

Spirent iTest

Overview

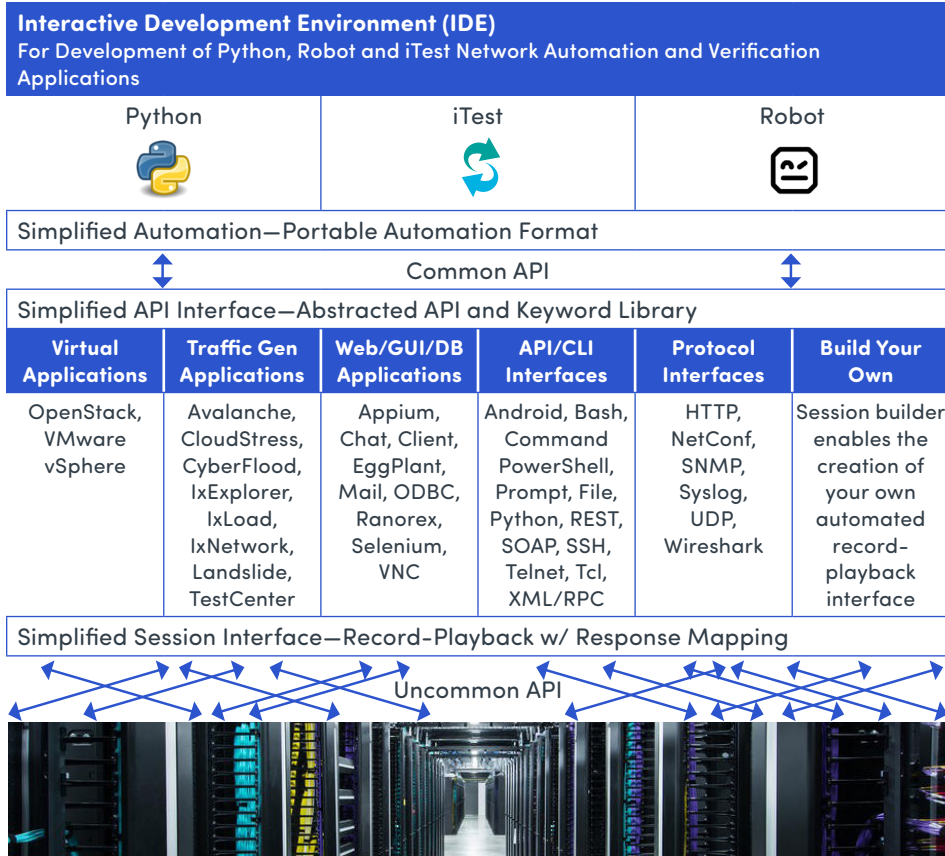
Spirent iTest was created to simplify the complexities presented by heterogeneous networks by allowing development and operations teams to rapidly create portable automation within an agnostic network automation development framework.

Spirent iTest delivers the first ever unified Interactive Development Environment for Python, Robot and iTest users. This release provides the optimal environment for creating modern network automation and verification applications. Our innovative Portable Automation Format (PAF) simplifies automation and adoption by enabling developers to leverage their automated applications for both lab and production use. Spirent iTest highly productive Record-Playback and patented Response Mapping are available within Python and Robot development environments.

Business Benefits

- Single solution for both lab and production environments: automate, verify and seamlessly deploy from lab to production
- Speed up time to market using reusable automation libraries for Python and Robot
- Improve efficiency with built-in DevOps workflows and sharing of environments

iTest IDE is designed to abstract the complexity of networks from the ground up.

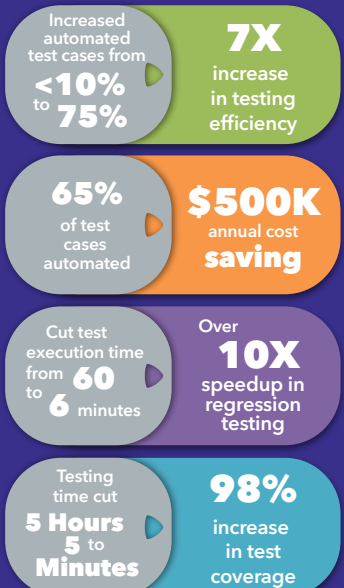


Heterogenous Hybrid (Physical and Virtual) Networks

Highlights

- Python-based platform for automation developers and network engineers
- Abstract the complexity of hybrid networks to create robust Python and Robot automation and verification applications
- Easily convert manual network configurations and testing workflows into secure distributable automation applications, API libraries and keywords
- Integrated system analysis w/ customizable reports for actionable verification
- Portable Automation Format accelerating deployments from lab to production

Testimonials



Spirent iTest IDE

Python-based Interactive Development Environment (IDE) for automation developers and network engineers.

Key IDE Features

Python-Based: Rapidly create, reuse and publish Python and Robot Framework automation content from a unified development environment.

Live Interaction: Real-time development and troubleshooting of network and test automation with any device, application, or system.

Record and Playback: Create automation scripts by capturing every action during a manual test and replaying the captured steps.

The screenshot displays the Spirent iTest IDE interface. The top menu bar includes File, Edit, Navigate, Search, Project, Run, iTest, Window, and Help. The left sidebar shows the iTest Explorer with a tree view of project files. The main workspace is divided into several panels:

- Steps:** A table showing test steps with columns for Action, Session, and Description. The first step is 'open' with session 's1' and description 'project://di_Ubuntu/session_profiles/ssh_1.ffsp'. The second step is 'getEthernetNic' with session 's1' and description '-intf eth0'. The third step is 'analyze' with session 's1' and description 'Speed() value == "10000Mb/s"'. The fourth step is 'close' with session 's1'.
- Response:** A panel showing the response of the selected step. It displays settings for 'eth0' and a list of supported link modes.
- Console:** A panel showing the output of the test execution, including a table of queries and matches.

Action	Session	Description
open	s1	project://di_Ubuntu/session_profiles/ssh_1.ffsp
getEthernetNic	s1	-intf eth0
analyze	s1	Speed() value == "10000Mb/s"
close	s1	

Query	Matches	Value	Location	XPath
isEmpty()	1	false		//isEmpty
responseLine()	1	Settings for eth0:		//responseLine
definedIn()	1	project://di_Ubuntu/session_profiles/ssh_1.ffsp		//definedIn
Supported_ports()	1	[TP]	line 1, cols 18:24	//Supported_ports
Supported_link_modes()	1	10000baseT/Full	line 2, cols 12:22	//Supported_link_modes
Supported_pause_frame_use()	1	No	line 4, cols 28:30	//Supported_pause_frame_use
Advertised_link_modes()	1	Not reported	line 6, cols 25:37	//Advertised_link_modes
Advertised_pause_frame_use()	1	No	line 7, cols 29:31	//Advertised_pause_frame_use
Speed()	1	10000Mb/s	line 9, cols 8:17	//Speed
Duplex()	1	Full	line 10, cols 9:12	//Duplex

Portable Automation Format: Develop and export automated tests for use in any environment including lab, staging and production networks.

Response Mapping: This patented feature automatically parses complex device messages to extract the key information.

Leverage existing automation: Enhance the value of your existing Python, Robot, Bash and PowerShell content by importing them via 'File -> Import'

The screenshot shows the 'Export' dialog box in the Spirent iTest IDE. It has two sections:

- Source and Destination:** A section with a 'Select source test case and destination location' label. It includes a text field with the path 'project://my_project/test_cases/unity.fftc' and a 'Browse...' button.
- Choose location of python script to be generated:** A section with a text field showing the path 'project://my_project/python_scripts/unity.py' and a 'Browse...' button.

 Below the dialog box, a snippet of Python code is visible, showing a function 'main' that interacts with a device and logs the results.

```

param = Params()

def main(slc, logger, status):
    procedure_result = {}
    di_Ubuntu = slc.open('di_Ubuntu')
    s1 = di_Ubuntu.ssh_1.ffsp.open(properties={'ipaddress': 'cw-vel.spirenteng.com', 'password': 'spirent'})
    response = s1.getEthernetNic(intf='eth0')
    handle_step_results(response, status, logger)

    if response.result == 'success':
        extracted = response.query('Speed()')
        if not isinstance(extracted, (list, tuple)):
            extracted = [extracted]
        for value in extracted:
            if value == "10000Mb/s":
                logger.info('Value \'{value}\'' is consistent with the condition'.format(value=status.pass_test_if_not_already_failed(log=logger)
            else:
                logger.error('Value \'{value}\'' is not consistent with the condition'.format(

```

Spirent iTest Portable Automation Format

Network automation and verification applications are easily created and securely distributed for both lab to production use.



Network Automation and Verification IDE

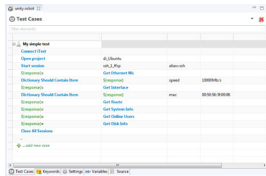


w/ Velocity LaaS/TaaS



Full capability of iTest applications, API libraries, keywords and results integrated with Python and Robot IDEs

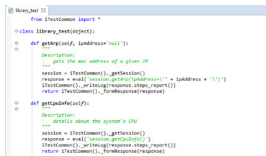
Results Analysis



- Apply complex analysis rules with Boolean logic
- Customizable reports with system integrated

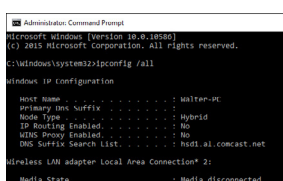
PORTABLE

Keyword Libraries



- Build higher level, abstracted automation and test libraries
- Publish as Python, Robot or RESTful APIs

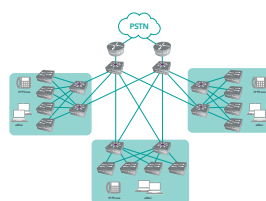
Session Record-Playback



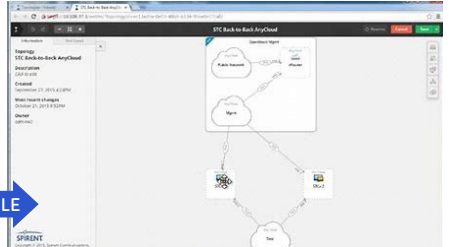
- Abstract devices and applications with common session interface
- Capture and replay user actions
- Troubleshoot issues easily: extensive logs available, including from RESTful sessions

PORTABLE

Network Topologies



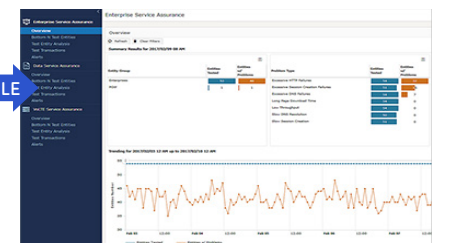
- Visually model and integrate network or test environments
- Abstract the make, model, and version specifics from the devices being automated



- Create and instantiate development and test environments instantly
- Deploy, schedule, run and share automation
- Store and report all automation results



w/ VisionWorks Service Assurance



- Publish automation from lab into production environments
- Test sequences for validating and monitoring network devices and services

Portable Automation Format

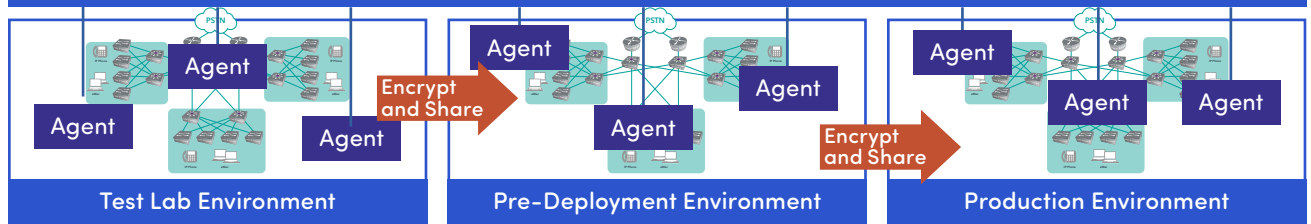
Network Automation and Verification Applications

Integrated Python, Robot and iTest Editors



Secure Agent Framework Environment

Distributable Secure Automation: Run only signed apps, encrypt all config inputs-outputs, stream encrypted data



Collaborative Network DevOps Workflow Environment



AUTOMATION DEVELOPERS



NETWORK ENGINEERS

