Core Virtualize Terms and Concepts

This topic introduces core terms and concepts that are essential for understanding and working with Parasoft Virtualize. It covers:

- System Scope
    - Application Under Test (AUT)
    - Dependent Component
- Core Parasoft Virtualize Components
    - Virtual Asset
    - Message Proxy
    - Parasoft JDBC Driver
    - Virtualize Desktop
    - Virtualize Server
    - Continuous Testing Platform
- Virtual Assets
    - Responders
    - Data Source
    - Data Groups
    - Performance Profiles
    - Responder Suite
    - .pva File
    - Project
- Other
    - Data Repository
    - Event Monitoring
    - Queue Browser
    - Change Advisor
    - Recording Proxies (not be confused with message proxies)
    - .pva Environments (not be confused with CTP environments)
    - Provisioning Action
    - Architecture Overview

System Scope

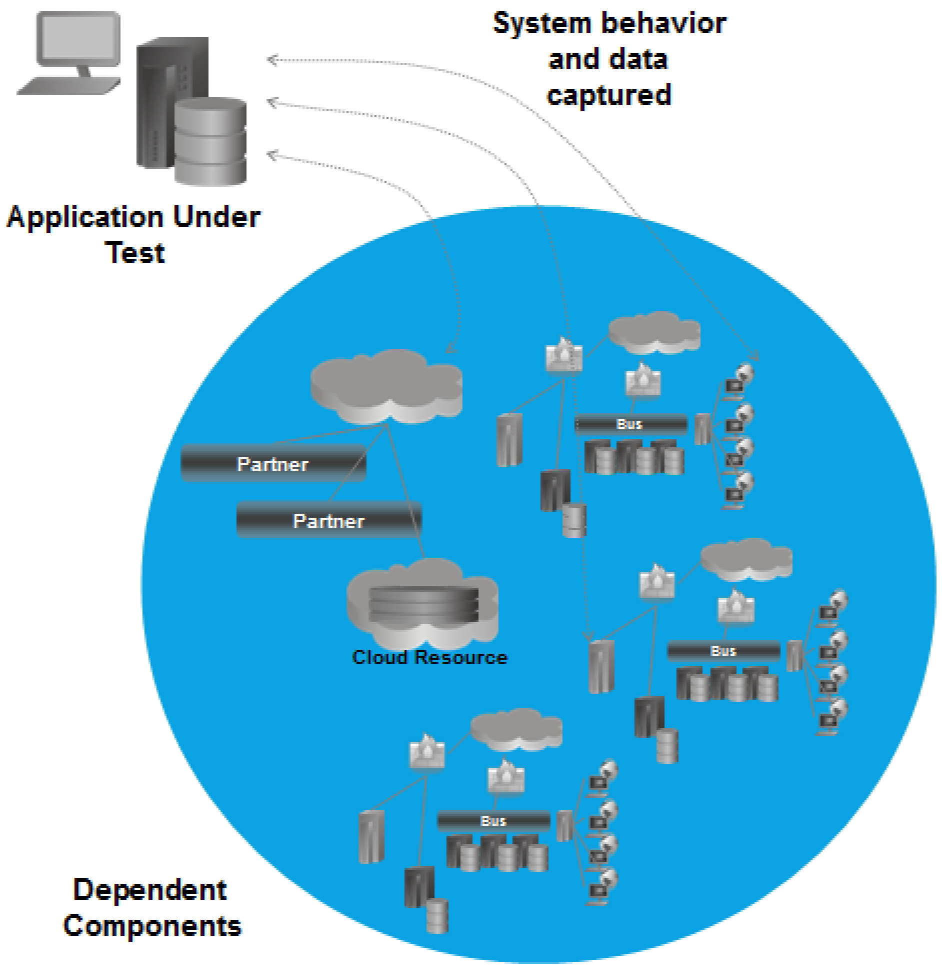
Application Under Test (AUT)

The part of the system that you are responsible for developing or testing. This is sometimes referred to as the *system under test* or an *in-scope system*. You and your team have direct control over (and immediate access to) the AUT.

Dependent Component

An existing application or component within a development, test, or production environment that the AUT needs to access in order for you to complete your development or testing tasks. This is sometimes referred to as an *out-of-scope system*. Dependent components might include mainframes, third-party applications or services, databases, etc.

These components might be difficult to access for development and testing purposes. Or, they might be difficult to configure for your test scenarios. When you need faster, more flexible access to behavior associated with these components (including data and performance profiles), you can "virtualize" this behavior with a Parasoft Virtualize "virtual asset."



Core Parasoft Virtualize Components

Virtual Asset

Virtual assets simulate the behavior, performance, and data of dependent components.They can be used for manual or automated testing with any testing platform. Virtual assets for existing components can be created from live recording, definition files, or log files. In addition, virtual assets can be modeled from sample messages or designed from scratch.

For details, see Working with Virtual Assets.

Message Proxy

Operates as a man-in-the-middle between an AUT and a dependent component. The proxy can record traffic that passes through it so that the corresponding behavior can be virtualized. It can also be used to control whether traffic is routed to a virtual asset or to the actual component. Once the AUT is configured to communicate with the proxy, you do not need to reconfigure the AUT in order to enable/disable recording, reroute traffic to the virtual asset or actual asset, etc.—you can achieve this by simply modifying the proxy configuration.

For details, see Working with Message Proxies.

Parasoft JDBC Driver

Similar to a message proxy, but designed to operate on database queries and results. Once the Parasoft JDBC Driver is configured and the AUT is set up to communicate with it, you can use the JDBC Controller (in the Virtualize Server view) or Parasoft Environment Manager to switch modes (passthrough, record, virtualize, etc.)—without having to restart the application server.

For details, see Using the Parasoft JDBC.

Virtualize Desktop

The interface for traffic capture, creation / modeling / deployment of virtual assets, and remote management of Parasoft Virtualize Servers.

For details, see Exploring the Virtualize UI.

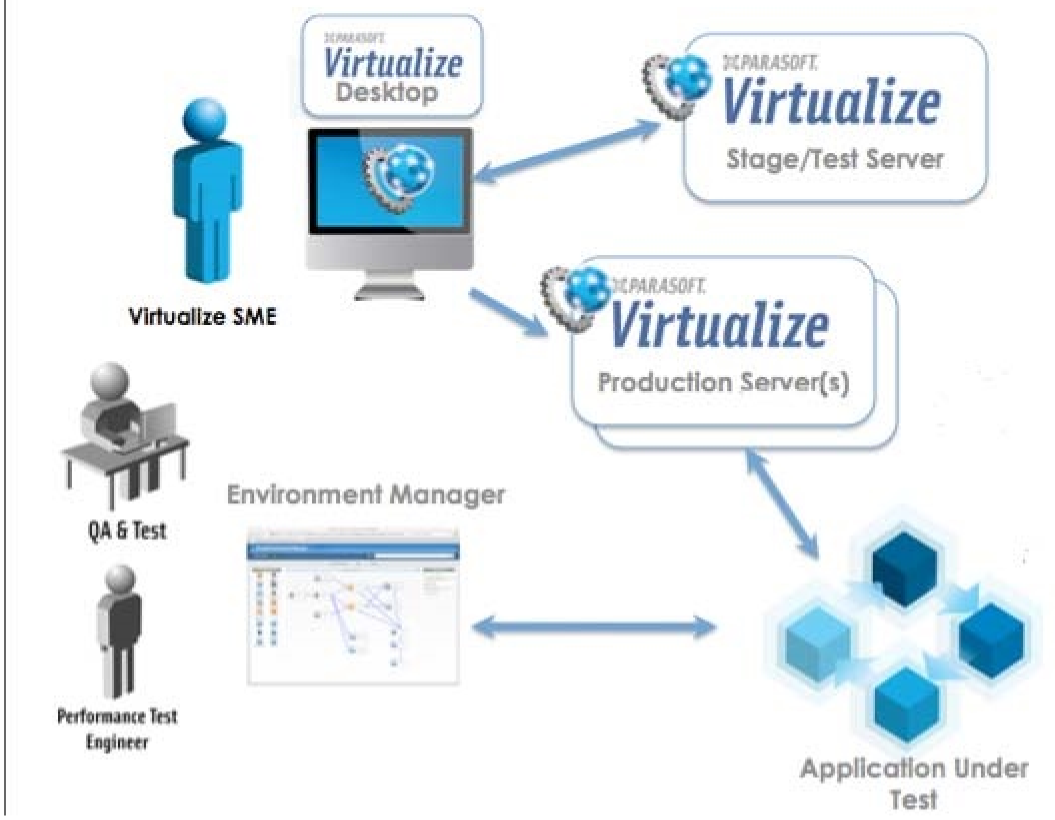
Virtualize Server

The Virtualize installation (with a server license) where virtual assets, proxies, and provisioning actions are deployed and hosted. Virtualize Servers can be controlled via Virtualize Desktop.

The recommended workflow is to first deploy a newly-created virtual asset to a "staging" remote server in order to validate that it works as expected and to fine-tune its behavior. Then, once the virtual asset is operating properly, you can move it to a "production" Virtualize server for centralized, team-wide access.

Note that Virtualize Server requires a server license.

For details, see Deploying Virtual Assets.


PARASOFT Virtualize Desktop
PARASOFT Virtualize Stage/Test Server
Virtualize SME
PARASOFT Virtualize Production Server(s)
QA & Test
Environment Manager
Performance Test Engineer
Application Under Test

Continuous Testing Platform

Parasoft Continuous Testing Platform (CTP) provides a web interface for developers and testers to select and access virtual assets in the context of test environments. Team members can review and provision pre-configured test environments that can include different combinations of real and virtual assets (set to different states with various performance profiles, data sets, etc.). The Virtualize Administrator can decide what environments are available to different users and what configurations and options each environment provides

Note that CTP is installed separately from Parasoft Virtualize.

Virtual Assets

Responders

Responders are tools that specify which response should be sent for a given incoming request. Each responder responds to incoming request messages that match its correlation criteria (described in Understanding the Message Correlation Process). Responses can be configured in a variety of modes, ranging from simple fixed messages to dynamic parameterized messages using a data source. Multiple responders can be configured, each with its own correlation strategy, data set, and message structure. Message Responders are protocol agnostic; the transport protocol or API to access a responder is defined in the deployment configuration of the associated virtual asset. Message Responders match incoming messages to responses; SQL responders match SQL queries to ResultSets.

For details, see Working with Virtual Assets.

Data Source

A data source is any set of data you want to use to populate response messages or other tool values. Data sources can include CSV, Excel, relational databases, files on disk, in-project spreadsheets, and Parasoft's data repository.

For details, see Parameterizing Tools with Data Source Values, Variables, and Extracted Values.

Data Groups

Data groups are a special type of data source. With data groups, you can group similar sets of data (such as development environment test data and load /performance test data), then easily switch which data set is used at any given time—without having to edit the tool configurations or data sources. To use data groups, you group together data sources with common columns. You can then specify "on the fly" which data source should be applied to the virtual asset at the given time. Data groups are configured in Virtualize desktop.

Performance Profiles

Performance profiles give development, QA, or performance engineers the flexibility to rapidly configure dependent system performance characteristics. This enables an array of performance testing scenarios to be executed one after another. The performance of each virtual asset can be set to reflect the dependent application's actual performance or to emulate specific performance models that you want to test against. Performance profiles are configured in Virtualize desktop.

For details, see Working with Performance Profiles.

Responder Suite

Responder suites are used to group and organize responders, variables applicable to multiple responders, data sources, performance profiles, and other assets within a .pva file.

For details, see Adding Projects, Virtual Assets, and Responder Suites.

.pva File

A .pva file can contain one or more nested responder suites. A .pva that is deployed on a Virtualize Server is considered a virtual asset.

For details, see Adding Projects, Virtual Assets, and Responder Suites.

Project

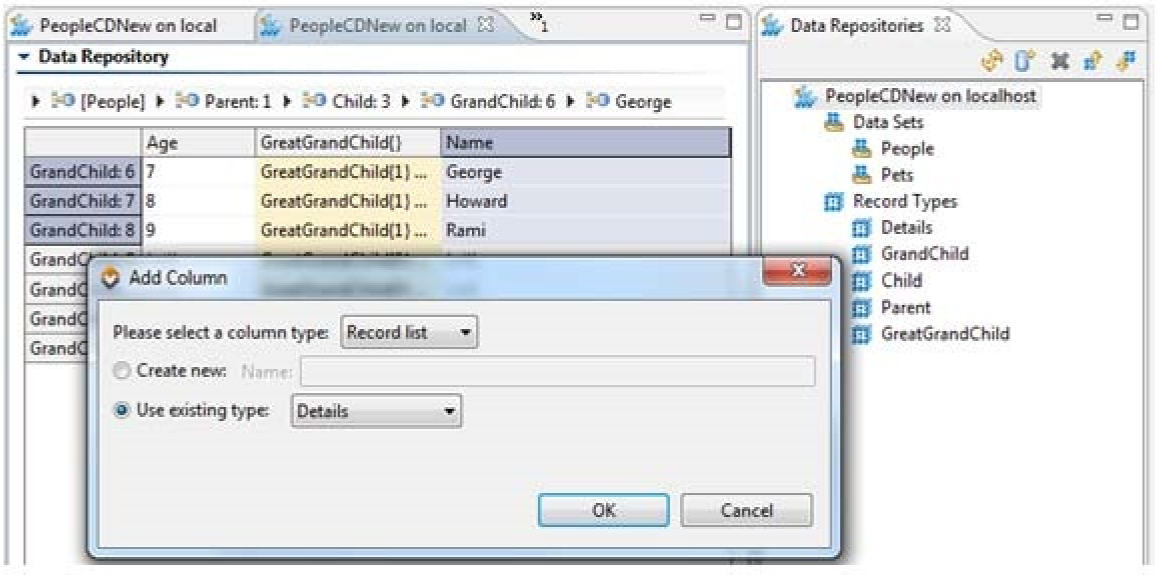
A project can contain any number of Virtualize-specific .pva files. It can also contain source files you want to use with Virtualize (e.g., Excel data sources), and any other resources that make sense for your environment.

For details, see Adding Projects, Virtual Assets, and Responder Suites.

Other

Data Repository

Parasoft Data Repository helps teams define, extend, and review large and/or hierarchical data sets for use in Parasoft messaging tools. Once a repository is established on a data repository server, it can be populated from existing data sources and/or updated manually. Through the graphical representation of hierarchical data, you can review and extend the repository structure and contents. Records from one data set can be referenced in other data sets to simplify editing and management of large data sets with a high level of data reuse.



	Age	GreatGrandChild()	Name
GrandChild: 6	7	GreatGrandChild(1) ...	George
GrandChild: 7	8	GreatGrandChild(1) ...	Howard
GrandChild: 8	9	GreatGrandChild(1) ...	Rami

Please select a column type: Record list
( ) Create new: Name:
(●) Use existing type: Details

For details, see Working with Large Hierarchical Data Sets.

Event Monitoring

Provides insight into Virtualize Server events (request messages received, response messages sent, errors, and so forth). This helps you monitor the behavior of the AUT and diagnose unexpected behavior as it occurs and better understand long-running transactions.

For details, see Gaining Visibility into Server Events.

Queue Browser

Allows you to connect to a JMS broker to retrieve the list of queues provided by that broker and see the messages currently sitting on each queue. The Queue Browser also lets you delete messages off the queues.

For details, see Browsing Queues.

Change Advisor

To keep pace with rapidly-evolving services and environment conditions, it's important to have a fast, easy, and accurate way to update virtual assets. Change Advisor enables you to assess the impact of changes to existing virtual assets and then quickly update existing assets (or create new ones) in response to the identified change impacts. This vastly reduces the amount of time required to modify assets as services and environment conditions change.

For details, see Change Management.

Recording Proxies (not be confused with message proxies)

If you do not want to deploy message proxies, recording proxies allow you to record traffic "on the fly." Like message proxies, these proxies can concurrently capture live traffic that passes through multiple endpoints.

See, details, see Recording Traffic on the Fly.

.pva Environments (not be confused with CTP environments)

An environment is a collection of variables that can be referenced in your Virtualize Responder suite. Using environments, you can define variables for endpoints, connection properties such as usernames and passwords, database table names, etc... then indicate which variable values should be applied in a given environment. By using environment variables instead of specifying" hard-coded" values in your responders and associated tools, you can instantly "flip a switch" and reconfigure them for use in a different environment (e.g., development, testing, UAT, pre-production, etc.).

For details, see Configuring Virtualize Environments.

Provisioning Action

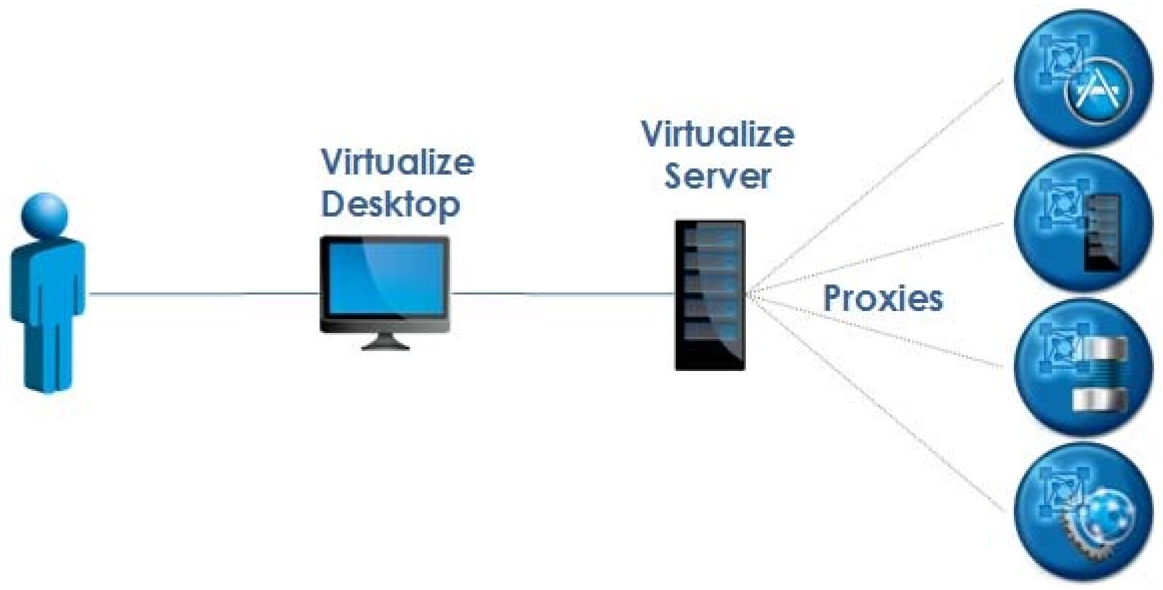
Provisioning actions can be used to automate configurations that are commonly applied in your environments. For example, you might run a script for changing certain environment settings, then update a settings file via FTP. Or, you might initialize a live database for testing.

Provisioning actions are defined in Provisioning Action (.pvn) files, which contain action suites. The structure and management of .pvn files and action suites is similar to that of .pva files and responder suites (respectively); each project can contain multiple .pvn/.pva files, each of those can contain any number of action suites or responder suites, data sources can be added and used to parameterize tools, tools can be added as outputs, and so on.

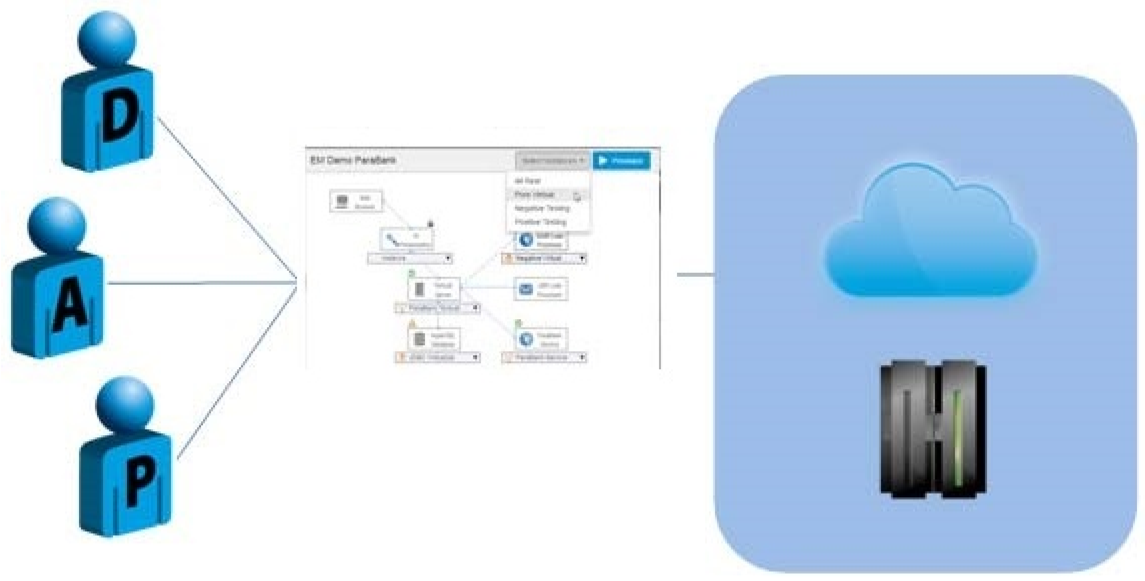
For details, see Defining Provisioning Actions.

Architecture Overview

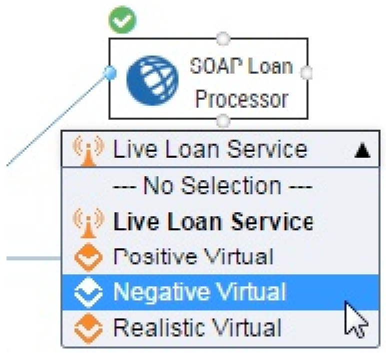
The team members responsible for creating, updating, and managing virtual assets, proxies, and related assets (e.g., data sets) use Virtualize Desktop to perform these activities.



Developers, testers, and performance test engineers can use CTP's graphical web UI to review and configure dev/test environments with dependent components set to the states needed for particular scenarios.



For example, a functional tester might want to test vs. an environment with the live database initialized for testing as well as the mainframe and 3rd party services represented by virtual assets configured for negative testing. At the same time, a performance test engineer might want to run one scenario vs. all dependent components simulating peak traffic conditions, then run another scenario vs. an environment that simulates average performance conditions.


SOAP Loan Processor
Live Loan Service
--- No Selection ---
Live Loan Service
Positive Virtual
Negative Virtual
Realistic Virtual