCII_Character
```

- Example

```
```

3.2.18 Get_BMP

- Description

  Return the BMP character from Source at Index position, if the character is not in BMP set then Substitute is returned.

- Usage

```
Inv_Q_B : BMP_Character renames Ada.Characters.Wide_Latin_1.Inverted-
_Question

function Get_BMP (Source : UXString;
                  Index : Positive;
```

```
                    Substitute : in BMP_Character := Inv_Q_B]
return BMP_Character
```

- Example

```

```

3.2.19 Get_Latin_1

- Description

  Return the Latin 1 character from Source at Index position, if the character is not in latin 1 set then Substitute is returned.

- Usage

```
Inv_Q_L : Latin_1_Character renames Ada.Characters.Latin_1.Inverted_-
Question

function Get_Latin_1 (Source : UXString;
                      Index : Positive;
                      Substitute : in Latin_1_Character := Inv_Q_L]
return Latin_1_Character
```

- Example

```

```

3.2.20 Get_Unicode

- Description

  Return the Unicode character from Source at Index position.

- Usage

```
function Get_Unicode (Source : UXString;
                      Index : Positive] return Unicode_Character
```

- Example

```

```

3.2.21 Has_Element

- Description

  Return True if a character of Source is present at Index position (actually Index <= Length (Source)).

- Usage

```
function Has_Element (Source : UXString;
```

```
                        Index : Positive] return Boolean
```

- Example

```
```

3.2.22 Head

- Description

  Return the first characters from Source up to Count concatenated with Pad characters if needed.

  Update Source to the first characters from Source up to Count concatenated with Pad characters if needed.

- Usage

```
function Head [Source : UXString;
               Count : Natural;
               Pad : Unicode_Character := Wide_Wide_Space]
return UXString

procedure Head [Source : in out UXString;
                Count : Natural;
                Pad : Unicode_Character := Wide_Wide_Space]
```

- Example

```
```

3.2.23 Index

- Description

  [1] Return the position of the first character where Pattern matches Source with respect of Going direction and Mapping

  [2] Return the position of the first character where Pattern matches Source with respect of Going direction and Mapping.

  [3] Return the position of the first character inside or outside Set matches Source with respect of Going direction and Test membership.

  [4] Return the position of the first character where Pattern matches Source starting at From position with respect of Going direction and Mapping.

  [5] Return the position of the first character where Pattern matches Source starting at From position with respect of Going direction and Mapping.

  [6] Return the position of the first character inside or outside Set matches Source starting at From position with respect of Test membership.

- Usage

```
[1] function Index [Source : UXString; Pattern : UXString;
                    Going : Direction := Forward;
                    Mapping : Wide_Wide_Character_Mapping := Identity]
return Natural

[2] function Index [Source : UXString;
                    Pattern : UXString;
                    Going : Direction := Forward;
                    Mapping : Wide_Wide_Character_Mapping_Function]
return Natural

[3] function Index [Source : UXString;
                    Set : Wide_Wide_Character_Set;
                    Test : Membership := Inside;
                    Going : Direction := Forward]
return Natural

[4] function Index [Source  : UXString;
                    Pattern : UXString; From : Positive;
                    Going : Direction := Forward;
                    Mapping : Wide_Wide_Character_Mapping := Identity]
return Natural

[5] function Index [Source  : UXString;
                    Pattern : UXString;
                    From : Positive;
                    Going : Direction := Forward;
                    Mapping : Wide_Wide_Character_Mapping_Function]
return Natural

[6] function Index [Source : UXString;
                    Set : Wide_Wide_Character_Set;
                    From : Positive
                    Test : Membership := Inside;
                    Going  : Direction := Forward]
return Natural;

With Ada.Strings - Direction is [Forward, Backward]
   type Alignment  is [Left, Right, Center]
   type Truncation is [Left, Right, Error]
   type Membership is [Inside, Outside]
   type Trim_End   is [Left, Right, Both]
```

- Example

```
```

3.2.24 Index_Non_Blank

- Description

Return the position of the first non space character of Source with respect of Going direction.

Return the position of the first non space character of Source starting at From position with respect of Going direction.

- Usage

```
function Index_Non_Blank (Source : UXString;
                          Going : Direction := Forward) return Natural

function Index_Non_Blank (Source : UXString;
                          From : Positive;
                          Going : Direction := Forward) return Natural
```

- Example

```
```

3.2.25 Insert

- Description

Return Source with New_Item inserted at position ahead of parameter Before.
Update Source with New_Item inserted at position ahead of parameter Before.

- Usage

```
function Insert (Source : UXString;
                 Before : Positive;
                 New_Item : UXString) return UXString

procedure Insert (Source : in out UXString;
                  Before : Positive; New_Item : UXString)
```

- Example

```
```

3.2.26 Is_ASCII, Is_ISO_646

- Description

Return True if all the characters of Source are in ASCII set.
Return True if the character of Source at Index position is in ASCII set.

- Usage

```
function Is_ASCII (Source : UXString) return Boolean

function Is_ASCII (Source : UXString; Index : Positive) return Boolean

function Is_ISO_646 (Item : UXString) return Boolean renames Is_ASCII
```

- Example

```
```

3.2.27 Is_Basic

- Description

  Return True if source is basic (with no diacritical mark).

- Usage

```
function Is_Basic (Source : UXString) return Boolean
```

- Example

```
```

3.2.28 Is_BMP

- Description

  Return True if all the characters of Source are in BMP set.
  Return True if the character of Source at Index position is in BMP set.

- Usage

```
function Is_BMP (Source : UXString) return Boolean

function Is_BMP (Source : UXString; Index : Positive) return Boolean
```

- Example

```
```

3.2.29 Is_Empty

- Description

  Return True is Source is empty (equal to Null_UXString).

- Usage

```
function Is_Empty (Source : UXString) return Boolean
```

- Example

```
```

3.2.30 Is_Latin_1

- Description

  Return True if all the characters of Source are in Latin 1 set.
  Return True if the character of Source at Index position is in Latin 1 set.

- Usage

  Inv_Q_L : Latin_1_Character renames Ada.Characters.Latin_1.Inverted_Question;

```
function Is_Latin_1 (Source : UXString) return Boolean

function Is_Latin_1 (Source : UXString; Index : Positive) return Boolean
```

- Example

```

```

3.2.31 Is_Lower

- Description

  Return True if Source is lowercase.

- Usage

```
function Is_Lower (Source : UXString) return Boolean
```

- Example

```

```

3.2.32 Is_Unicode

- Description

  Return True if all the characters of Source are in Unicode set (actually True).
  Return True if the character of Source at Index position is in Unicode set (actually True).

- Usage

```
function Is_Unicode (Source : UXString) return Boolean

function Is_Unicode (Source : UXString; Index : Positive) return Boolean
```

- Example

```

```

3.2.33 Is_Upper

- Description

  Return True if Source is lowercase.

- Usage

```
function Is_Upper (Source : UXString) return Boolean;
```

- Example

```

```

3.2.34 Last

- Description

  Return the position of the last character of Source (actually Length (Source)).

- Usage

```
function Last (Source : UXString) return Natural;
```

- Example

```

```

3.2.35 Length

- Description

  Return the number of (Unicode) characters.

- Usage

```
function Length (Source : UXString) return Natural;
```

- Example

```

```

3.2.36 Less_Case_Insensitive

- Description

  Performs a lexicographic comparison of strings Left and Right, converted to lower case.

- Usage

```
function Less_Case_Insensitive (Left, Right : UXString) return Boolean;
```

- Example

```
```

3.2.37 Line_Mark

- Description

  Apply end of line mode.

- Usage

```
function Line_Mark return UXString;
procedure Line_Mark (Ending : Line_Ending);
```

- Example

```
```

3.2.38 Next

- Description

  Return the position of the next character of Source after Index position (actually Index + 1).

- Usage

```
function Next (Source : UXString; Index : Positive) return Positive

procedure Next (Source : UXString; Index : in out Positive)
```

- Example

```
```

3.2.39 Overwrite

- Description

  Return Source whom characters starting at Position are replaced with parameter New_Item
  Update Source whom characters starting at Position are replaced with parameter New_Item

- Usage

```
function Overwrite (Source : UXString;
                    Position : Positive;
                    New_Item : UXString) return UXString
procedure Overwrite (Source : in out UXString;
                     Position : Positive;
```

```
                        New_Item : UXString]
```

- Example

```
```

3.2.40 Prepend

- Description

  Update Source to the concatenation of New_Item and Source

- Usage

```
procedure Prepend [Source : in out UXString;
                   New_Item : UXString]

procedure Prepend [Source : in out UXString;
                   New_Item : Unicode_Character]
```

- Example

```
```

3.2.41 Remove

- Description

  [1] Return Source where every occurrence of Pattern have been removed with respect of case sensitivity.

  [2] Update Source where every occurrence of Pattern have been removed with respect of case sensitivity.

  [3] Return Source where every occurrence of Pattern have been removed with respect of case sensitivity.

  [4] Update Source where every occurrence of Pattern have been removed with respect of case sensitivity.

- Usage

```
[1] function Remove [Source : UXString; Pattern : Unicode_Character; Sen-
sitivity : Case_Sensitivity := Sensitive] return UXString

[2] procedure Remove [Source : in out UXString; Pattern : Unicode_Charac-
ter; Sensitivity : Case_Sensitivity := Sensitive]

[3] function Remove [Source : UXString; Pattern : UXString; Sensitivity :
Case_Sensitivity := Sensitive] return UXString

[4] procedure Remove [Source : in out UXString; Pattern : UXString; Sen-
sitivity : Case_Sensitivity := Sensitive]
```

- Example

```
```

3.2.42 Replace

- Description

  [1] Return a string which has had the before character replaced with the after character wherever the before character is found with respect of sensitivity.

  [2] Update Source which has had the before character replaced with the after character wherever the before character is found with respect of sensitivity.

  [3] Return a string which has had the before text replaced with the after text wherever the before text is found with respect of sensitivity.

  [4] Update Source which has had the before text replaced with the after text wherever the before text is found with respect of case sensitivity

- Usage

```
[1] function Replace [Source : UXString; Before, After : Unicode_Character; Sensitivity : Case_Sensitivity := Sensitive] return UXString

[2] procedure Replace [Source : in out UXString; Before, After : Unicode_Character; Sensitivity : Case_Sensitivity := Sensitive]

[3] function Replace [Source : UXString; Before, After : UXString; Sensitivity : Case_Sensitivity := Sensitive] return UXString

[4] procedure Replace [Source : in out UXString; Before, After : UXString; Sensitivity : Case_Sensitivity := Sensitive]
```

- Example

```
```

3.2.43 Replace_ASCII

- Description

  Update Source such as the character at Index position is set to the ASCII character parameter By.

- Usage

```
procedure Replace_ASCII [Source : in out UXString;
                         Index : Positive;
                         By : ASCII_Character]
```

- Example

```
```

3.2.44 Replace_Latin_1

- Description

  Update Source such as the character at Index position is set to the Latin 1 character parameter By.

- Usage

```
procedure Replace_Latin_1 (Source : in out UXString;
                           Index : Positive;
                           By : Latin_1_Character)
```

- Example

```
```

3.2.45 Replace_BMP

- Description

  Update Source such as the character at Index position is set to the BMP character parameter By.

- Usage

```
procedure Replace_BMP (Source : in out UXString;
                       Index : Positive;
                       By : BMP_Character)
```

- Example

```
```

3.2.46 Replace_Slice

- Description

  Return Source whom characters with positions from Low to High are replaced with parameter By.

  Update Source whom characters with positions from Low to High are replaced with parameter By.

- Usage

```
function Replace_Slice (Source : UXString;
```

```
                          Low : Positive;
                          High : Natural;
                          By : UXString] return UXString

procedure Replace_Slice [Source : in out UXString;
                         Low : Positive;
                         High : Natural;
                         By : UXString]
```

- Example

```
```

3.2.47 Replace_Unicode

- Description

  Update Source such as the character at Index position is set to the Unicode character parameter By.

- Usage

```
procedure Replace_Unicode [Source : in out UXString;
                           Index : Positive
                           By : Unicode_Character];
```

- Example

```
```

3.2.48 Set

- Description

  Set Target to an UXString from the array of Unicode characters parameter Source.

- Usage

```
procedure Set [Target : out UXString; Source : Unicode_Character_Array]
```

- Example

```
```

3.2.49 Scheme

- Description

  Get or set encoding scheme for file operation.

- Usage

```
type Encoding_Scheme is (ASCII_7, Latin_1, UTF_8, UTF_16BE, UTF_16LE)

function Scheme (File : in File_Type) return Encoding_Scheme
procedure Scheme (File : in File_Access; Value : in Encoding_Scheme)
```

- Example

```

```

3.2.50 Slice

- Description

  Return the slice at positions Low through High from Source.

  Set Target to the slice at positions Low through High from Source.

- Usage

```
function Slice (Source : UXString;
                Low : Positive;
                High : Natural) return UXString

procedure Slice (Source : UXString;
                 Target : out UXString;
                 Low : Positive;
                 High : Natural)
```

- Example

```

```

3.2.51 Split

- Description

  (1) Return a string list resulting in spliting Source into substrings wherever Separator occurs.

  (2) Return a string list resulting in spliting Source into substrings wherever Separator occurs.

  (3) Return a string list resulting in spliting Source into substrings wherever Separator occurs with respect of Test membership.

- Usage

```
(1) function Split (Source : UXString; Separator : Unicode_Character;
Sensitivity : Case_Sensitivity := Sensitive) return UXStrings.Lists.UXString_List
```

```
[2] function Split [Source : UXString; Separator : UXString;
Sensitivity : Case_Sensitivity := Sensitive] return UXStrings.List-
s.UXString_List

[3] function Split [Source : UXString; Separator :
Wide_Wide_Character_Set; Test : Membership := Inside] return
UXStrings.Lists.UXString_List
```

- Example

```
```

3.2.52 Tail

- Description

  Return the last characters from Source up to Count concatenated with Pad characters if needed

  Update Source to the last characters from Source up to Count concatenated with Pad characters if needed

- Usage

```
function Tail [Source : UXString;
               Count : Natural;
               Pad : Unicode_Character := Wide_Wide_Space]
return UXString

procedure Tail [Source : in out UXString;
                Count : Natural;
                Pad : Unicode_Character := Wide_Wide_Space]
```

- Example

```
```

3.2.53 To_ASCII, To_ISO_646

- Description

  Return an array of ASCII characters from Source, if a character is not in ASCII set then Substitute is returned.

- Usage

```
Q_L : Latin_1_Character renames Ada.Characters.Latin_1.Question

function To_ASCII [Source : UXString;
                   Substitute : in ASCII_Character := Q_L]
return ASCII_Character_Array

function To_ISO_646   [Item : UXString;
                       Substitute : Ada.Characters.Handling.ISO_646:=Q_L]
```

```
return ASCII_Character_Array renames To_ASCII
```

- Example

```

```

3.2.54 To_Basic

- Description

  Returns the letter corresponding to Item but with no diacritical mark, if Item is a letter but not a basic letter; returns Item otherwise.

- Usage

```
function To_Basic (Item : UXString) return UXString
```

- Example

```

```

3.2.55 To_BMP

- Description

  Return an array of BMP characters from Source, if a character is not in BMP set then Substitute is returned.

- Usage

```
Inv_Q_B : BMP_Character renames Ada.Characters.Wide_Latin_1.Inverted_-
Question

function To_BMP (Source : UXString;
                 Substitute : in BMP_Character := Inv_Q_B)
return BMP_Character_Array
```

```

```

3.2.56 To_Latin_1

- Description

  Return an array of Latin 1 characters from Source, if a character is not in latin 1 set then Substitute is returned.

- Usage

```
Inv_Q_L : Latin_1_Character renames Ada.Characters.Latin_1.Inverted_-
Question

function To_Latin_1 (Source : UXString;
                     Substitute : in Latin_1_Character := Inv_Q_L)
```

```
return Latin_1_Character_Array
```

- Example

```

```

3.2.57 To_Lower

- Description

  Returns the corresponding lower-case value for Item if Is_Upper(Item), and returns Item otherwise.

- Usage

```
function To_Lower (Item : UXString) return UXString
```

- Example

```

```

3.2.58 To_Unicode

- Description

  Return an array of Unicode characters from Source.

- Usage

```
function To_Unicode (Source : UXString) return Unicode_Character_Array
```

- Example

```

```

3.2.59 To_Upper

- Description

  Returns the corresponding upper-case value for Item if Is_Lower(Item) and Item has an upper-case form, and returns Item otherwise.

- Usage

```
function To_Upper (Item : UXString) return UXString
```

- Example

```

```

3.2.60 To_UTF_8

- Description

  Return an array of UTF-8 characters from Source, prepend UTF-8 BOM if Output_BOM is set to True.

- Usage

```
function To_UTF_8 (Source : UXString;
                   Output_BOM : Boolean := False)
return UTF_8_Character_Array
```

3.2.61 To_UTF_16

- Description

  Return an array of UTF-16 characters from Source according to the encoding scheme specified by Output_Scheme, prepend UTF-16 BOM if Output_BOM is set to True.

- Usage

```
function To_UTF_16 (Source : UXString;
                    Output_Scheme : UTF_16_Encoding_Scheme;
                    Output_BOM : Boolean := False)
return UTF_16_Character_Array
```

- Example

3.2.62 Translate

- Description

  Return Source updated with respect of Mapping.

  Update Source with respect of Mapping.

  Return Source updated with respect of Mapping

  Update Source with respect of Mapping

- Usage

```
function Translate (Source : UXString;
                    Mapping : Wide_Wide_Character_Mapping)
return UXString

procedure Translate (Source : in out UXString;
                     Mapping : Wide_Wide_Character_Mapping)

function Translate (Source : UXString;
                    Mapping: Wide_Wide_Character_Mapping_Function)
```

```
return UXString

procedure Translate [Source : in out UXString;
                     Mapping : Wide_Wide_Character_Mapping_Function]
```

```

```

3.2.63 Trim

- Description

  Return Source with Space characters removed from left, right or both with respect of Side.

  Update Source with Space characters removed from left, right or both with respect of Side.

  Return Source with leading characters in Left and trailing characters in Right removed

  Update Source with leading characters in Left and trailing characters in Right removed

- Usage

```
function Trim [Source : UXString; Side : Trim_End] return UXString

procedure Trim [Source : in out UXString; Side : Trim_End]

function Trim [Source : UXString;
               Left : Wide_Wide_Character_Set;
               Right : Wide_Wide_Character_Set] return UXString

procedure Trim [Source : in out UXString;
                Left : Wide_Wide_Character_Set;
                Right : Wide_Wide_Character_Set]

With Ada.Strings - type Trim_End is [Left, Right, Both]
```

- Example

```

```

3.2.64 &

- Description

  Return the concatenation of Left and Right.

- Usage

```
function "&" [Left : UXString; Right : UXString] return UXString

function "&" [Left : UXString;
              Right : Unicode_Character] return UXString
```

```
function "&" (Left : Unicode_Character;
              Right : UXString) return UXString
```

- Example

```

```

3.2.65 *

- Description

  Return Right string duplicated Left times

  Return Right character duplicated Left times

- Usage

```
function "*" (Left : Natural; Right : UXString) return UXString

function "*" (Left : Natural; Right : Unicode_Character) return UXString
```

- Example

```

```

3.2.66 =

- Description

  Return True if Left equals Right.

- Usage

```
function "=" (Left : UXString; Right : UXString) return Boolean
```

- Example

```

```

3.2.67 <

- Description

  Return True if Left is less than Right.

- Usage

```
function "<" (Left : UXString; Right : UXString) return Boolean
```

- Example

```
```

3.2.68 <=

- Description

  Return True if Left is less or equal than Right.

- Usage

```
function "<=" (Left : UXString; Right : UXString) return Boolean
```

- Example

```
```

3.2.69 >

- Description

  Return True if Left is greater than Right.

- Usage

```
function ">" (Left : UXString; Right : UXString) return Boolean
```

- Example

```
```

3.2.70 >=

- Description

  Return True if Left greater or equal than Right.

- Usage

```
function ">=" (Left : UXString; Right : UXString) return Boolean
```

- Example

```
```

3.3 Uxstrings.Text_IO - Text console management

3.3.1 New_Line

- Description

  Create a new blank line, or more than one when Spacing is passed.

- Usage

```
procedure New_Line [Spacing : in Positive_Count := 1]
```

- Example

```
```

3.3.2 Get_Immediate

- Description

  Get a character validated by [Enter]

- Usage

```
function Get_Line return UXString
procedure Put_Line [Item : in UXString]
```

- Example

```
```

3.3.3 Get_Line

- Description

  Get a string validated by [Enter]

- Usage

```
procedure Get_Line [Item : out UXString]
function Get_Line return UXString
```

- Example

```
```

3.3.4 Put

- Description

  Print to the console.

- Usage

```
procedure Put [Item : in UXString]
procedure Get [Item : out UXString; Length : in Count]
```

- Example

```

```

3.3.5 Put_Line

- Description

  Print to the console then add a new line.

- Usage

```
 procedure Put_Line [Item : in UXString]
```

- Example

```

```

3.4 Uxstrings.Text_IO - Text file management

Settings for v22

```
-- with UXStrings;         use UXStrings;

with UXStrings.Text_IO;
with UXStrings.Text_IO.Text_Streams;
with UXStrings.Conversions;

procedure Example is

   function Image is new UXStrings.Conversions.Scalar_Image [Encoding_Scheme];
   function Value is new UXStrings.Conversions.Scalar_Value [Encoding_Scheme];

   package UXS  renames UXStrings;
   package UTI  renames UXStrings.Text_IO;
   package UTS  renames UXStrings.Text_IO.Text_Streams;
```

3.4.1 Close

- Description

  Close a file.

- Usage

```
procedure Close (File : in out File_Type)
```

- Example

```

```

3.4.2 Create

- Description

Create a file.
Default file mode is “Out” (write mode).

- Usage

```
type File_Mode is (In_File, Out_File, Append_File)
type Encoding_Scheme is (ASCII_7, Latin_1, UTF_8, UTF_16BE, UTF_16LE)
type Line_Ending is (CR_Ending, LF_Ending, CRLF_Ending)

procedure Create (File     : in out File_Type; Mode : in File_Mode := Out_File;    Name    :    in    UXString    :=    Null_UXString; Scheme    :    in Encoding_Scheme := Latin_1; Ending : Line_Ending := CRLF_Ending)
```

- Example

```

```

3.4.3 End_Of_File

- Description

Return true if end of file is reached.

- Usage

```
function End_Of_File (File : in out File_Type) return Boolean
```

- Example

```

```

3.4.4 End_Of_Line

- Description

Return true if end of line is reached.

- Usage

```
function End_Of_Line (File : in out File_Type) return Boolean
```

- Example

```
```

3.4.5 End_Of_Page

- Description

  Return true if end of page is reached.

- Usage

```
function End_Of_Page (File : in File_Type) return Boolean;
```

- Example

```
```

3.4.6 Flush

- Description

  Flush file buffer to disk.

- Usage

```
procedure Flush (File : in File_Type)
```

- Example

```
```

3.4.7 Get_Line

- Description

  Get the current line and then move the file pointer to the next line.

- Usage

```
procedure Get_Line (File : in out File_Type; Item : out UXString)
function Get_Line (File : in out File_Type) return UXString
```

- Example

```
```

3.4.8 Is_Open

- Description

  Returns true if Handle file is open.

- Usage

```
function Is_Open (Handle : in File) return Boolean
```

- Example

```

```

3.4.9 Line

- Description

  Create a new blank line, or more when Spacing is passed.

- Usage

```
procedure  New_Line (Handle : File; Spacing : Positive)
```

- Example

```

```

3.4.10 Open

- Description

  Open a file.

- Usage

```
type File_Mode is (In_File, Out_File, Append_File)
type Encoding_Scheme is (ASCII_7, Latin_1, UTF_8, UTF_16BE, UTF_16LE)
type Line_Ending is (CR_Ending, LF_Ending, CRLF_Ending)

procedure Open (File : in out File_Type;
                Mode : in File_Mode;
                Name : in UXString;
                Scheme : in Encoding_Scheme := Latin_1;
                Ending : Line_Ending := CRLF_Ending)
```

- Example

```

```

3.4.11 Put

- Description

  Write to a file.

- Usage

```
procedure Put (File : in File_Type; Item : in UXString)
```

- Example

```
```

3.4.12 Put_Line

- Description

  Write a file and then add a new line.

- Usage

```
procedure Put_Line (File : in File_Type; Item : in UXString)
```

- Example

```
```

3.4.13 Reset

- Description

  Reset the file pointer to the start of the file.

- Usage

```
type File_Mode is (In_File, Out_File, Append_File)

procedure Reset (File : in out File_Type; Mode : in File_Mode)
procedure Reset (File : in out File_Type)
```

- Example

```
```

https://this-page-intentionally-left-blank.org

FAQ

With the Wildebeest and the Penguin, there's no Bull.
Number Six

1 v22

1.1 Constants

1.1.1 ANSI colors for console

This constants conforms to ISO 6429 standard :

```
CONSOLE_COLOR_GREEN  : constant String := ESC & "[1;32m";
CONSOLE_COLOR_RED    : constant String := ESC & "[1;31m";
CONSOLE_COLOR_YELLOW : constant String := ESC & "[1;33m";
CONSOLE_COLOR_RESET  : constant String := ESC & "[0m";
```

1.1.2 Control characters

Common control characters :

```
HT   : constant String := Character'Val[9]  & ""; -- 09d 09h Tab
LF   : constant String := Character'Val[10] & ""; -- 10d 0Ah Line Feed
CR   : constant String := Character'Val[13] & ""; -- 13d 0Dh Carriage return
ESC  : constant String := Character'Val[27] & ""; -- 27d 1Bh Escape
DQ   : constant String := Character'Val[34] & ""; -- 34d 22h Double quote
CRLF : constant String := CR & LF;
```

1.1.3 Delimiter characters

V22 conventional delimiter characters :

```
ND : constant String := "~"; -- 126d 7Eh Name/value delimiter
CD : constant String := "^"; --  94d 5Eh Column delimiter
RD : constant String := "\"; --  92d 5Ch Row delimiter
VD : constant String := ","; --  44d 2Ch Virgule (comma) delimiter
SD : constant String := ":"; --  58d 3Ah String delimiter
```

```
SP : constant String := " ";  --  32d 20h Space
```

Some of these delimiters are heavily used in the v20.Vst.Field_* functions.

1.1.4 Flag files

Useful names for testing mounts or install completed :

```
ACCESS_OK : constant String := "access_ok_dont_delete_this_file";
INSTALL_OK : constant String := "install_ok_dont_delete_this_file";
```

1.1.5 Redirection

Output redirections for standard and error outputs.

```
STD_OUT_REDIRECT  : constant String := " 1>/dev/null";
ERR_OUT_REDIRECT  : constant String := " 2>/dev/null";
STD_ERR_OUT_REDIRECT : constant String := " 2>/dev/null 1>/dev/null";
```

1.2 Conventional exit codes

- 1 -h or --help switches
- 2 invalid switch
- 3 invalid parameter
- 4 SQL error
- 5 reserved for future use
- 6 reserved for future use
- 7 reserved for future use
- 8 reserved for future use
- 9 if an exception occurs during execution

Exit codes greater than 9 are reserved to applications using v22. Typically, an application may use a base exit code by class command with local increment. Example: exit code for command "service backup" (backup being the first command of class service) could be Base_Exit_Code_Service + 1 :

```
   Base_Exit_Code_App :       constant Positive := 10;
   Base_Exit_Code_Cluster :   constant Positive := 20;
   Base_Exit_Code_Db :        constant Positive := 30;
   Base_Exit_Code_Domain:     constant Positive := 40;
   Base_Exit_Code_Group :     constant Positive := 50;
   Base_Exit_Code_Help :      constant Positive := 60;
   Base_Exit_Code_Instance :  constant Positive := 70;
   Base_Exit_Code_Info :      constant Positive := 80;
   Base_Exit_Code_Ip :        constant Positive := 90;
   Base_Exit_Code_Node :      constant Positive := 100;
   Base_Exit_Code_Owner :     constant Positive := 110;
   Base_Exit_Code_Remote :    constant Positive := 120;
   Base_Exit_Code_Service :   constant Positive := 130;
   Base_Exit_Code_User :      constant Positive := 140;
```

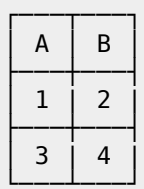
1.3 Gui

1.3.1 Content Group elements usage

Most element come with an On_Change callback, fired once the user stops focusing the given element.

Every item (except button and warning), have an explaining text next to it in the container.

1.3.2 Content List layout example




```
Content_List_Create (*, key);
Content_List_Add_Column (*, "A", key);
Content_List_Add_Column (*, "B", key);
R1 := Content_List_Add_Item (*, key, 1);
Content_List_Add_Text (*, "1", R1, key);
Content_List_Add_Text (*, "2", R1, key);
R2 := Content_List_Add_Item (*, key, 2);
Content_List_Add_Text (*, "3", R2, key);
Content_List_Add_Text (*, "4", R2, key);
```

1.3.3 Persistence within a connection

This essential function is provide by using v22.Gui API Set_Connection_Data, Set_Connection_Data and Clear_Connection_Data.

Connection_Data is a free to use dictionary unique to each user.

Example :

```
procedure Example_Set (Object : in out GGB.Base_Type'Class) is
begin
   Gui.Set_Connection_Data (Object, “Parameter”, “Value”);
end Example_Set;

procedure Example_Get (Object : in out GGB.Base_Type'Class) is
   Dummy : String;
begin
   Dummy := Gui.Get_Connection_Data (Object, “Parameter”);
end Example_Get;
```

1.4 Sql

1.4.1 General notes

This SQL package is a high-level API to simplify development. It includes, among others, the following features:

- Redundancy-free integration with low-level binding Gnoga.Server.Database;
- Standardized interface for all target databases;
- Database version management with table, index, relationship and field creation, followed by automatic update;
- Integrated data dictionary in Sys_Schema table;
- Set and Get configuration parameters in Sys_Config dedicated table;
- General purpose SQL API to ease the programmer.

See testapi demo program for further details.

Sql is (with the exception of initialization and finalization routines) a task aware package, using a mutex schema to avoid collision, for writing routines:
- Delete
- Insert
- Set_Config
- Update

1.4.2 MySQL notes

UTF-8 and MySQL is a long story. To keep it short, just check in /etc/mysql/mariadb.conf.d the following lines:

```
[mysqld]
...
skip-character-set-client-handshake
character-set-server = utf8mb4
collation-server = utf8mb4_general_ci
```

The first ensures that there will be no character set negotiation from the client to the server, while the other two guarantee optimum UTF-8 compatibility. Yes, you must specify utf8mb4 instead of utf8. As you may have guessed (no kidding), mb4 stands for *most bytes 4*, to be compatible with four-byte Unicode.

1.4.3 SQLite notes

A comprehensive "SQLite digest manual" is available to ease SQLite newcomers. See Sowebio Github repository.

To ease customization, SQLite DB is directly compiled and linked against sqlite3.c source located in src/sql.

1.4.4 Password encoding

- Description

  v22 passwords are SHA512 encoded, with the help of the GNAT.SHA512 package, as described in FIPS PUB 180-3. The complete text of FIPS PUB 180-3 can be found at: http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf The most recent standard is available here https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

- Usage

```
From_Latin_1 (GNAT.SHA512.Digest ("password"))
```

The resulting SHA512 string for “password” is this 128 bytes string:

```
b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8b-
b980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86
```

1.4.5 v20 to v22 transition

List of obsolete v20 functions, superseded by Gnoga.Server.Database functions.

- Column_Integer, Column_Text

  Use Gnoga.Server.Database.Field_Value and Gnoga.Server.Database.Field_Decimals.

- Column_Count

  Use Gnoga.Server.Database.Number_Of_Fields.

- Column_Type

  Use Gnoga.Server.Database.Field_Type.

- Error

  Use Gnoga.Server.Database.Error_Message.

- Exec

  Use Gnoga.Server.Database.Execute_Query.

- Last_Insert_RowID

  Use Gnoga.Server.Database.Insert_ID.Insert_ID.

- Open

  Use Gnoga.Server.Connect.

- Prepare

  Use Gnoga.Server.Query.

- Reset

  Use Gnoga.Server.Close.

- Step

  Use Gnoga.Server.Next.

2 Gnoga

2.1 Callstack to display text

Simplified pathway to display a simple text:

```
TestGui
   Gui.Content_Set_Title (Object, "User Help Title");
   Gui.Content_Set_Text (Object, “This is the help text”);
|
V
```

```
v22.Gui

   procedure Content_Set_Title (Object : in out GGB.Base_Type'Class; Ti-
tle : String) is
      App : constant App_Access := App_Access (Object.Connection_Data);
   begin
      App.Content_Header.Text (Title);
   end Content_Set_Title;

     procedure Content_Set_Text (Object : in out GGB.Base_Type'Class;
Text : String) is
      App : constant App_Access := App_Access (Object.Connection_Data);
   begin
      App.Content_Text.Text (Text);
   end Content_Set_Text;
|
V

Gnoga.Gui.View
   Content_Text : aliased GGV.View_Type;
|
V

Gnoga.Gui.Element (968)

   procedure Text (Element : in out Element_Type; Value : in String) is
   begin
      Element.jQuery_Execute ("text ('" & Escape_Quotes (Value) & "');");
   end Text;
```

2.2 Gnoga tips

From https://github.com/alire-project/gnoga/blob/master/TIPS.md

Many tips will be found also in source code specification for types and subprograms.

2.2.1 Want to take an HTML snapshot of your page

Gnoga.Server.Template_Parser.Write_String_To_File ("site.html", Main_Window.Document.Document_Element.Outer_HTML);

2.2.2 Hidden and remove

Use View.Hidden to remove from browser, then View.Remove to remove from DOM. When View finalizes on the Ada side it will also tell the browser to reclaim the elements memory as well.

2.2.3 Add scrollbars

View.Overflow or View.Overflow_X or View.Overflow_Y with Visible, Hidden, Scroll, Auto.

2.2.4 Best results with the Grid view

When using the Grid view you always get best results when placing Views in to the grids instead of using the individual cell directly. When encapsulating what you want

in to another container (especially if its size is known) will get a more predictable layout of the underlying table.

2.2.5 Gnoga side memory leaks

If I create an object dynamically and add it to the DOM via create on a parent. If I later free that object manually and replace it with a newly created version (still set to dynamic), does that cause any memory leaks?

If you free the memory Ada's runtime will finalize the object which will remove any references to it in the Gnoga cache on the browser side. If you leaving "dangling pointer" on the Ada side you will of course have a leak. (Unless of course they were marked dynamic before creation and so the parent Gnoga view has a reference and will deallocate it when it finalizes)

2.2.6 Browser side memory

Does the DOM remember every object I add to a parent or will freeing it remove it completely from the DOM?

On the browser side elements created by the Gnoga framework have a reference in a global cache called gnoga[]. When the Gnoga object on the Ada side is finalized, the reference in the cache is removed. As long as that element is not in the DOM, the browser will release the object's memory as part of its garbage collection on the browser side.

2.2.7 Using Dynamic

Would something like this cause any problems (memory leak or otherwise)?

If you are using dynamic you shouldn't be deallocating manually. If you do then when the garbage collection starts with App.Console is finalized it will try to deallocate already deallocated blocks of memory. Unless you have extreme memory constraints, just overwrite the pointer since the parent view will take care of things or do not use dynamic and deallocate on connection close.

2.2.8 Raw JS execution

Use Gnoga.Server.Connection.Execute_Script for raw JS execution (there is both a procedure and function version). You can get the Connection ID from any object on a connection (View.Connection_ID, etc).

2.2.9 Define window Title and the closed connection message

To define at connection time the window Title and the closed connection message use Main_Window.Document.Title and Gnoga.Server.Connection.HTML_On_Close.

2.2.10 Don't use Unrestricted_Access

In general if you find yourself needing to use Unrestricted_Access you are likely going to have issues. For the "tutorials" it was used since it was wanted to keep everything to a single procedure, etc. but perhaps was not the smartest thing to have laying around. Try and keep handlers at package level when possible is a good principle to live by.

2.2.11 Get a hold on a child element, on the gnoga side, knowing its ID?

My_Button.Attach_Using_Parent (View, ID => "my_button");

2.2.12 Setting any callbacks to null before deleting visual elements

That would remove any race conditions.

2.2.13 Prevent the user to close the browser window

There is no way to prevent the user to close the browser window.

2.2.14 Knowing what row the submit button was on

An On_Submit_Handler is set in the table widget. It gets called for the submit buttons. How to access anything in the handler that lets me know what row the submit button was on and how to access the view data structure for the row?

You will need to add an On_Focus handler to the forms, you can write a single handler and just attach to all the widgets. Have it store the ID or a pointer to the last focused widget. The browser doesn't store or report the focus so no other way to get it.

2.2.15 Remove first, then release it

Just freeing an object on the Gnoga side will not remove it from the DOM, that is intended. You need to call Remove first.

2.2.16 Avoid flickering

To avoid two visual elements to disappear and then reappear, somewhat slowly and flickering. Create visual element that is hidden. Fill it in. Hide first and show second. Can either delete the old one or recycle it.

2.2.17 Colors

Colors in Gnoga.Types.Colors can be displayed or chosen from www.w3schools.com/cssref/css_colors.asp or www.w3schools.com/colors/colors_picker.asp.

2.2.18 Use browser console

Want to see what is going on in the browser console ? Turn debug on in your boot page <script>var gnoga_debug = true;</script> or use debug.html.

2.2.19 Proper use of know what row the submit button was on

Trying to use Connection_Data in a handler bounded to On_Destroy_Handler did not work when closing the browser window because at that point Object.Connection_Data returns null. Use instead On_Before_Unload_Handler. Note: it is not effective when only closing the connection.

2.2.20 How about reloading the whole page when On_Resize is called?

Call Main_Window.Location.Reload.

2.2.21 The browser refresh button case

If the user clicks the refresh button on the browser, it breaks the connection and creates a new one. This is extremely annoying, since the program starts over from scratch.
Refresh implies a lost connection and new session on web browsers, that is expected functionality. Local storage could be used on the browser side (Gnoga.Client.Storage) to store session information or any other data on the client side and use that data after a refresh or even weeks later to restore them to some state in your software.

2.2.22 Display text when connection is closed

HTML_On_Close text is displayed only when connection is broken and not when connection is closed. In order to display some text before closing connection, remove current view, create a new one with your text and then close connection, for instance:
App.My_View.Remove; View.Create (App.My_Window.all);
View.Put_Line ("Application exited.");
App.My_Window.Close_Connection;

2.2.23 Generate Gnoga Api documentation

You want to browse through Gnoga API, generate them with gnatdoc: $ make rm-docs $ open docs/html/gnoga_rm/index.html or, in GnatStudio, just use : Analyse > Documentation > Generate Project.

2.2.24 Gnoga makefile usage

Command "make" alone or "make help" prints main make targets and make environnement variables.

2.2.25 How to create a button of a specified size?

Button.Create (Parent => Parent, Content => Content); creates a button of the default size for Content, with a thin border with rounded corners. Adding

Button.Box_Width (Value => Button_Size); Button.Box_Height (Value => Button_Size);
has no effect. But also adding Button.Border; results in a button of the desired size, with a fairly thick, black border with square corners. Adding (Width => "thin") to the call to Border may be desirable.

2.2.26 How do I add a class attribute to an element in code?

Use the class property: Gnoga.Gui.Element.Class, which replaces all class with the value of Class or the method Gnoga.Gui.Element.Add_Class which will add a class but not remove other classes that may already apply to the element.

2.2.27 Latency

Take care of latency, many subprograms do query the client side and wait for an answer, so they can be very slow if server and client are far from each other!

2.3 Events sorted by packages

From Pascal (Blady) document “Premiers pas avec Gnoga”.

Events	Packages	Lines
On_Connect_Handler	gnoga-application-multi_connect.ads	74
On_Resize_Handler	gnoga-gui-base.ads	345
On_Scroll_Handler	gnoga-gui-base.ads	350
On_Focus_Handler	gnoga-gui-base.ads	357
On_Blur_Handler	gnoga-gui-base.ads	362
On_Change_Handler	gnoga-gui-base.ads	367
On_Focus_In_Handler	gnoga-gui-base.ads	371
On_Focus_Out_Handler	gnoga-gui-base.ads	375
On_Input_Handler	gnoga-gui-base.ads	379
On_Reset_Handler	gnoga-gui-base.ads	383
On_Search_Handler	gnoga-gui-base.ads	390
On_Select_Handler	gnoga-gui-base.ads	394
On_Submit_Handler	gnoga-gui-base.ads	398
On_Click_Handler	gnoga-gui-base.ads	407
On_Mouse_Click_Handler	gnoga-gui-base.ads	412
On_Context_Menu_Handler	gnoga-gui-base.ads	418
On_Mouse_Right_Click_Handler	gnoga-gui-base.ads	423
On_Double_Click_Handler	gnoga-gui-base.ads	429
On_Mouse_Double_Click_Handler	gnoga-gui-base.ads	434
On_Mouse_Enter_Handler	gnoga-gui-base.ads	440
On_Mouse_Leave_Handler	gnoga-gui-base.ads	445
On_Mouse_Over_Handler	gnoga-gui-base.ads	450
On_Mouse_Out_Handler	gnoga-gui-base.ads	455
On_Mouse_Down_Handler	gnoga-gui-base.ads	460
On_Mouse_Up_Handler	gnoga-gui-base.ads	466
On_Mouse_Move_Handler	gnoga-gui-base.ads	472
On_Drag_Start_Handler	gnoga-gui-base.ads	480
On_Drag_Handler	gnoga-gui-base.ads	489
On_Drag_End_Handler	gnoga-gui-base.ads	493
On_Drag_Enter_Handler	gnoga-gui-base.ads	497
On_Drag_Leave_Handler	gnoga-gui-base.ads	501
On_Drop_Handler	gnoga-gui-base.ads	505
On_Character_Handler	gnoga-gui-base.ads	516
On_Wide_Character_Handler	gnoga-gui-base.ads	521
On_Key_Down_Handler	gnoga-gui-base.ads	526
On_Key_Up_Handler	gnoga-gui-base.ads	531
On_Key_Press_Handler	gnoga-gui-base.ads	536
On_Copy_Handler	gnoga-gui-base.ads	543
On_Cut_Handler	gnoga-gui-base.ads	547
On_Paste_Handler	gnoga-gui-base.ads	551
On_Create_Handler	gnoga-gui-base.ads	557
On_Destroy_Handler	gnoga-gui-base.ads	563
On_Child_Added_Handler	gnoga-gui-base.ads	572
On_Child_Removed_Handler	gnoga-gui-base.ads	577
On_Message_Handler	gnoga-gui-base.ads	582
On_Resize	gnoga-gui-base.ads	601
On_Create	gnoga-gui-base.ads	604
On_Destroy	gnoga-gui-base.ads	608
On_Child_Added	gnoga-gui-base.ads	612
On_Child_Removed	gnoga-gui-base.ads	616
On_Message	gnoga-gui-base.ads	621
On_Media_Abort_Handler	gnoga-gui-element-multimedia.ads	139
On_Media_Error_Handler	gnoga-gui-element-multimedia.ads	
On_Can_Play_Handler	gnoga-gui-element-multimedia.ads	151
On_Can_Play_Through_Handler	gnoga-gui-element-multimedia.ads	157
On_Duration_Change_Handler	gnoga-gui-element-multimedia.ads	164
On_Emptied_Handler	gnoga-gui-element-multimedia.ads	170
On_Ended_Handler	gnoga-gui-element-multimedia.ads	176
On_Loaded_Data_Handler	gnoga-gui-element-multimedia.ads	182
On_Loaded_Meta_Data_Handler	gnoga-gui-element-multimedia.ads	188
On_Load_Start_Handler	gnoga-gui-element-multimedia.ads	194
On_Pause_Handler	gnoga-gui-element-multimedia.ads	200
On_Play_Handler	gnoga-gui-element-multimedia.ads	205
On_Playing_Handler	gnoga-gui-element-multimedia.ads	210
On_Progress_Handler	gnoga-gui-element-multimedia.ads	215
On_Rate_Change_Handler	gnoga-gui-element-multimedia.ads	221

Events	Packages	Lines
On_Seeked_Handler	gnoga-gui-element-multimedia.ads	227
On_Seeking_Handler	gnoga-gui-element-multimedia.ads	233
On_Stalled_Handler	gnoga-gui-element-multimedia.ads	239
On_Suspend_Handler	gnoga-gui-element-multimedia.ads	245
On_Time_Update_Handler	gnoga-gui-element-multimedia.ads	251
On_Volume_Change_Handler	gnoga-gui-element-multimedia.ads	257
On_Waiting_Handler	gnoga-gui-element-multimedia.ads	262
On_Message	gnoga-gui-element-multimedia.ads	273
On_Resize	gnoga-gui-plugin-ace_editor.ads	157
On_Open_Handler	gnoga-gui-plugin-jqueryui-widget.ads	
On_Close_Handler	gnoga-gui-plugin-jqueryui-widget.ads	150
On_Message	gnoga-gui-plugin-jqueryui-widget.ads	164
On_Resize	gnoga-gui-view-card.ads	100
On_Child_Added	gnoga-gui-view-console.ads	66
On_Resize	gnoga-gui-view-docker.ads	119
On_Child_Added	gnoga-gui-view.ads	187
On_Abort_Handler	gnoga-gui-window.ads	270
On_Error_Handler	gnoga-gui-window.ads	274
On_Before_Unload_Handler	gnoga-gui-window.ads	278
On_Hash_Change_Handler	gnoga-gui-window.ads	283
On_Orientation_Change_Handler	gnoga-gui-window.ads	288
On_Storage_Handler	gnoga-gui-window.ads	293
On_Resize	gnoga-gui-window.ads	304
On_Child_Added	gnoga-gui-window.ads	309
On_Message	gnoga-gui-window.ads	314
On_Connect_Handler	gnoga-server-connection.ads	111
On_Post_Request_Handler	gnoga-server-connection.ads	134
On_Post_Handler	gnoga-server-connection.ads	
On_Post_File_Handler	gnoga-server-connection.ads	152

2.4 Keyboard handling with On_Key_Pressed

On_Connect_Pressed handler is set in the On_Connect handler, which is itself set on Gui.Setup procedure, usally the first Gnoga procedure called in v22 application.

On_Key_Pressed is useful to directly get ASCII codes from keyboard. On_Key_Pressed reflects ASCII table for printable characters as lowercase and uppercase letters, numbers and punctuation.

On_Key_Pressed is not relevant for characters lower than 32.

- Example

a is 97, A is 65, & is 38 and so on.

2.5 Keyboard handling with On_Key_Down

On_Connect_Down handler is set in the On_Connect handler, which is itself set on Gui.Setup procedure, usally the first Gnoga procedure called in v22 application.

On_Key_Down is not intended to manipulate ASCCI codes like On_Key_Pressed but useful to get the keyboard state. By example pressing [a] or [Shift] + [a] allways returns a Key_Code value of 65 but pressing [Shift] only returns Key_Code 16.

On_Key_Down reflects Keyboard Key codes and not ASCII codes.

Key	Key code	Enter	13	ALT	18
Backspace	8	Shift	16	Pause/ Break	19
Tab	9	CTRL	17	Caps Lock	20

ESC	27	i	73	+[Num Lock]	107
Page Up	33	j	74	-[Num Lock]	109
Page Down	34	k	75	.[Num Lock]	110
End	35	l	76	/[Num Lock]	111
Home	36	m	77	F1	112
Arrow Left	37	n	78	F2	113
Arrow Up	38	o	79	F3	114
Arrow Right	39	p	80	F4	115
Arrow Down	40	q	81	F5	116
Insert	45	r	82	F6	117
Delete	46	s	83	F7	118
0	48	t	84	F8	119
1	49	u	85	F9	120
2	50	v	86	F10	121
3	51	w	87	F11	122
4	52	x	88	F12	123
5	53	y	89	Num Lock	144
6	54	z	90	Scroll Lock	145
7	55	Windows	91	My Computer	182
8	56	Right Click	93	My Calculator	183
9	57	0[Num Lock]	96	,<	188
;:	59	1[Num Lock]	97	.>	190
=+	61	2[Num Lock]	98	/?	191
a	65	3[Num Lock]	99	`~	192
b	66	4[Num Lock]	100	[{	219
c	67	5[Num Lock]	101	|	220
d	68	6[Num Lock]	102	]}	221
e	69	7[Num Lock]	103	‘”	222
f	70	8[Num Lock]	104		
g	71	9[Num Lock]	105		
h	72	*[Num Lock]	106		

On_Key_Down gets theses information through this structure:

```
   type Keyboard_Event_Record is record
      Message  : Keyboard_Message_Type := Unknown;
      Key_Code : Integer;
      Key_Char : Wide_Character;
      Alt      : Boolean               := False;
      Control  : Boolean               := False;
      Shift    : Boolean               := False;
      Meta     : Boolean               := False;
   end record;
```

With:

```
   type Keyboard_Message_Type is (Unknown, Key_Down, Key_Up, Key_Press);
```

2.6 HTTPS/TLS setup

2.6.1 Direct connection

Using layouts_ssl.adb from Gnoga examples.

- create TLS certificate

Delete server.crt et server.key.

With the station name, for example, "ro6.genesix.org", create the certificate via Let's Encrypt with your favorite generator1 and install it, for example according to this tree structure:

```
├── certs
│   ├── ca.cer
│   ├── fullchain.cer
│   ├── ro6.genesix.org.cer
│   ├── ro6.genesix.org.conf
│   ├── ro6.genesix.org.csr
│   ├── ro6.genesix.org.csr.conf
│   └── ro6.genesix.org.key
├── html
│   ├── debug.html
│   ├── favicon.ico
│   └── robots.txt
├── js
│   ├── boot.js
│   └── jquery.min.js
├── layouts_ssl
├── layouts_ssl.adb
└── test_ssl.gpr
```

- Changes to layouts_ssl.adb

For a "true" https experience, it may be preferable to choose port 443 instead of 8443. Since ports below 1024 require root rights via sudo, this option should be considered depending on your rights.

Comment in layouts_ssl.adb, around line 143:

```
   Gnoga.Server.Connection.Secure.Register_Secure_Server
     (Certificate_File => Gnoga.Server.Application_Directory & "/test_ssl/server.crt",
      Key_File => Gnoga.Server.Application_Directory & "/test_ssl/server.key",
      Port     => 8_443, Disable_Insecure => False);
```

And replace with:

```
   Gnoga.Server.Connection.Secure.Register_Secure_Server
     (Certificate_File => "certs/ro6.genesix.org.cer",
      Key_File => "certs/ro6.genesix.org.key",
      Port => 443, Disable_Insecure => False);
```

Where Port => 8443 if you don't have sudo or root rights.

- Create test_ssl_ins.gpr

Puisque gnoga.gpr est déjà installé dans l'environnement de développement :

Comment in:

```
with "../src/gnoga.gpr"
```

Remplace with:

```
-         with "../src/gnoga.gpr"
+         with "gnoga.gpr"
```

2.6.2 Connection through proxy

- Without load balancer

  For standard use with a single database.

```
#------------------------------------------------------------------------------
# demo01.v22.soweb.io - proxy websocket
#------------------------------------------------------------------------------
#
# 20170613 - Initial release
# 20170716 - Error managment
# 20170731 - DNS-01 method
# 20231102 - Proxy websocket
#
#------------------------------------------------------------------------------

# For a working websocket proxy, HTTP protocol must be exclusively HTTP 1.1 with no session cache

map $http_upgrade $connection_upgrade {
    default upgrade;
    '' close;
}

# http port 80 switching to https 443
server {listen 80; server_name demo01.v22.soweb.io; return 301 https://$server_name$request_uri; }

server {

    #------------------------------------------------------------------------------
    # Server settings
    #------------------------------------------------------------------------------

    server_name               demo01.v22.soweb.io;
    access_log /var/log/nginx/demo01.v22.soweb.io.access.log;
    error_log  /var/log/nginx/demo01.v22.soweb.io.error.log;

    location / {
        proxy_pass http://localhost:8001;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
        proxy_set_header Host $host;
    }

    #------------------------------------------------------------------------------
    # SSL Settings
    #------------------------------------------------------------------------------

    listen 443 ssl; # ssl in parameter is preferable to 'ssl on', according to the nginx doc
    ssl_protocols TLSv1.2 TLSv1.3; # Authorized TLS versions

    # Site certificates
    ssl_certificate         /etc/acme/certs/demo01.v22.soweb.io/fullchain.cer;
    ssl_certificate_key     /etc/acme/certs/demo01.v22.soweb.io/demo01.v22.soweb.io.key;
    ssl_trusted_certificate /etc/acme/certs/demo01.v22.soweb.io/fullchain.cer; # For OCSP Stapling

    # Specifies that server ciphers should be preferred over client ciphers when using the SSLv3 and TLS proto-
cols.
    ssl_prefer_server_ciphers on;

    # https://wiki.mozilla.org/Security/Server_Side_TLS
```

```
     # Cipher suites (TLS 1.2) :  ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-
GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-
AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
    # Cipher suites (TLS 1.3) :  TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256

     ssl_ciphers  ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-
RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-
RSA-AES256-GCM-SHA384:TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256;

    # Disable session ticket
     ssl_session_tickets off;  # See https://community.letsencrypt.org/t/errors-from-browsers-with-ssl-session-
tickets-off-nginx/18124/4

    # Strong Diffie-Hellman group
    ssl_dhparam /etc/ssl/certs/dhparam.pem;

    # Certificate validity check at TLS handshake
    # OSCP = Online Certificate Status Protocol - RFC4366
    ssl_stapling on;
    ssl_stapling_verify on;
    ssl_stapling_responder http://r3.o.lencr.org;    # See https://letsencrypt.org/docs/lencr.org

    # If resolver not addded, the resolver defaults to the server´s DNS default.
    # ipv6=off to avoid "OCSP response not successful (6: unauthorized) while requesting certificate status"
    resolver 80.67.169.12 80.67.169.40 ipv6=off valid=300s;  # Resolver for ssl_stapling - ns0.fdn.fr ns1.fdn.fr
    resolver_timeout 15s;                                    # Short resolution timeout (default is 30s)

    #-------------------------------------------------------------------------------
    # Headers
    #-------------------------------------------------------------------------------

    # Force browser to strictly HTTPS
    add_header Strict-Transport-Security "max-age=63072000; includeSubdomains" always;

}

#-------------------------------------------------------------------------------
# EOF
#-------------------------------------------------------------------------------
```

- With load balancer

  For a loaded application with, for example, database replication.

```
#-------------------------------------------------------------------------------
# demo01.v22.soweb.io - proxy websocket
#-------------------------------------------------------------------------------
#
# 20170613 - Initial release
# 20170716 - Error managment
# 20170731 - DNS-01 method
# 20231102 - Proxy websocket
#
#-------------------------------------------------------------------------------

# For a working websocket proxy, HTTP protocol must be exclusively HTTP 1.1 with no session cache

map $http_upgrade $connection_upgrade {
    default upgrade;
    '' close;
}

upstream websocket {

  server 192.168.1.11:8001;
  server 192.168.1.12:8001;
  server 192.168.1.13:8001;
  #server ...
  #server ...

  # allways after servers declarations
  keepalive 50;
}


# http port 80 switching to https 443
server {listen 80; server_name demo01.v22.soweb.io; return 301 https://$server_name$request_uri; }

server {

    #-------------------------------------------------------------------------------
    # Server settings
    #-------------------------------------------------------------------------------

    server_name               demo01.v22.soweb.io;
    access_log /var/log/nginx/demo01.v22.soweb.io.access.log;
```

```
    error_log  /var/log/nginx/demo01.v22.soweb.io.error.log;

    location / {
        proxy_pass http://websocket;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
        proxy_set_header Host $host;
    }

    #------------------------------------------------------------------------------
    # SSL Settings
    #------------------------------------------------------------------------------

    listen 443 ssl; # ssl in parameter is preferable to 'ssl on', according to the nginx doc
    ssl_protocols TLSv1.2 TLSv1.3; # Authorized TLS versions

    # Site certificates
    ssl_certificate         /etc/acme/certs/demo01.v22.soweb.io/fullchain.cer;
    ssl_certificate_key     /etc/acme/certs/demo01.v22.soweb.io/demo01.v22.soweb.io.key;
    ssl_trusted_certificate /etc/acme/certs/demo01.v22.soweb.io/fullchain.cer; # For OCSP Stapling

    # Specifies that server ciphers should be preferred over client ciphers when using the SSLv3 and TLS proto-
cols.
    ssl_prefer_server_ciphers on;

    # https://wiki.mozilla.org/Security/Server_Side_TLS
     # Cipher suites (TLS 1.2) : ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-
GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-
AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
    # Cipher suites (TLS 1.3) : TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256

     ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-
RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-
RSA-AES256-GCM-SHA384:TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256;

    # Disable session ticket
     ssl_session_tickets off; # See https://community.letsencrypt.org/t/errors-from-browsers-with-ssl-session-
tickets-off-nginx/18124/4

    # Strong Diffie-Hellman group
    ssl_dhparam /etc/ssl/certs/dhparam.pem;

    # Certificate validity check at TLS handshake
    # OSCP = Online Certificate Status Protocol - RFC4366
    ssl_stapling on;
    ssl_stapling_verify on;
    ssl_stapling_responder http://r3.o.lencr.org;    # See https://letsencrypt.org/docs/lencr.org

    # If resolver not addded, the resolver defaults to the server´s DNS default.
    # ipv6=off to avoid "OCSP response not successful (6: unauthorized) while requesting certificate status"
    resolver 80.67.169.12 80.67.169.40 ipv6=off valid=300s; # Resolver for ssl_stapling - ns0.fdn.fr ns1.fdn.fr
    resolver_timeout 15s;                                  # Short resolution timeout (default is 30s)

    #------------------------------------------------------------------------------
    # Headers
    #------------------------------------------------------------------------------

    # Force browser to strictly HTTPS
    add_header Strict-Transport-Security "max-age=63072000; includeSubdomains" always;

}

#------------------------------------------------------------------------------
# EOF
#------------------------------------------------------------------------------
```

Implementation notes

It's not a problem until it happens twice.
Jim Van Sickle

1 Introduction

Implementation notes is subject to change. To be seen for now as a working documentation chapter.

2 API Rest

geta : aws ada api rest

max url length : 2K (some browsers) or 8K per convention

https://en.wikibooks.org/wiki/Ada_Programming/Libraries/Web/AWS

https://github.com/BrentSeidel/BBS-BBB-Ada/tree/master
https://github.com/BrentSeidel/BBS-Ada

https://docs.adacore.com/aws-docs/aws

https://restfulapi.net/rest-put-vs-post
https://www.baeldung.com/rest-http-put-vs-post
https://blog.apilayer.com/an-ultimate-guide-to-http-put-vs-post-in-rest-api-in-2023

Use get & post (see API IONOS)

2.1 OVH

2.1.1 Links

https://eu.api.ovh.com
https://eu.api.ovh.com/console-preview
https://docs.ovh.com/fr/api/api-v2

https://help.ovhcloud.com/csm/en-api-getting-started-ovhcloud-api?id=kb_article_view&sysparm_article=KB0042777

https://help.ovhcloud.com/csm/en-gb-api-console-exploration?id=kb_article_view&sysparm_article=KB0057325

2.1.2 SMS

https://help.ovhcloud.com/csm/fr-sms-sending-via-api-php?id=kb_article_view&sysparm_article=KB0051373
https://help.ovhcloud.com/csm/fr-sms-api-cookbook?id=kb_article_view&sysparm_article=KB0039149

2.1.3 Messages

- The key set does not have the rights for this command

  Examples :

```
{"message":"This call has not been granted","httpCode":"403 Forbidden",
"errorCode" :"NOT_GRANTED_CALL"}

{"class":"Client::Forbidden","message":"This call has not been granted"}
```

3 Configuration file

3.1 MySQL

```
# ------------------------------------------------------------------------------
#  testgui.cfg - Configuration file
# ------------------------------------------------------------------------------
#
#  20230920-181257 - First release
#
# ------------------------------------------------------------------------------

[Connection]
Domain = 192.168.0.36 #localhost
Port = 8080
Tls_Certificate = A relative path to .cer file
Tls_Private_Key = A relative path to .key file

[Database]
Type = MySQL
Host = 192.168.0.243
Port = 3306
Name = testgui
User = user_testgui
Password = user_pass

# ------------------------------------------------------------------------------
#  EOF
# ------------------------------------------------------------------------------
```

3.2 SQLite

```
# ------------------------------------------------------------------------------
#  testgui.cfg - Configuration file
# ------------------------------------------------------------------------------
#
#  20230920-181257 - First release
#
# ------------------------------------------------------------------------------

[Connection]
Domain = 192.168.0.36 #localhost
Port = 8080
Tls_Certificate = A relative path to .cer file
Tls_Private_Key = A relative path to .key file

[Database]
Type = SQLite
Name = testgui

# ------------------------------------------------------------------------------
#  EOF
# ------------------------------------------------------------------------------
```

4 CRC

https://crccalc.com [outil interactif 8/16/32 bits]

https://sourceforge.net/p/avr-ada/wiki/CRC

4.1 CRC 16

https://srecord.sourceforge.net/crc16-ccitt.html

http://computer-programming-forum.com/44-ada/ce7afed4795fb0e4.htm
https://www.avrfreaks.net/s/topic/a5C3l000000UaabEAC/t154443
https://www.nongnu.org/avr-libc/user-manual/group__util__crc.html
https://www.carnetdumaker.net/snippets/29

4.2 CRC 32

https://rosettacode.org/wiki/CRC-32

5 Databases

5.1 Introduction

<<<<TODO>>>

5.2 Benchmarking

5.2.1 Locust

https://simonwillison.net/2022/Oct/23/datasette-gunicorn/#benchmarking-sqlite

5.2.2 Exists

- Description

  Returns True if file or directory Name exists.

- Usage

```
function Exists (Name : String) return Boolean
```

- Example

```
if Fls.Exists (HAC_Dir & "/hac") then
   Tio.Put_Line ("HAC installation is done :)");
end if;
```

6 Database MySQL

6.1 Charset and collation

Charset must be : utf8mb4
Collation must be : utf8mb4_unicode_520_ci

Never use utf8mb4_general_ci. For new implementations, always use utf8mb4_unicode_520_ci against the older utf8mb4_unicode_ci. See your preferred search engine for details.

6.2 Transactions sequences

DB.Execute_Query ("START TRANSACTION");
DB.Execute_Query ("SAVEPOINT SV_Index_Exists");
 <<<work>>>
DB.Execute_Query ("COMMIT");

<<<TODO>>>

7 Database SQLite

7.1 Transactions sequences

DB.Execute_Query ("BEGIN TRANSACTION");
DB.Execute_Query ("SAVEPOINT SV_Index_Exists");
 <<<work>>>
DB.Execute_Query ("COMMIT");

<<<TODO>>>

7.2 Compatibility with MySQL

7.2.1 Problem description

Number_Of_Rows is not implemented in SQLite: this is acceptable because it is clearly expressed in the doc and it can be implemented with the reservations below.

Affected_Rows is not implemented correctly for SQLite: this is annoying because it's not expressed at all as Gnoga doc says: executes a SQL query and returns the number of rows affected, but affected_Rows calls sqlite3_total_changes() which only handles modifications (i.e. only INSERT, UPDATE or DELETE) and not the result of the query. A simple SELECT * table will return 0 even if the table is full to bursting. 2) Implementation has an impact on performance (see below).

Execute_Update is not implemented correctly for SQLite: since, after calling Execute_Query, we call Affected_Rows, we end up with the same problem.

7.2.2 Suggested corrections

Correcting these inconsistencies is not difficult, but the result will range from transparent to very slow (depending on the type of query and/or the volume of data returned, since the entire recordset have to be iterated).

Number_Of_Rows: As the name don't suggests, returns the number of rows in the record set, not in the table. Can be implemented with the above reservations by iterating through the entire record set.

Affected_Rows: you'll need to remember the last query performed by Execute_Update (in the connection record, next to UTF8_String). If this query contains INSERT, UPDATE or DELETE, we return the result unchanged; if it contains a SELECT *| Column_Name, we transform it into a SELECT COUNT(*|Column_Name). If the query contains anything else, we execute a standard Query with the above reservations, since we have to iterate through the entire record set.

7.2.3 Suggested implementations

Two possibilities:
- Either I implement it at Gnoga level, but Pascal (Gnoga's maintainer) has to agree. It's an imperfect hack, but it'll be better than this unusable existing one.
- Or I can implement this in v22 and ban the use of Execute_Query, Execute_Update, Number_Of_Rows in the docs in favor of functions with identical names but prefixed v22.Sql, which is pretty dirty and wouldn't solve the problem of these rogue functions in Gnoga.Server.Database.

7.3 Add-ons

7.3.1 Mycelite

https://github.com/mycelial/mycelite

7.3.2 Compile time options

https://www.sqlite.org/compile.html

7.4 Concurrent access

7.4.1 Activating WAL mode

```
PRAGMA journal_mode=WAL;
sqlite3 github.db 'PRAGMA journal_mode=WAL;'
```

7.4.2 Managing busy error

- Setting delay [internal control]

  <<<TODO>>>

  https://www.sqlite.org/pragma.html#pragma_busy_timeout

- Call back function in Ada [external control]

  <<<TODO>>>

  https://www.sqlite.org/c3ref/busy_handler.html

7.4.3 Experimental

https://stackoverflow.com/questions/4060772/sqlite-concurrent-access

7.4.4 Production

SQLite can handle concurrent access in WAL [Write Ahead Log] mode. In this case all writes are appended to a WAL temporary fille which is periodically merged with the original database.

When SQLite is searching for something it would first check this temporary file and if nothing is found proceed with the main database file.

As a result, readers don't compete with writers and performance is much better compared to the traditional SQLite Rollback journaled mode.

https://www.sqlite.org/walformat.html
https://fly.io/blog/sqlite-internals-wal [How SQLite Scales Read Concurrency]

7.5 FAQ

7.5.1 Release history

https://www.sqlite.org/draft/changes.html

7.5.2 SQL virtual machine

https://fly.io/blog/sqlite-virtual-machine

7.5.3 Opening a new database connection

https://www.sqlite.org/c3ref/open.html
https://www.sqlite.org/uri.html
https://stackoverflow.com/questions/56416437/confusion-about-uri-path-to-configure-sqlite-database

7.5.4 UPSERT how to

- First pass (record does not exists)

  Update on a non existent record
  UPDATE System SET Value='0.0' WHERE Parameter='Shema_Version';
  Success : 0 record(s) affected

  Create ok because Changes() = 0 (no change from the last operation)
  INSERT INTO System (Parameter, Value) SELECT 'Shema_Version', '0.0' WHERE (Select Changes() = 0);
  Success : 1 record(s) affected

- Second pass (record already exists)

  Update an existent record successfull
  UPDATE System SET Value='0.1' WHERE Parameter='Shema_Version';
  Success : 1 record(s) affected

  No create since Changes() = 1 (last operation has change something)
  INSERT INTO System (Parameter, Value) SELECT 'Shema_Version', '0.1' WHERE (Select Changes() = 0);
  Success : 0 record(s) affected

7.5.5 File naming

Never have a filename with a part of it including reserved word.

v22.Gui.Connect is good but v22.Gui.Access is bad. This latter package will be silently ignored without apparent reason nor warning message.

7.6 Replication

7.6.1 Distributed SQLite

https://fly.io/docs/litefs
https://fly.io/docs/litefs/how-it-works
https://fly.io/blog/litefs-cloud
https://fly.io/blog/introducing-litefs
https://news.ycombinator.com/item?id=36602970

- Installing LiteFS without Docker

  https://github.com/superfly
  https://github.com/superfly/litefs/releases
  https://fly.io/docs/litefs/getting-started-fly

7.6.2 Cr-SQLite

https://github.com/vlcn-io/cr-sqlite

7.6.3 LiteStream

https://litestream.io

https://github.com/benbjohnson/litestream
https://fly.io/blog/all-in-on-sqlite-litestream
https://litestream.io/guides/systemd

7.7 Utilities

7.7.1 Sqlite-utils

https://sqlite-utils.datasette.io/en/stable/cli.html

7.7.2 SQLCrush

https://github.com/coffeeandscripts/sqlcrush

7.7.3 Visidata

https://www.visidata.org

```
user@system: sudo apt install visidata
```

7.8 Why SQLite ?

https://tailscale.com/blog/database-for-2022
https://simonwillison.net/2021/Jul/28/baked-data

8 Encryption

8.1 Key expansion

A SHA256 could be used as key expansion routine [from 32 to 256 bits]

https://ritul-patidar.medium.com/key-expansion-function-and-key-schedule-of-des-data-encryption-standard-algorithm-1bfc7476157

https://www.researchgate.net/figure/Structure-of-RC4-key-scheduling-process_fig8_267858656

8.2 RC4

geta : RC4 minimum key length

https://en.wikipedia.org/wiki/RC4
https://en.wikipedia.org/wiki/Stream_cipher
https://www.researchgate.net/publication/337743183_Modernized_RC4_encryption_algorithm/link/5de7eb71a6fdc-c28370658dd/download
https://iopscience.iop.org/article/10.1088/1757-899X/420/1/012131
https://iopscience.iop.org/article/10.1088/1757-899X/680/1/012025

https://en.wikipedia.org/wiki/Variably_Modified_Permutation_Composition
https://microchipdeveloper.com/harmony:middleware-crypto

http://www.winpicprog.co.uk/pic_tutorial.htm

https://github.com/cforler/Ada-Crypto-Library

8.3 SHA-1

https://stackoverflow.com/questions/8860635/getting-the-sha1-block-from-gnat-sha1
https://en.wikipedia.org/wiki/SHA-1

8.4 SHA-256

https://emn178.github.io/online-tools/sha256.html (interactive tool)

https://en.wikipedia.org/wiki/SHA-2
https://rosettacode.org/wiki/SHA-256

https://github.com/oilulio/Microcontroller-hashes/blob/master/sha256.c
https://www.mdpi.com/2071-1050/13/8/4324

9 JSON

https://datatracker.ietf.org/doc/html/rfc7159

9.1 Gnatcoll

Gnatcoll is preferred, as we already use it.

9.1.1 Usage

https://docs.adacore.com/live/wave/gnatcoll-core/html/gnatcoll-core_ug/json.html

9.2 Json-Ada

Json-Ada is way more faster when dealing with huge JSON files. See below.

https://github.com/onox/json-ada

9.3 Benchmarking

https://github.com/AJ-Ianozi/json_benchmark

With a ~ 450 MB JSON file.

9.3.1 GNATCOLL

GNATCOLL.JSON read is 28.289018000 seconds
GNATCOLL.JSON iterate is 0.014264000 seconds

9.3.2 json-ada

json-ada read is 3.295961000 seconds
json-ada iterate is 0.009062000 seconds

10 List forward & backward

With LIMIT = 10

10.1 First display

```
SELECT Login FROM Sys_Users WHERE Login >= '' ORDER BY Login LIMIT 10

admin
admin2
admin3
alpha
bravo
charlie
delta
echo
foxtrot
golf
```

10.2 Forward

```
SELECT Login FROM Sys_Users WHERE Login >= 'golf' ORDER BY Login LIMIT 10

golf
hotel
india
juliet
kilo
lima
mike
november
oscar
papa < _Last
```

```
SELECT Login FROM Sys_Users WHERE Login >= 'papa' ORDER BY Login LIMIT 10

papa
quebec
romeo
sierra
tango
uniform
victor
whiskey
xray
yankee < _Last
```

```
SELECT Login FROM Sys_Users WHERE Login >= 'yankee' ORDER BY Login LIMIT 10

yankee
zoulou < _Last
```

if _Max < 10:

```
SELECT Login FROM Sys_Users WHERE Login <= 'zoulou' ORDER BY Login DESC LIMIT 10

zoulou
yankee
xray
whiskey
victor
uniform
tango
sierra
romeo
```

```
quebec < _Last
```

Then:

```
SELECT Login FROM Sys_Users WHERE Login >= 'quebec' ORDER BY Login LIMIT 10

quebec < _First
romeo
sierra
tango
uniform
victor
whiskey
xray
yankee
zoulou
```

10.3 Backward

```
SELECT Login FROM Sys_Users WHERE Login <= 'quebec' ORDER BY Login DESC LIMIT 10

quebec
papa
oscar
november
mike
lima
kilo
juliet
india
hotel < _Last
```

Then:

```
SELECT Login FROM Sys_Users WHERE Login >= 'hotel' ORDER BY Login LIMIT 10

hotel < _First
india
juliet
kilo
lima
mike
november
oscar
papa
quebec
```

```
SELECT Login FROM Sys_Users WHERE Login <= 'hotel' ORDER BY Login DESC LIMIT 10

hotel
golf
foxtrot
echo
delta
charlie
bravo
alpha
admin3
admin2 < _Last
```

Then:

```
SELECT Login FROM Sys_Users WHERE Login >= 'admin2' ORDER BY Login LIMIT 10

admin2 < _First
admin3
```

```
alpha
bravo
charlie
delta
echo
foxtrot
golf
hotel
```

```
SELECT Login FROM Sys_Users WHERE Login <= 'admin2' ORDER BY Login DESC LIMIT 10

admin2
admin < _Last
```

if _Max < 10:

```
SELECT Login FROM Sys_Users WHERE Login >= 'admin' ORDER BY Login LIMIT 10

admin
admin2
admin3
alpha
bravo
charlie
delta
echo
foxtrot
golf
```

11 Logs

11.1 Standard mode

```
20231031-101841 - INIT    - MSG - v22 Framework - GUI test program
20231031-101841 - INIT    - MSG - Copyright (C) Sowebio SARL 2020-2023, according to
LGPLv3
20231031-101841 - INIT    - MSG - testgui v0.2 - v22 v0.1 - build 2023-10-31 10:18:26

20231031-101841 - INIT    - MSG - TestGui.Init.App > Configuration file testgui.cfg
loaded
20231031-101841 - INIT    - MSG - TestGui.On_Connect > Starting Gnoga server
20231031-101841 - INIT    - MSG - TestGui.SQL_Ping > Armed for 3600s cycles
20231031-101853 - INIT    - MSG - TestGui.On_Connect > Login screen
20231031-101929 - INIT    - MSG - Gui.User_Login > New user on logging screen
20231031-101929 - INIT    - MSG - Gui.User_Login > User alpha connected
20231031-103450 - INIT    - MSG - TestGui.On_Connect > Login screen
20231031-103502 - INIT    - MSG - Gui.User_Login > New user on logging screen
20231031-103502 - INIT    - MSG - Gui.User_Login > User alpha connected
20231031-112414 - INIT    - MSG - TestGui.On_Connect > Login screen
20231031-125618 - INIT    - MSG - TestGui.On_Connect > Login screen
20231031-125636 - INIT    - MSG - Gui.User_Login > New user on logging screen
20231031-125636 - INIT    - MSG - Gui.User_Login > User alpha connected
```

11.2 Debug mode

```
20231031-101357 - INIT    - MSG - v22 Framework - GUI test program
20231031-101357 - INIT    - MSG - Copyright (C) Sowebio SARL 2020-2023, according to
LGPLv3
20231031-101357 - INIT    - MSG - testgui v0.2 - v22 v0.1 - build 2023-10-31 10:13:41
```

```
20231031-101357 - INIT     - MSG - TestGui.Init.App > Configuration file testgui.cfg
loaded
20231031-101357 - INIT    - DBG - Load Database_Pragma: journal_mode=WAL
20231031-101357 - INIT    - DBG - Load Database_Pragma: foreign_keys=ON
20231031-101357 - INIT    - MSG - TestGui.On_Connect > Starting Gnoga server
20231031-101357 - INIT    - MSG - TestGui.SQL_Ping > Armed for 3600s cycles
20231031-101433 - INIT    - DBG - User > On_Mgt
20231031-101433 - INIT    - MSG - TestGui.On_Connect > Login screen
20231031-101439 - INIT    - DBG - Dispensers > Demo_Mode_1
20231031-101525 - INIT    - DBG - Dispensers > Demo_Complex_Form_Create
20231031-101533 - INIT    - DBG - User > On_Mgt
20231031-101551 - INIT    - DBG - User > On_Mgt
20231031-101552 - INIT    - DBG - User > On_Adm
20231031-101553 - INIT    - DBG - Users > Main_Menu
20231031-101553 - INIT    - DBG - Gui.List > Start
20231031-101554 - INIT    - DBG - Key/Value: Users_List_1/alpha
20231031-101554 - INIT    - DBG - Key/First: Users_List_First/alpha
20231031-101554 - INIT    - DBG - Key/Value: Users_List_2/bravo
20231031-101554 - INIT    - DBG - Key/Value: Users_List_3/charlie
20231031-101554 - INIT    - DBG - Key/Value: Users_List_4/delta2
20231031-101554 - INIT    - DBG - Key/Value: Users_List_5/echo
20231031-101554 - INIT    - DBG - Key/Value: Users_List_6/foxtrot
20231031-101554 - INIT    - DBG - Key/Value: Users_List_7/golf
20231031-101554 - INIT    - DBG - Key/Value: Users_List_8/hotel
20231031-101554 - INIT    - DBG - Key/Value: Users_List_9/india
20231031-101554 - INIT    - DBG - Key/Value: Users_List_10/juliet
20231031-101554 - INIT    - DBG - Key/Last: Users_List_Last/juliet

^C
20231031-101837 - INIT    - MSG - testgui > Ctrl-C detected, finalize application
20231031-101837 - INIT    - MSG - Total execution time:  0h0 4m 40s
20231031-101837 - INIT    - MSG - Closing SQLITE database: testgui
```

12 TCP/IP

https://stackoverflow.com/questions/58361758/tcp-ip-using-ada-sockets-how-to-correctly-finish-a-packet

https://comp.lang.ada.narkive.com/V7LjubmR/tcp-ip-sockets-with-gnat-sockets

13 Unicode

13.1 Introduction

https://mcilloni.ovh/2023/07/23/unicode-is-hard

13.2 Glossary

Paragraphs are from theses Wikipedia links:
- https://en.wikipedia.org/wiki/Plane_[Unicode]
- https://en.wikipedia.org/wiki/Byte_order_mark
- https://en.wikipedia.org/wiki/Unicode
- https://en.wikipedia.org/wiki/UTF-8
- https://en.wikipedia.org/wiki/UTF-16

13.2.1 BMP

In the Unicode standard, a plane is a continuous group of 65,536 (216) code points. There are 17 planes, identified by the numbers 0 to 16, which corresponds with the

possible values 00–1016 of the first two positions in six position hexadecimal format (U+hhhhhh).

Plane 0 is the Basic Multilingual Plane (BMP), which contains most commonly used characters.

13.2.2 BOM

BOM stands for Byte Order Mark. The BOM character is, simply, the Unicode code-point U+FEFF ZERO WIDTH NO-BREAK SPACE, encoded in the current encoding.

The encoded representation of the BOM depends of the representation. A text beginning with the sequence EF BB BF suggests an UTF-8 encoding.

```
UTF-8          EF BB BF
UTF-16 (BE)    FE FF
UTF-16 (LE)    FF FE
```

A BOM can signal several things to a program reading the text:

- The byte order, or endianness, of the text stream in the cases of 16-bit and 32-bit encodings;

- The fact that the text stream's encoding is Unicode, to a high level of confidence;

- Which Unicode character encoding is used.

BOM use is optional. Its presence interferes with the use of UTF-8 by software that does not expect non-ASCII bytes at the start of a file but that could otherwise handle the text stream.

According to the Unicode standard, the BOM for UTF-8 files is not recommended.

13.2.3 Unicode

Unicode, formally The Unicode Standard is an information technology standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems.

Unicode can be stored using several different encodings, which translate the character codes into sequences of bytes. The Unicode Standard defines three encodings but several others exist, mostly variable-length encodings. The most common encodings are the ASCII-compatible UTF-8 and the ASCII-incompatible UTF-16, itself compatible with the obsolete UCS-2.

13.2.4 UTF-8

UTF-8 is a variable-length character encoding standard used for electronic communication. Defined by the Unicode Standard, the name is derived from Unicode (or Universal Coded Character Set) Transformation Format – 8-bit.

13.2.5 UTF-16

UTF-16 is a variable-length character encoding capable of encoding all 1,112,064 valid code points of Unicode (in fact this number of code points is dictated by the design of UTF-16). The encoding is variable-length, as code points are encoded with

one or two 16-bit code units. UTF-16 arose from an earlier obsolete fixed-width 16-bit encoding, now known as UCS-2 (for 2-byte Universal Character Set), once it became clear that more than 216 (65,536) code points were needed.

UTF-16 is used by systems such as the Microsoft Windows API, the Java programming language and JavaScript/ECMAScript. It is used by SMS (the SMS standard specifies UCS-2, but almost all users actually implement UTF-16 so that emojis work).

UTF-16 is the only web-encoding that is incompatible with ASCII and never gained popularity on the web. The Web Hypertext Application Technology Working Group (WHATWG) considers UTF-8 "the mandatory encoding for all [text]" and that for security reasons browser applications should not use UTF-16.

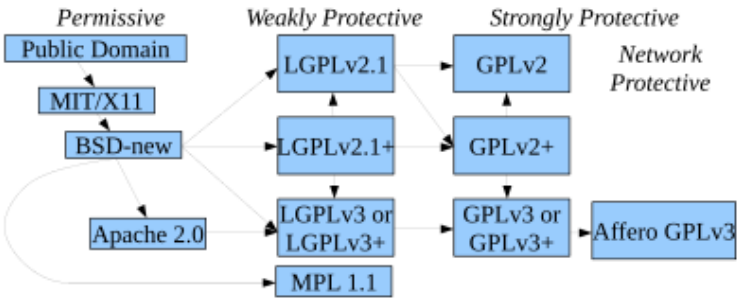
UTF-16 BE stands for Big Endian and UTF-16 LE for Little Endian.

Appendices

1 Copyrights & credits

1.1 Library Licence

v22 is copyright Sowebio under GPL v3 license.



1.1.1 GPL v3 compatibility with others licenses

https://en.wikipedia.org/wiki/License_compatibility: MIT license is compatible with GPL and can be re-licensed as GPL. European Union Public License [EUPL] is *explicitly compatible* with GPL v2 v3, OSL v2.1 v 3, CPL v1, EPL v1, CeCILL v2 v2.1, MPL v2, LGPL v2.1 v3, LiLIQ R R+ AGPL v3.

1.2 Manual license

1.3 v22 Packages copyrights, credits and licenses

Andreas Almroth: https://web.archive.org/web/20070403105909/http://www.almroth.com/adacurl [Curl binding, LGPL v2]

David Botton: https://www.linkedin.com/in/david-botton-3741b210 [Gnoga, LGPL v3]

Dmitry A. Kazakov: http://www.dmitry-kazakov.de/ada/components.htm [Simple Components, LGPLv2]

Jeffrey R. Carter: https://github.com/jrcarter/PragmARC [PragmAda Reusable Components, 3 Clause BSD]

Michael Rohan: https://sourceforge.net/projects/zanyblue [Internationalization, 3 Clause BSD]

Pascal Pignard: https://github.com/Blady-Com [UXStrings, CECILL-C v2.1]

2 To-do list

2.1 New features

2.1.1 v22.Api

20231102 - API OVH with provisions for other providers.

2.1.2 v22.Gui

20231102 - Add double clic to an item list to open record view.
20231102 - Add automatic keyboard short-cut to Module_Menu and User_Menu.
20231108 - The user list does not indicate whether it extends over several pages.

2.1.3 v22.Net

- Mail sending

  Sending mail currently depends on /usr/bin/sendemail, a particularly well thought-out perl script. See if you can use https://docs.adacore.com/gnatcoll-docs/email.html instead + a bit of code for the sending itself.

- SMS sending.

  Using OVH API or a local Teltonika 4G router.

2.1.4 v22.Pie

Print, import and export module.

2.1.5 v22.Tio

20241213 - Make Tio UTF-8 aware, usins Uxstrings.Text_IO.

20231102 - Add procedures Tio.Cursor_On and Cursor_Off using “tput civis” cursor invisible and “tput cnorm” cursor visible) or To hide the cursor: ESC + “?25l" and to To re-enable the cursor: ESC + “?25h" see https://gist.github.com/fnky/458719343aabd01cfb17a3a4f7296797

20231102 - Add functions “tput lines” and “tput cols” to get current console lines and columns values or the oneliner echo -e "lines\ncols"|tput -S or use https://stackoverflow.com/questions/27902721/ioctl-tiocgwinsz-in-gnat-ada-returns-errno-25-but-c-program-work-fine [should be better] and https://www.pegasoft.ca/resources/boblap/99_e.html

Tput overview : https://stackoverflow.com/questions/5947742/how-to-change-the-output-color-of-echo-in-linux/20983251#20983251

Add ANSI full color control including this work https://github.com/mosteo/ansi-ada https://en.wikipedia.org/wiki/ANSI_escape_code#CSI_sequences

20231102 - Add function [enter] or [quit]

20231102 - Add function [Yes] or [no] with Yes/No default choice

2.2 Other changes

<<<TODO>>>

2.3 Bugs correction

2.3.1 v22.Gui

20231108 - The 'Enter' key causes a disconnection when searching for in a CRUD.
20231108 - The 'Enter' key is not active when entering login + password.

2.4 Doc

2.4.1 The never-ending task

Hunt **<<<TODO>>>** tags :]

3 Coding recommendations

Pragmatic recommendations based on experience.

3.1 Use Ada style

Developing in Ada is the greatest proof of respect for the next reader of your code.

Ada code is timeless. There is no such thing as modern or ancient code in Ada. All the more reason to code for the next reader of your code.

For the next reader of your code, follow the recommendations of the book “Ada 95 Quality and Style”, available free of charge:

- Online: https://ada-lang.io/docs/style-guide/Ada_Style_Guide
- In PDF: https://www.adaic.org/resources/add_content/docs/95style/95style.pdf

3.2 Use SP constant instead of literal space in double quoted strings

Instead of:

```
Net.Command [Vpn_Host, "openssl x509 -outform der " &
   "-in " & Cert_Root_Path & "/pki/issued/" & Rut2m_Mac_Address & ".crt " &
   "-out " & Get_Tmp_Dir & Rut2m_Mac_Address & ".der"];
```

You should use:

```
Net.Command [Vpn_Host, "openssl x509 -outform der" & SP &
   "-in" & SP & Cert_Root_Path & "/pki/issued/" & Rut2m_Mac_Address & ".crt" & SP &
   "-out" & SP & Get_Tmp_Dir & Rut2m_Mac_Address & ".der"];
```

Rationale: Too much debugging time lost for one space too many or too few in system command concatenation strings, for example [but not only].

4 Quality control

Check list

<<< TODO>>>

5 Release check list

Things to do to release to Github.

<<< TODO>>>

6 History

6.1 Phase 1

2022 – 7 weeks: Théodore Gigault, first-year ENSEIRB intern.

```
v22.Gui: Gnoga study, extensive documentation and early developments, in-
cluding breadcrum.
```

6.2 Phase 2

2023 – 7 weeks: Arthur Le Floch, first-year ENSEIRB intern.

```
v22.Gui: Completion of the first release, as the web brick that justifies
the v22 framework in its own right.
```

6.3 Phase 3 - a.1

2023 – 4 weeks: Stéphane Rivière, Sowebio CTO.

```
v22: Porting the v20 library to the v22 framework.
v22.Uxs: Add UXStrings and delete v20.Vst.
v22.Sql: Removing Dmiti Kazakov's SQLite bind and replacing it with David
Botton's MySQL and SQLite bind and upgrading v22.Sql functions as the two
binds have a different approach. Debugging of the Gnoga.Database.Server
binding and work on v22.Sql to make the MySQL and SQLite interface per-
fectly normalized.
v22.Sql: This work shows that v22.Sql is far superior to v20.Sql. N data-
bases can now be opened simultaneously [interesting for gateways], inte-
grated data dictionaries are more elaborate, an exception automatically
closes all database connections, etc.
```

6.4 Phase 4 - a.2

2023 – 6 weeks: Stéphane Rivière, Sowebio CTO.

```
v22: Source width increased to 131 columns.
v22: Generalization of .adb chaptering with API & Private as in v22.Gui.
v22.Gui: Source reformatting: v22 standardization.
v22.Gui: To_UXStings replaced by v22.Uxs.To_String_Unsigned.
v22.Gui: Replacement of View, Base, Element, Common with package initials
for v22 naming standardization.
v22.Gui: Rename CRUD with Main_Menu.
```

```
v22: Parallel management of <application>.log and <application>-gnoga.log
with setters to control them.
v22: Automatic creation of configuration file <application>.cfg. handling
of parameters read.
v22: Correct management of the <application>.log file in the event of un-
expected termination (the current buffer is correctly emptied in the
file).
v22: Ctrl-C interception for orderly application termination (file clos-
ing, SQL disconnection, etc.).
v22.Gui: Add host and port parameters to the connection function.
v22.Gui: Online help, HTML file display.
v22.Gui  Rename Set_Text/Title to Put_Text/Title.
v22: Unification of v22 and Gnoga debug modes.
v22: Adding On_Off getters.
v22: General conversion of getters and setters from Boolean type (True/
False) to On_Off type (On/Off).
v22:  On_Off  type  added  for  Msg.Std,  Tio.Put,  Tio.Put_Line  and
Uxs.To_String
v22: Validation or inhibition of Ctrl-C.
v22.Gui: Installation of the free Overpass font (optimized for web read-
ing) in regular, bold, italic and bolditalic.
v22.Gui: Preparing for application theming.
v22.Gui: Harmonization of v22 resource directories (icons, images) and
Gnoga (use of Gnoga's standard /img directory).
v22.Gui: Main_Menu_Add_Element, add "event On_Click : GGB.Action_Event :=
null" parameter to extend Main_Menu caps.
v22.Gui: alpha reordering.
v22.Msg: Log procedures names refactoring.
v22: Change all Line procedures to New_Line.
v22: Handling of the default 8 hours connection timout associated to the
infamously unclear "MySQL Server Has Gone Away" error message with a task
pinging the MySQL server every hour.
v22.Gui: Main_Menu mode 1 (direct mode) implementation in addition to
mode 2 (sub menu mode).
v22.Gui: First (intermediate) version of user management, as a demonstra-
tion of a more elaborate future model (automatic generation from a data
dictionary).
v22.Uxs: Remove routines now implemented in UXStrings. v22 refactoring.
v22: Documentation update v34 to v66, 165 to 228 pages.
```

6.5 Phase 5 - v0.5

2024 - 8 weeks: Stéphane Rivière, Sowebio CTO.

```
v22.Gui: Enter key handling for login and new passwords screens.
v22.Gui: Lost password management with email notification and time lim-
ited temporary password.
v22.Prg: New, more portable password generation algorithm.
v22.Net: Sending mail function.
v22.Gui: Complete Gui.Dialog_Popup and make it compliant with the graphic
charter.
v22.Gui: Complete CSS refactoring and jQuery-UI theme creation process.
v22.Gui: Fix some CSS bugs. Add some displays enhancements.
v22.Gui.Crud: New package to ease CRUD screens management.
v22.Uxs: New functions to manage Money type with v22.Gui and v22.Sql
v22.Net: New Api (OVH management) and Send_SMS functions.

v22: Fix tons of bugs.
v22: Make tons of enhancements.
```

```
v22: Documentation update v66 to v120, 228 to 275 pages.
```

6.6 Phase 6 - v0.6

2025 - 2 weeks: Stéphane Rivière, Sowebio CTO.

```
v22. Uxf: New package to handle big UTF-8 data up to 1024 TB.

v22: Fix tons of bugs.
v22: Make tons of enhancements.
v22: Documentation update v66 to v150, 275 to 281 pages.
```

6.7 Phase 7 - v0.7

2025 - 4 weeks: Lounès Souakri, BTS SIO Merleau-Ponty Rochefort-sur-Mer intern.

```
v22. Net: New Api [Matomo management].
v22. Pdf: New package to handle PDF with high level functions.
v22. Prg: Add date function.

v22: Documentation update v66 to v167, 281 to 291 pages.
```

7 Issues

7.1 Update jQuery

In early 2024, the latest version of jQuery is 3.7.1. The Gnoga version is still 2.1.1, to preserve the alternating colors of the list lines. **The alternating rows colors are no longer displayed when jQuery is updated to version 3.**

Excerpt v22-colors.css:

```
    /* [31] List background and odd listed items */
    --content-list-color: #e6f4fa;

    /* [32] List header font color */
    --content-list-header-color: white;

    /* [33] List header background color */
    --content-list-header-color-background: #96b0b6;

    /* [34] List selected font color */
    --content-list-selected-color: white;

    /* [35] List selected background color */
    --content-list-selected-color-background: #66a8a8;

    /* [36] List even listed items */
    --content-list-secondary-color: #b5d0d7;
```

Excerpt v22-content.css:

```
.content-list {
    background-color: var(--content-list-color);
    width: 100%;
    border-radius: 8px;
    margin: 8px;
    padding: 8px;
```

```
    box-sizing: border-box;
    overflow-x: auto;
}

.content-list-header {
    font-weight: bold;
    text-align: center;
    color: var(--content-list-header-color);
    background-color: var(--content-list-header-color-background);
}

.content-list-item {
    text-align: center;
}

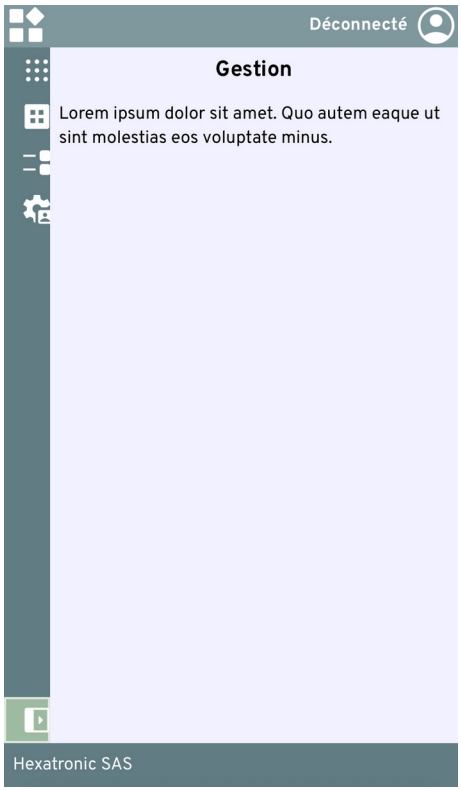
.content-list-table {
    width: 100%;
    border-collapse: separate;
    border-spacing: 4px 0;
}

.content-list-table > tbody > tr:not(:first-child):nth-child(odd) {
    background-color: var(--content-list-secondary-color);
```

Action: Find the problem to be able to use the latest jQuery version.

7.2 iPhone Safari web browser weirdness

On an iPhone 11 with Safari, the left margin of the main menu is shifted by a few pixels, off-centering the icons to the right. Attempts to adjust the CSS settings were to no avail:



Action: investigate with a more recent iPhone and/or Safari.

7.3 Avoiding Unrestricted_Access

The Don't use Unrestricted_Access FAQ tip recommends to not use Unrestricted_Access or keeping them at package level.

Action: none. To our knowledge, v22 already follows this recommendation.

Ada, « it's stronger than you ».
Tribute to Daniel Feneuille, a legendary french Ada teacher [and much more][7]

The link below is kept here for future use...

https://this-page-intentionally-left-blank.org

[7] http://d.feneuille.free.fr