SOAtest and Virtualize 9.10.0 with CTP 3.0.0

Parasoft provides an environment-based approach to Continuous Testing—integrating service virtualization, API testing, and test data management with test environment management to enable early, rapid, and rigorous testing of continuously-evolving, distributed systems.

In this release

Overview | New Features | Migration Notes | Resolved Issues

Overview

The latest Continuous Testing release (Continuous Testing Platform 3.0, Virtualize 9.10, and SOAtest 9.10) features the industry's most powerful and comprehensive thin client interface for service virtualization and API testing. From any browser, a broad range of team members can rapidly create, leverage, and share API testing and service virtualization assets to "shift left" testing. These assets can then be seamlessly extended and reused across the extended Continuous Testing ecosystem to support more specialized and sophisticated use cases as well as to enable advanced automation within CI/CD pipelines. Environments correlate the resulting service virtualization assets, test assets, test data, and provisioning actions within an intuitive interface so that even novice users can instantly provision the exact test environment needed and start testing immediately.

The new release also introduces advanced traffic processing capabilities, Burp Suite (security penetration testing) integration for API and web functional tests, HTTP/2 support, Fiddler support, and a SOAtest Jenkins plugin. The solution is now available on the Microsoft Azure Marketplace and VSTS Marketplace.

New Features

Extended Continuous Testing Ecosystem Enhances Collaboration and Automation

With Parasoft Virtualize and Parasoft SOAtest connected to the new Continuous Testing Platform thin-client interface, team members can do the following directly from their browser:

- Visualize the dependencies associated with an AUT and specific test environments
- Record interactions with dependent components and create virtual assets that simulate this behavior
- Create tests and virtual assets from API definitions (Swagger, RAML, WSDL, etc.)
- Configure individual test environment components—including virtual asset performance, test data, and response logic
- Correlate specific virtual asset, test, and test data configurations with the context of a test environment
- Instantly deploy and destroy “disposable” test environments
- Reconfigure a test environment's real and simulated components for different testing needs
- Define and execute test jobs that correlate test execution to specific environment states, enabling complete traceability and easy, accurate defect reproduction
- Identify test environment issues before they compromise the accuracy of test results
- Manage, manipulate, and modify test data from Parasoft Data Repository

From Virtualize and SOAtest desktop, users can immediately access the artifacts created in CTP and extend them for more advanced needs—for example, using APM data to drive virtual asset performance, extending API tests into multi-channel test scenarios that also include web-layer, database, and mobile application tests, or extending the existing assets with the 120+ supported protocols and message types. These extended assets are seamlessly updated and available for use across the desktop, thin client, or RESTful API.

Parasoft CTP can also be deployed in the cloud (e.g., via containers and AWS or Microsoft Azure). You can use automation to instantly provision complete test environments, as well as centrally reconfigure the performance, data, and behavior characteristics of all the environment's virtual assets for different automated test scenarios and for different stages of the delivery pipeline. Using the provided build tasks, the exact environment needed for a given testing need is instantly spun up when required, then destroyed as soon as it's no longer needed. Docker images are provided to facilitate setup. Moreover, the Parasoft solution is also available on the Microsoft Azure Marketplace and VSTS Marketplace—with "pay-as-you-go" or "bring-your-own license" models.

Additional automation opportunities are available through CTP's API for testing, service virtualization, test environment management, and test data management tasks.

UI Redesign to Facilitate Access to the Platform's Expanded Feature Set

Over the past years, the product originally introduced as Parasoft Environment Manager has extended beyond "test environment management," adding functionality to create, manage, and share service virtualization assets, test assets, and test data—and to tie it all together in the context of environments. With this release, we redesigned the UI to provide fast, easy access to the rich set of features that the CTP thin client provides when connected to Parasoft Virtualize and Parasoft SOAtest.

Thin-Client Workspace Widgets Provide Quick Access to Common Tasks and Real-Time Status Data

Customizable CTP "workspaces" feature widgets that provide instant access to the artifacts and real-time status data most important to each user. For example, a workspace could provide at-a-glance access to server status, recently-used environments and test/service virtualization artifacts, errors and risks that should be reviewed, and quick links to common tasks (such as recording traffic for virtual asset or test creation, virtualizing a REST endpoint, testing a REST endpoint, and managing test data). Every team member can create their own workspace tailored to their particular needs and preferences.

Advanced Traffic Processing Engine Available in Thin Client Interface

The advanced traffic processing engine (available in CTP when connected to Virtualize 9.10) enables users of the thin client interface to take advantage of the advanced traffic processing capabilities which were previously introduced in Virtualize desktop. This enables more team members to rapidly create virtual assets with sophisticated correlations and relationships between the request and response pairs captured in traffic. As with all virtual assets created within the Parasoft solution, these artifacts can be instantly accessed and customized from the Virtualize desktop as well as the thin client. Changes made from either the thin client or the desktop are instantly available to all team members, no matter what interface they are using.

Burp Suite Integration Provides API and Web Security Penetration Testing

Integration with Burp Suite, the application security testing tool that is fast becoming recognized as the industry standard, brings a new level of API and web security penetration testing to the Parasoft solution. Burp Suite analyzes the HTTP data that SOAtest processes during functional testing (including browser tests and API calls to SOAP or RESTful services). It attempts to expose security vulnerabilities in that data, then creates a report with vulnerability impact, severity, certainty, background, and remediation details. It also reports findings back to SOAtest, which then describes those vulnerabilities in the context of functional tests. As a result, you not only know what vulnerabilities exist within your application, but you also understand which use cases are impacted. In addition to Burp Suite and SOAtest reporting, these findings will also be sent to Parasoft DTP, which provides complete visibility into your entire testing process from static analysis, to unit testing, to functional testing, all the way to security penetration testing. Burp Suite integration is available via a Marketplace artifact for SOAtest.

HTTP/2 Support for Testing and Service Virtualization

Expanding upon its industry-leading message/protocol support, Parasoft's solution now supports testing and simulation of HTTP/2. Whether teams are adopting HTTP/2 or working with dependencies using it, they can continue to use their Parasoft products and test/service virtualization assets as they test their AUT. HTTP/2 testing is supported via a Marketplace plugin for SOAtest; HTTP/2 service virtualization is built in to the Virtualize server.

HTTP Archive (HAR) Support for Creating Tests and Virtual Assets from Fiddler Traffic Files

This Marketplace artifact for SOAtest and Virtualize converts HTTP Archive (HAR) files into traffic files that can be used for creating Parasoft virtual assets or test scenarios. This enables automated asset generation from traffic captured by Fiddler, Charles Web Proxy, and other programs that generate HAR files.

Jenkins Plugin for Parasoft SOAtest

If you're using Jenkins to execute SOAtest tests during Continuous Integration, the new Jenkins plugin lets you integrate these test results directly into your Jenkins job. With the plugin, teams can not only view SOAtest results within Jenkins, but also use the results of these tests to control Jenkins workflows. For example, a SOAtest test failure could stop a build from progressing to the next build phase. This plugin is available as a Marketplace artifact.

Migration Notes

The following information will help you migrate to CTP 3.0.0.

JDBC Driver for Recording Database Behavior

The Parasoft JDBC Driver (for recording live database behavior so it can be simulated) was updated in 9.10. JDBC Drivers released prior to Virtualize 9.10 will not able to communicate with the 9.10 Virtualize Server. If you were using an older JDBC driver, be sure to replace it with the new driver from the 9.10 installation. The new driver will be able to communicate with older versions of Virtualize.

Traffic File Format Change

Virtualize 9.10 introduces a new traffic format to accommodate enhancements in the traffic processing engine. Traffic files created in Virtualize 9.10 cannot be used in earlier versions.

Service Enabled License

A Service Enabled license is now required for:

- Connecting the Virtualize server to CTP
- Managing Virtualize server from external Virtualize desktops
- Accessing the REST API of the Virtualize server
- Starting the Virtualize server in headless mode

Built on Eclipse 4.6 (Neon)

Parasoft Virtualize and Parasoft SOAtest desktop (standalone version) are now built on Eclipse 4.6, which ships with Java 8.

JavaScript Engine Changes

SOAtest/Virtualize versions prior to 9.10 shipped with Java 7, which included the Mozilla Rhino JavaScript engine. SOAtest/Virtualize 9.10 ships with Java 8, which uses the Oracle Nashorn JavaScript engine as the default. Scripts saved in earlier versions of SOAtest/Virtualize with the Mozilla Rhino engine will now run in the Oracle Nashorn engine with Mozilla compatibility extensions loaded. The scripts will continue to function as before, but with significantly improved runtime performance. For additional details, see the SOAtest/Virtualize user's guide.

JDBC ODBC Driver Changes

The recommended best practice for connecting to databases via ODBC in SOAtest/Virtualize is to use vendor-specific ODBC drivers. SOAtest/Virtualize versions prior to 9.10 shipped with Java 7, which included a JDBC-ODBC driver in the associated JDK. This is deprecated in Java 8, so it is no longer available with SOAtest/Virtualize 9.10. Any DB Tools or data sources which were configured to use the previous JDBC ODBC driver will need to be reconfigured to use a different driver.

Resolved Issues

The following issues were resolved in this release.

Parasoft CTP/Environment Manager

PR/FR ID	Description
118091	TDM: Support for streaming extremely large data repositories
120838	Environment Manager: Creating/editing environment instances is delayed
121067	API Testing: tst files are no longer visible in Workspace after 401 status error

Parasoft Virtualize

PR/FR ID	Description
106986	Support Traffic files with gzipped content to create tst & pva
116398	Updating from 9.9.0 to 9.9.1 results in switch install directory of product
118885	Unexpected behavior of PVA with message forwarder
119731	Virtual Asset Performance Profiles not carried over to remote servers
119855	Traffic File creates corrupted .PVA and fails to create Repository
120481	Error trying to parse an empty field as a number
120834	Incorrect URL Path correlation generated when PVA from traffic is created outside Virtual Asset folder
120896	Virtualize to EM not registering SSL port
120917	Error when using a Proxy with a Service URL that contains escape characters
121094	Inconsistent encoded/decoded characters for path parameters when creating PVA from Traffic

Parasoft SOAtest

PR/FR ID	Description
90119	Saving problem with Test Variables and Test Flow Logic
108887	Allow specification of Content-Type header when HTTP method is not POST or PUT
117442	REST API returns HTML report that contains garbled Japanese characters
118528	Data source partitions not respected for data sources that read data from external sources
118887	Write File and FileStreamWriter REST APIs should allow specifying target directory that does not yet exist
119356	32-bit Windows installation throws a Java Virtual Machine Launcher Error

119364	Error thrown when custom tool is run with a disabled output tool
119638	UTF-8 datamodel read using platform default encoding
119704	Configure for Load Test adds '=' at the end of payloads whose content type is unrecognized
119741	Datasource is loaded into memory multiple times in one .tst
119748	RabbitMQ Transport "replyTo" Header Issue
119849	Unable to import environment when variable is marked as masked but is not encoded
120118	Document that Text Data Bank only operates on MIME Multipart Responses that are text
120290	"Content-type" Header is not allowed with GET in REST client tool
120335	Parameterized variables are not recognized in Reference Tests when in Literal "File" mode
120361	Generated XPath is incorrect when Asserting all members of an element sequence when xmlns is not SOAP-ENV
120464	Input to Extension tool does not contain output when chained to Event Monitor
120934	Unexpected Header Format when running a REST Client using a MIME Multipart payload
121154	Load Test: File dialog issue with Mac OS El Capitan
121184	"Url wrong format!" error when saving SQL data source
121279	Data Source rows show empty row 0 in detailed Load Test report