

# Testing

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

## **Contents**

---

Testing is an important part of quality assurance. Different levels of testing are required for different products. For example, exhaustive testing with code coverage close to 100% is required for the products which malfunction can be injurious to health or environment or can cause high material losses. However, such exhaustive testing is quite expensive and looks unreasonable in many cases. We can provide any quality of testing and can help you to choose and order required quality of testing. Jointly with other quality assurance services provided by the company (like Code Review) we can help you to achieve required quality of your product.

Our principles in the testing:

- automated testing;
- centralized control and logging in the case of distributed testing;
- easy-to-understand test cases;
- easy-to-use interface to run required test scenario (subset of complete test suite).

We can:

- test your product as white or black box;
- analyse for completeness an existing test specification and design missing test cases;
- design test specification in accordance with your requirements and implement it;
- estimate test coverage achieved by the created/existing test harness;
- train your engineers to use created test harness in the development process;
- train your engineers to use the framework to create new test cases;
- help you with fixing of found bugs.

We are the most experienced in the testing of the following areas:

- network protocols (ATM, Switching, STP, ARP, IPv4, IPv6, ICMPv4, ICMPv6, IGMP, UDP, TCP, DHCP, DNS, NAT, Firewall, Routing, etc);
- network management (SNMP agents, SNMP MIBs);
- API (Berkeley/Windows Socket API);
- file systems.

**Test Environment** maintained and actively developed by OKTET Labs employees is a software product which provides a powerful framework for testing of communication protocols, management and other services. In fact, TE application domain is restricted by TE architecture and by provided services only. We have already successfully used TE for testing of:

- SNMP agent MIB implementations;
  - HyperLAN/2 MAC-PHY C model;
  - DNS client/server;
  - DHCP client/relay/server;
  - IGMP;
  - iSCSI;
  - Ethernet switch (STP, ARL, VLAN, management, etc);
-

- IP routing;
- Socket API;
- Linux services (SSH, FTP, SMTP, DNS server, DHCP server, X, VNC, etc);
- ATM;
- 802.1X and RADIUS for WiFi Access Point.

The list of successful stories of TE usage permanently grows. Each successful story adds support of corresponding protocols in TE and, therefore, minimizes efforts for further projects which require such features.

Services provided by TE define its usefulness and application domain:

- Control of testing process actors (called Test Agents) which may be used to retrieve configuration (e.g. IP addresses, statically configured routes, supported SNMP protocol versions) and state (e.g. dynamic ARP entries, network statistics, etc.) information, generate and receive network traffic (e.g. ICMP messages, SNMP requests/replies), call API functions remotely (e.g. Socket API), etc.
- Centralised time-ordered multilevel structured log collected from all entities which take part in testing process. Filtering of logs at compile time, run time and report generation stage.
- Unified interface to any kind of configuration and state information, centralised control and roll-back of configuration changes by request and at the end of testing session.
- Execution of complex test scenarios. Structured test cases representation, iteration of test parameters, random and simultaneous tests run, test sanity control, running or excluding from run tests targeted/used some feature.

If you don't find answers to all your questions, please, feel free to contact us.

---