

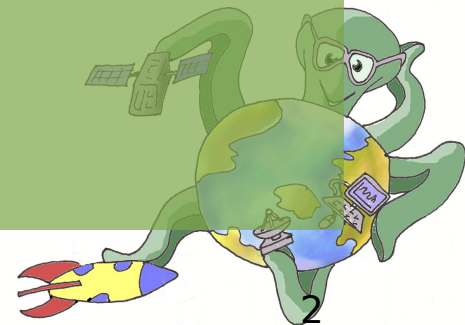


The TSP SDK by examples

Eric NOULARD
eric.noulard@onera.fr
with the sustained help of TSP Team Worldwide

TSP SDK: Documentation

- TSP comes with documentation
 - TSP Consumer/Provider Programming Tutorial
(http://download.savannah.nongnu.org/releases/tsp/tsp_programming_tutorial-1.0.pdf)
 - TSP Design & Programming Guide
(http://download.savannah.nongnu.org/releases/tsp/tsp_programming_guide-1.1.pdf)
 - TSP Group Explained
(http://cvs.savannah.nongnu.org/viewcvs/tsp_docs/tsp_progguide/tsp_group_explained.pdf?rev=1.1&root=tsp&view=log)
 - Doxygen generated documentation
(http://www.ts2p.org/tsp/API_doc/html/index.html)
 - Mailing List: tsp-devel@nongnu.org



TSP SDK: Multi-Language









- TSP in C
 - TSP cores libraries
 - Many applications
- TSP in Java : jtsp
- TSP in Ruby : rubytsp
- TSP in Perl : perltsp (SWIG based)
- TSP in Python : pytsp (SIWG based)

Savannah CVS Surfing - project [tsp](#)

[\[tsp\]](#)

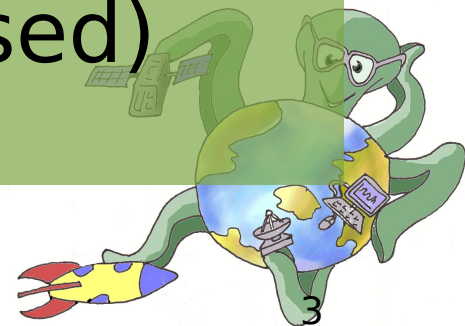
cvcs -d:pserver:anonymous@cvcs.sv.gnu.org:/sources/tsp co .
Files shown:0

[ViewCVS and CVS Help](#)

File ▾	Rev.	Age	Author	Last log entry
 dlock/				
 jtsp/				
 perltsp/				
 pytsp/				
 rubytsp/				
 tcltsp/				
 tsp/				
 tsp_docs/				

Show files using tag:

Send suggestions and report system problems to the [Savannah Hackers](#).



TSP in C : Configurable

- CMake configurable sources
(see TSP Programming Tutorial)

```
Page 1 of 1
BUILD_API_DOC OFF
BUILD_BB_KERNEL_MODULE OFF
BUILD_CONSUMER ON
BUILD_ONLY_TSP_PORT OFF
BUILD_PROVIDER ON
BUILD_SHARED ON
BUILD_XMLRPC ON
CMAKE_BACKWARDS_COMPATIBILITY 2.4
CMAKE_BUILD_TYPE
CMAKE_INSTALL_PREFIX /opt/tsp-0.8.3cvs
EXECUTABLE_OUTPUT_PATH
GTK_CONFIG_PROGRAM /usr/bin/gtk-config
LEX_PROGRAM /usr/bin/lex
LIBRARY_OUTPUT_PATH
ONCRPC_RPCGEN_EXECUTABLE /usr/bin/rpcgen
RPMTools_RPMBUILD_EXECUTABLE RPMTools_RPMBUILD_EXECUTABLE- NOTFOUND
USE_XDR_LIB_ENCODING OFF
XML2_CONFIG_PROGRAM /usr/bin/xml2-config
YACC_PROGRAM /usr/bin/yacc
```

Build part of TSP (red arrows pointing to BUILD_CONSUMER, BUILD_ONLY_TSP_PORT, BUILD_PROVIDER)

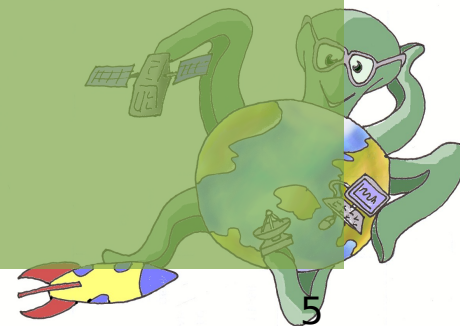
Shared or Static Build (green arrow pointing to BUILD_SHARED)

```
BUILD_API_DOC: Build doxygen documentation
Press [enter] to edit option
Press [c] to configure
Press [h] for help          Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

CMake Version 2.4 - patch 5

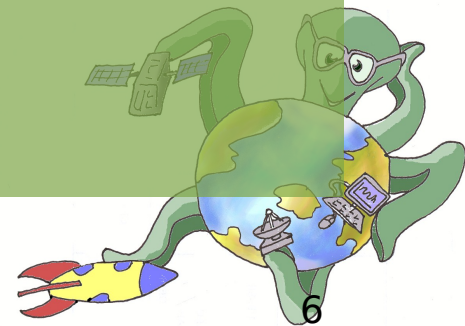
TSP in C : Applications

- Providers
 - tsp_stub_server
 - tsp_generic_reader
 - tsp_bb_provider
(see later Frederik DEWEERDT)
- Consumers
 - tsp_stdout_client
 - tsp_request_generic
 - tsp_ascii_writer
 - tsp_gdisp
 - targa



TSP Stub Server

- Test & Tutorial TSP Provider
- Pseudo Simulator that generates 1000 TSP symbols at 100Hz [not real time]
- May specify frequency (`-f arg`) [not real time either :)]
- May be used to test and validate your consumer application:
`tsp/src/providers/stub/*`



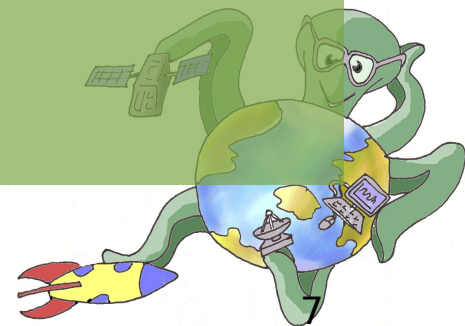
TSP Stdout Client

- Test & Tutorial TSP Consumer
- Subscribe to 's' first symbol(s) at period 'p' and display 'n' sample(s)

```
tsp_stdout_client -u rpc://localhost  
-p 10 -n 10 -s 2
```

- May be used to test and validate your provider application:

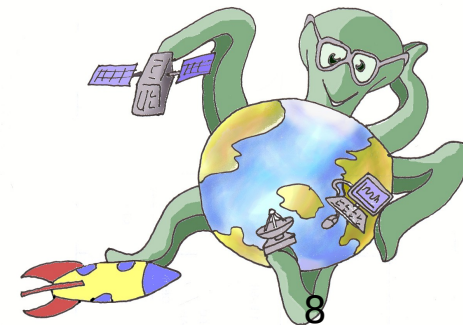
```
tsp/src/consumers/stdout/*
```



TSP Stub Server



Launch Stub & Stdout



TSP Request Generic

- TSP Test Consumer

`tsp/src/consumers/generic/*`

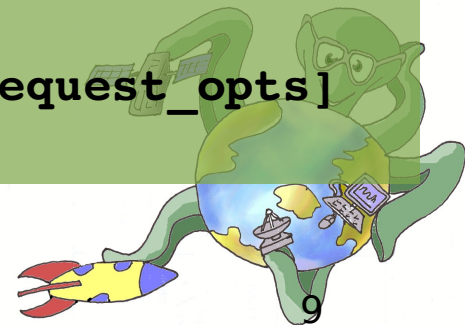
- may send any TSP request from the command line
- may be used to automate testing TSP itself (*coming soon*)

TSP generic consumer v0.8.3-rc1 (<https://savannah.nongnu.org/projects/tsp>)

TSP XDR encoding/decoding method is: `<TSP_NO_XDR_ENCODE>`

Usage:

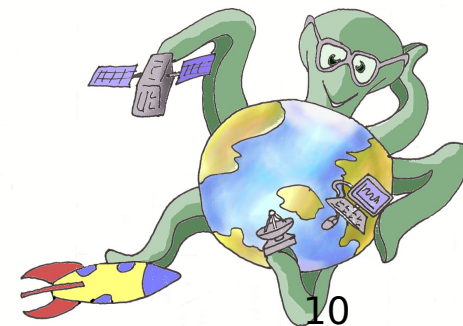
`tsp_request_generic [generic_opts] <tsp_request> [request_opts]`



TSP Request Generic

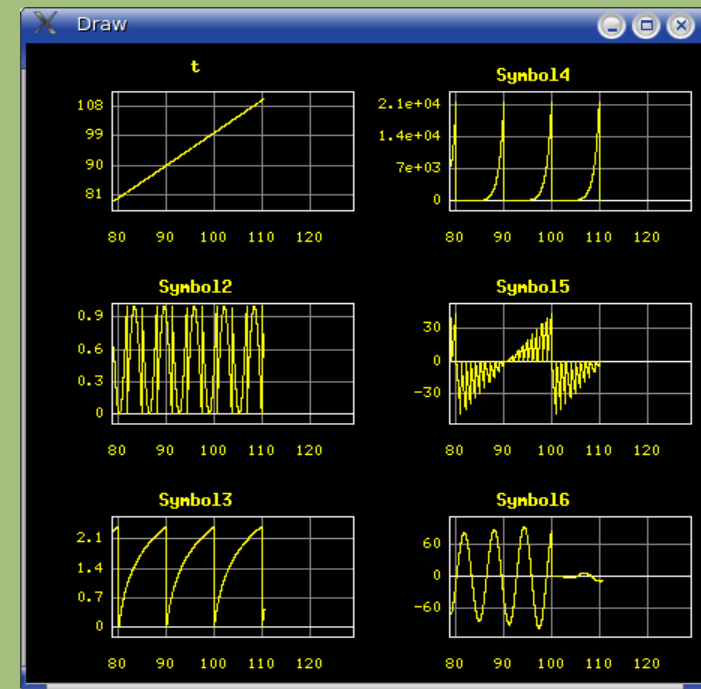
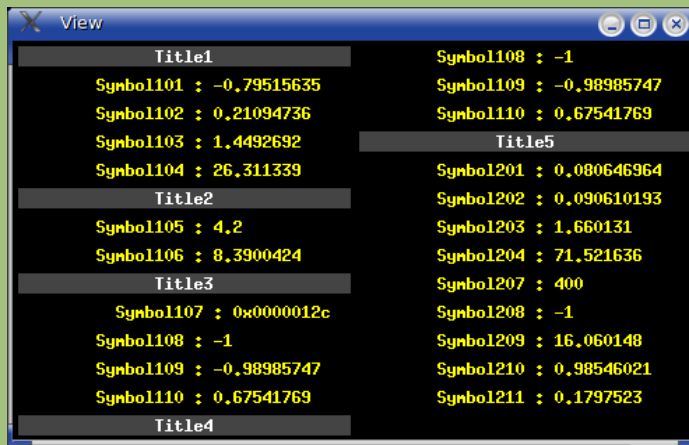


TRY Request Generic

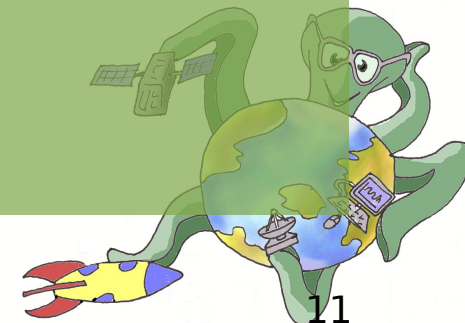


TSP GDisp Consumer

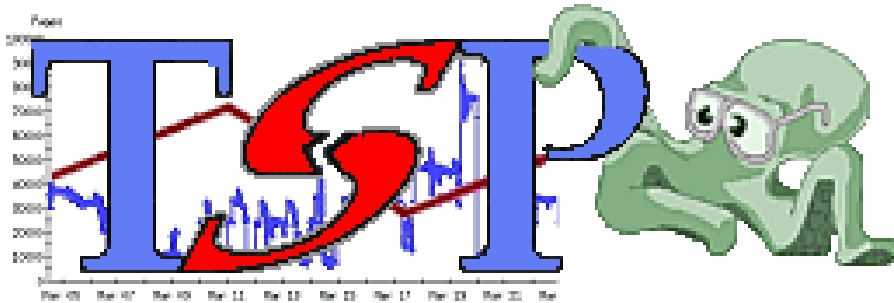
- Simple but very efficient graph plotter
- Simple (read-only) XML configuration file



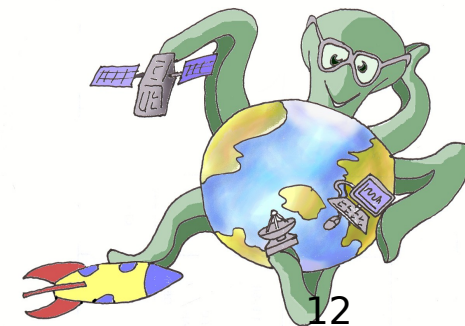
tsp/src/consumers/gdisp/*



TSP GDisp Consumer



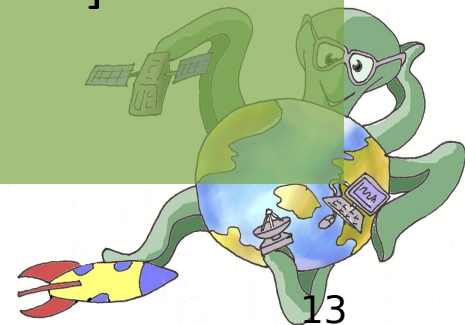
Launch GDisp



TSP ASCII Writer Consumer

- TSP ASCII Writer consumer
`tsp/src/consumers/ascii_writer/*`
 - simple configuration file (XML or ASCII)
 - generates ASCII tabular output
 - limited size output (circular file)
 - more generic than ASCII (see `-f arg`)

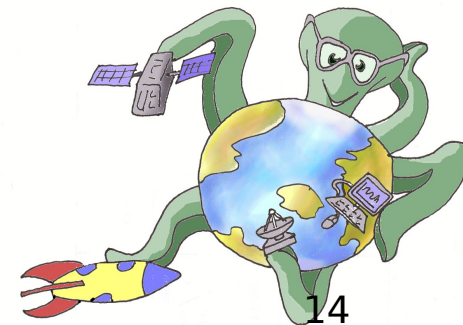
Usage: `tsp_ascii_writer [-n] -x=<sample_config_file>`
`[-o=<output_filename>] [-f=<output file format>]`
`[-l=<nb sample>] [-u=<TSP_URL>]`



TSP ASCII Writer Consumer

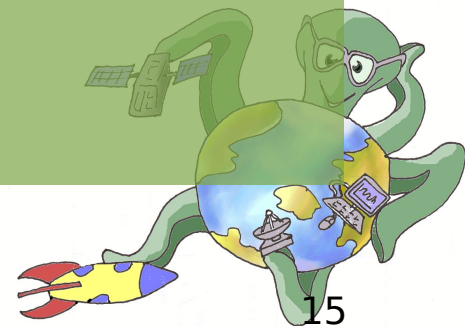


Launch ASCII Writer



TSP Generic Reader Consumer

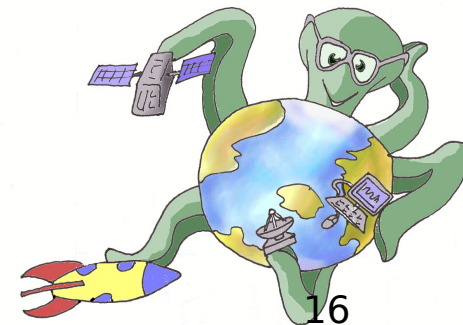
- Generic file reader provider
 - Reads TSP symbols values stream from file
 - « Passive » provider
(may wait for consumer)
 - Design for multi-format file handling
(Only MACSIM file format supported)
 - May replay ASCII Writer files



TSP Generic Reader Provider

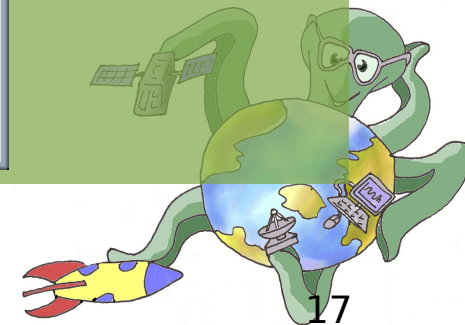


Launch Generic Reader



TARGA TSP Consumer

- A better gdisp
 - `tsp/src/consumers/gdisp+`
- Modular Graphical TSP Consumer
- Author: Stéphane GARAY
(asked to be excused)



Targa TSP Consumer



Launch Targa

