Bionic Buffalo Tech Note #112

Introduction to the Faroe Islands Test Tools

last revised Wednesday 2004.12.01

Introduction

The Faroe Islands Test Tools are a collection of programs to assist in the testing of distributed, networked applications and systems. The Tools are intended to ease the development and use of test and verification operations where individual tests involve more than one computer or user, where test data is gathered from more that one computer or user, and where test software at various network locations must be controlled centrally.

The Tools include these components:

- a test support subroutine library to incorporate into applications
- a test manager application to deploy, coordinate, and monitor running tests
- utilities to collate and report accumulated test data

Although applications may be run without incorporating the test support library, the library's API provides easy mechanisms to provide hooks which meet the conventions of the Faroe Islands Test Tools. An existing, unmodified standalone application usually may be tested or run with limited options, however.

The Tools are designed using CORBA. However, applications being tested need not be CORBA compliant. If an application is CORBA compliant, then it may use IIOP to communicate with the Faroe Islands objects. Otherwise, it may employ text based interfaces based on pipes, files, buffers, and command lines.

Test environments may vary widely. The Tools are provided in source code form for adaptation to a wide variety of systems. The Tools are free, and licensed under the FSF General Public License (GPL).

Typical Uses

Typical uses for the Faroe Islands Test Tools include the following:

- *Network protocol development* – the Tools can be installed on multiple network nodes to monitor operation, errors, data rates, and other statistics

- *Analysis of non-networked systems* – when a network is available in a test environment, but not normally used among the components of a system, the Tools can be used to report performance and other information not usually available
- *Failure simulation and analysis* – the Tools can observe the behaviour of system components under simulated network failure situations, where the normal (non-Tools) network channels become unavailable
- *Acquisition of performance and operational statistics* – the Tools can provide an easy mechanism to acquire and present data on the behaviour of systems which do not have built-in functions to gather such data
- *Scaffolding for distributed systems development* – the Tools include mechanisms to control programs remotely, and retrieve remote data, which can be used for prototyping of distributed systems before incorporated mechanisms are completed
- *Auditing other tools* – statistics gathered by the Tools can be compared against other data, verifying the accuracy of the information
- *Security analysis* – since they provide a back-channel, the Tools can be used to probe for certain vulnerabilities by attempting to circumvent normal controls

Application and System Test Environments

The basic division of a test environment is an *addressable process space* (or *APS*). Although each APS might be on a separate computer or network node, it is possible (and sometimes necessary) to partition a single node into multiple such spaces. (Security or architectural considerations might mandate this, especially when processes on a node cannot always communicate freely among themselves.)

Each APS hosts a *master process test object* (or *MPTO*), which must have a network address. Currently, the Faroe Islands Test Tools support internet protocols (based on UDP/IP and TCP/IP). If DNS or some other name resolution mechanism is available, then names instead of addresses may be used. Communication among APSs is through the MPTOs.

There are two ways to implement a MPTO: it may be embedded within a program by use of the test support library, or it may be a standalone program (such as the test manager application).

Within an APS, the MPTO controls tests; monitors operation; and manages, transmits, and receives data. The MPTO may start and stop standalone (non-Faroe Islands enabled) programs, or it may work through *slave process test objects* (or *SPTO*s) in programs built using the test support library.

Starting the Test Environment

Testing begins with a running test manager application. (The test manager application is provided with the Faroe Islands Test Tools.) The test manager initiates and controls the tested applications by comunicating with MPTOs in the various APSs. In order for this to happen, the MPTOs must be started or running already. In order for them to be running already, the test developer or user must start them manually, by shell script, or by some other mechanism external to the Tools. Otherwise, a test manager may attempt to start remote test managers itself using `ssh` or `telnet` protocols.

Alternatively, `ssh` or `telnet` may be used to start applications which include embedded MPTOs, or the developer or user may provide a proprietary launch mechanism.

Test Scripts

Process test objects (both MPTOs and SPTOs) are controlled by test scripts, which may be viewed as “test programs”. The test scripts allow starting and and stopping applications (processes and threads), starting other test scripts, passing data to applications, accepting data from applications, transmitting and receiving files, and conditional execution of the scripts depending on data values.

Test scripts are written in a subset of Perl, and can employ any features of the Perl language. However, there are certain presumed aspects of the environment, including preambles, postambles, and variables, which are used to interact with the rest of the test manager.

For machines which do not implement Perl, a small subset of the script language can be used. Subset scripts must be compiled; a compiler is included with the Tools.

Where Perl is available, it is possible to communicate interactively with the test environment. By running a program (started with the shell or otherwise), script commands can be entered at the keyboard and executed on-the-fly. This allows the *ad hoc* control of tests, data display, and setting of parameters.

Parallel Operation

Parallelism can exist within a test script, using the thread capabilities of Perl. In addition, multiple test scripts may run independently of each other. The only impediment to simultaneous execution of multiple scripts is the limitations of the software being tested. (For instance, if a program is not reentrant it cannot be run twice at the same time, unless in separate processes, in which case there might be contention for external resources such as files, semaphores, or memory.)

Each MPTO is a peer in the test topology. A test can be started at node alpha, interacting with node beta, and, at the same time, a different or the same test can be started at node beta, interacting with node alpha. Test scripts do not interact with one another unless explicitly asked to do so.

Embedding the Library into Applications

The test support library provides an API to testable applications, allowing interaction with the test environment and with process test objects (MPTOs and SPTOs). Functions include process and thread control; data preparation, transmission, and reception; and interaction with test scripts (rendezvous, mutexes, and other actions). The library also allows embedding the test objects into the application, where they exist as asynchronous and generally independent threads.

Data Acquisition and Management

The Faroe Islands Test Tools standardize the naming and organization of data, including files, records, fields, and aliases for nominal values. Each dataset, and the format in which it is held, are given a unique name in a hierarchical name space which allows the import and export of formats and datasets in a fashion which allows sharing of data, and prevents duplication of names.

Data accumulated during testing may be passed from APS to APS using protocols included with the tools, and exported from any APS using tagged flat-file, spreadsheet, HTML, or SQL formats. The tools themselves include basic mechanisms for data analysis, presentation, and reporting, but most such activities will require separate programming by the test developer or user.

Applications not using the test support library can pass data through memory buffers, files, or pipes to the test support software.

Exported data can be time stamped and may include traceability information such as the message digest of the controlling script and the process identifiers of participating applications, enabling audits of the test operation themselves.

Programming Language Support

The Faroe Islands Test Tools are written in C and in Perl. A subset of the scripting language, with some operational, semantic and syntactic limitations, can be implemented using only the C language.

Applications which can call C subroutines or Perl functions can access the test support library API. Other programs not so capable can still be tested when called as separate processes or applications.

A Java API is being considered for future development.