

# CTP Overview

In this section:

- [Introduction](#)
- [Service Virtualization](#)
- [API Testing](#)
- [Test Environment Management](#)
- [Marketplace](#)

## Introduction

Parasoft Continuous Testing Platform (CTP) is browser-based interface for leveraging Parasoft's API testing, service virtualization, and test environment management functionality across your organization. CTP enables you to create virtual assets for AUT dependencies that are not ready or available for testing, then immediately start defining and executing end-to-end tests against a combination of the real and virtual assets. CTP also acts as a central repository for the necessary configuration details of the dependent systems associated with the application under test.

## Service Virtualization

Parasoft Service Virtualization helps development and QA teams access the complete, realistic test environments needed to develop or test an application—anytime, anywhere. Testing is often delayed or cut short because it's difficult to access the complete, realistic test environments required to test meaningful transactions. By replacing unavailable dependencies with easily accessible and configurable "virtual assets", service virtualization allows teams to start testing earlier, test more completely, and test continuously.

- Capture and simulate live system behavior from recording
- Rapidly model incomplete/unavailable components from service definitions and logs
- Easily configure complex test conditions (e.g., "what-if," security, concurrency, fail-over, performance, and negative test scenarios)
- Simulate any system interactions—supports REST & web services, plus an industry-leading 120+ protocols/message types
- Get the exact test environment you need, on-demand

## API Testing

You can use CTP to add, review, extend, and reorganize SOAtest .tsts, test suites, and tools.

From CTP's Test Scenario management page, you can add, configure, and manage test scenarios and test clients for REST and SOAP APIs (powered by SOAtest). The test configuration interface is designed to help new users rapidly define a core set of tests. In addition, users can review, modify, and extend the team's existing library of test assets from SOAtest.

There are three main ways that your library of test scenarios can be used in CTP:

- **Automated provisioning during testing:** You can configure a test job that you want to execute one or more times, then associate an environment context with that test scenario so that the appropriate environment is always provisioned before that test job is run. For example, a test scenario can use one set of test data and endpoint variables for execution in a development testing environment and another in a system integration testing environment. The job execution history stores the associated test environment settings and variables along with results, enabling complete traceability.
- **Automated testing during provisioning:** This is essentially the flip side of the above use case. Test scenarios can be automatically executed when a specific test environment is provisioned. For instance, you can configure functional test scenarios to execute whenever your functional testing environment is provisioned—or your security test scenarios to execute whenever your security test environment is provisioned.
- **Customized health checks:** Your test scenarios can be used to assess the health of a specific component (a real system component such as a third-party service, mainframe, database, etc.—or a virtual asset that simulates the behavior of an actual system component). For example, you could run tests to validate that the virtual asset is behaving in the same way as the actual asset whose behavior it is simulating. These tests can be associated with any component type.

## Test Environment Management

Parasoft Environment Manager integrates and extends the above API testing and service virtualization capabilities via an intuitive web-based interface for creating, managing, sharing, and executing test environments (including service virtual assets) and tests. Environment Manager unites test environment management (test environment setup, visualization, validation, and provisioning), service virtualization (virtual asset creation, configuration, and deployment), and API testing (test creation, management, and execution).

- Visualize the dependencies associated with an AUT and specific test environments
- Easily configure all test environment components—including virtual asset performance, test data, and response logic
- Instantly provision "disposable" test environments with zero impact to the organization
- Identify test environment issues before they compromise the accuracy of your test results
- Assess the impact of a change to any specific component

## Test Data Management

Centralized creation and management of secure test data that can be applied across all solutions and integrated tools (including open source tools), as well as across team roles and test types (API/service, integration, performance, security...)

- Capture request and response traffic and reuse the data for subsequent scenarios.
- Import data from multiple sources and leverage it across test and service virtualization artifacts.
- Ensure that the data can be reused and shared across multiple teams, projects, versions, and releases
- Visualize, restructure, and generate data for your tests (additional license required)

## Marketplace

The Marketplace application provides an interface for sharing service virtualization and API testing extensions across your organization.