

# OpenText Service Virtualization

Quickly create realistic simulations of APIs and virtual services to speed application delivery

## Benefits

- Accelerate time-to-market by speeding up release cycles
- Increase efficiencies by eliminating bottlenecks
- Shorten test and reproduction cycles
- Improve overall product quality with fewer defects

Composite application initiatives often create a dilemma for development and testing teams: do they meet the project deadline OR deliver a high-quality service? The core issue is growing interdependency among functional components driving reliance on resources from third-party vendors, which leads to waiting and extra cycles.

The OpenText™ Service Virtualization solution enables application teams to easily create virtual services that can replace targeted services in a composite application or multi-step business process. By accurately simulating the behavior of the actual component, it enables developers and testers to begin performing functional or performance testing right away, in parallel. This is possible even when real services are not available, when data access is restricted, when data is difficult to attain, or when services are not suitable for the test.

OpenText Service Virtualization also saves you money by eliminating the requirement for access to constrained business-critical infrastructure, third-party systems, or pay-per-use cloud components for testing.

Major impediments addressed by OpenText Service Virtualization include:

- Non-availability of production and third-party systems for development, functional, and performance testing causing delays.
- Increased complexity in building and maintaining testing environments with high maintenance costs.
- Inability of developers to quickly identify and replicate the root cause of failure and deliver fixes in a timely manner.

The result of using OpenText Service Virtualization is not only the ability to “shift-left” and execute tests earlier in the delivery cycle, but also to focus on service quality attributes, such as performance, reliability, and scalability. In short, OpenText Service Virtualization delivers both faster delivery times and higher quality services. It also creates significant benefits for all participants in the application delivery process.

Feature	Description
<b>User friendly</b>	<p>An intuitive design IDE includes data-oriented modeling independent of the user's technical background and skills.</p> <p>Users can quickly model access to dependent application components and shared services, expose unfinished components to testing teams and other projects for dev/test, and eliminate the need to create and maintain programming stubs.</p>
<b>Broad protocol and simulation coverage</b>	<p>A range of pre-built industry-wide protocols are enabled for out-of-the-box simulation. Advanced simulation capabilities, like hybrid simulation and performance batch processing, cover even the most complicated testing scenarios.</p>
<b>Developer-focused simulation with extended mobile and IoT capabilities</b>	<p>Solve advanced simulation use cases in testing of enterprise applications spanning web and mobile user interfaces, from legacy back-ends to cloud-native applications, connected devices, and Internet of Things.</p> <p>Dev testers can use simulation language, simulation of publish-subscribe pattern, and simulation models as invocation scenarios, streams of data, or API tests.</p>
<b>One solution for unit, functional, and performance testing</b>	<p>Pre-built integrations to the OpenText™ functional testing solutions and OpenText™ performance engineering solutions allow you to easily provision and control virtual services directly from automation tools and collect metrics during test execution and simulation time.</p> <p>Stand up working test environments faster and with lower costs. Conduct more realistic tests by modeling back-end functional, performance, and network behavior.</p>
<b>Scalable and secured simulation infrastructure</b>	<p>Flexible simulation infrastructure is capable of handling large numbers of concurrent simulations while delivering thousands of transactions per second.</p> <p>Virtual services are deployed in OpenText Service Virtualization Server nodes serving multiple virtual services and controlled over API, command line interface, or web-based portal. Access, which is secured by authentication and virtual services, can be restricted using Access Control Lists (ACLs).</p>
<b>Web-based management</b>	<p>Web-based OpenText Service Virtualization Management Interface brings visibility and control to virtual services across multiple server nodes.</p> <p>It allows provisioning and control of virtual environments, management and configuration of protocol agents, parameterized search and filtering, as well as access to virtual service and server statistics, event audit, logged messages, and simulation reports.</p>
<b>Flexible deployment</b>	<p>Deploy multiple OpenText Service Virtualization Servers depending on performance requirements, test environments, system architecture, or organizational needs.</p> <p>Users can choose either to use direct OS installation or containerized deployment with pre-created Docker® images.</p>

Feature	Description
<b>Mobile testing with simulation</b>	<p>Easily create virtual services that can replace targeted services in a composite application or multi-step business process.</p> <p>The OpenText Service Virtualization Lab is deployed together with OpenText™ Functional Testing Lab for Mobile and Web, allowing simulation of REST API and communication to physical devices over NFC and Bluetooth services that Application Under Test (AUT) consumes.</p>
<b>Test asset management</b>	<p>Pre-built integrations to Application Lifecycle Management (ALM) and Source Code Management (SCM), allow storage and management of OpenText Service Virtualization projects as test resources in ALM or together with testing assets in SCM.</p> <p>Shared OpenText Service Virtualization asset management allows for easier maintenance, versioning, and facilitates re-use of the virtual services by other users and other testing tools.</p>
<b>DevOps and continuous integration</b>	<p>Integration with continuous integration (CI) tools using OpenText Service Virtualization management API and CI plugins such as Jenkins® or Bamboo allows OpenText Service Virtualization to be part of the continuous integration process.</p> <p>When combined with test automation, these capabilities enable enhanced workflows for developers and testers through shortened feedback cycles in continuous integration, continuous testing, and DevOps practices.</p>

## Key software components and licensing

OpenText Service Virtualization Software consists of the following applications:

### OpenText Service Virtualization Designer

A client application enabling creation of virtual services and running simulations of real service behavior. It allows design and validation of virtual services within the same desktop environment and includes an embedded server for hosting virtual services.

### OpenText Service Virtualization Server

A standalone server application that hosts virtual services. Optimized for performance, it can contain many more services than the OpenText Service Virtualization Designer and can be accessed by multiple designers.

### OpenText Service Virtualization Lab Server

A standalone application that can run a subset of protocols and only virtual services defined by simulation language—in context of API, mobile, and IoT testing.

### OpenText Service Virtualization Management Interface

A web application enabling management of virtual services and configuration on OpenText Service Virtualization Servers, without opening the Designer or individual projects. Installed by default when you install the OpenText Service Virtualization Server.

## Associated products

- OpenText Core Performance Engineering
- OpenText Professional Performance Engineering
- OpenText Enterprise Performance Engineering
- OpenText Performance Engineering for Developers
- OpenText Core Software Delivery Platform
- OpenText Functional Testing
- OpenText Functional Testing Lab for Mobile and Web
- OpenText Software Delivery Management
- OpenText Application Quality Management

## Flexible licensing

OpenText Service Virtualization offers two editions and two OpenText Service Virtualization Server licensing options to accommodate almost any organizational need.

### 1. Express edition

- OpenText Service Virtualization Designer: 3 VS, 10 TPS (free)

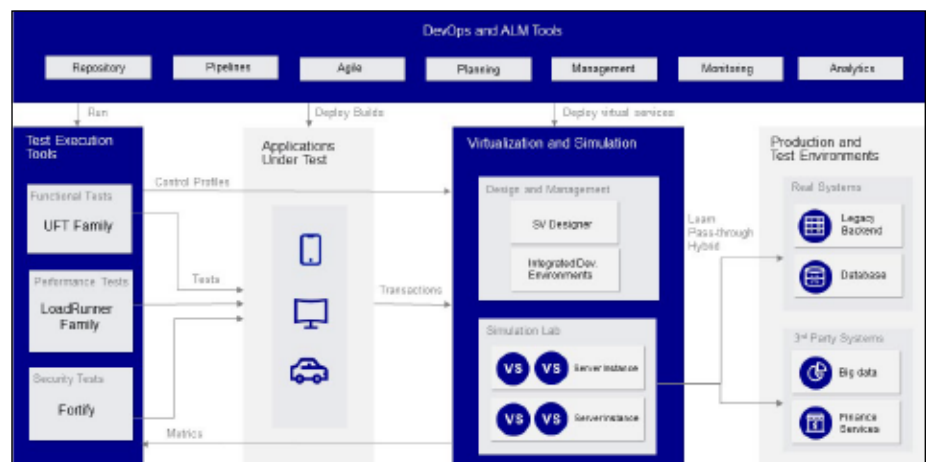
### 2. Enterprise edition

- OpenText Service Virtualization Designer: seat, unlimited VSs, 10 TPS
- OpenText Service Virtualization Server:
  - Static: server instance, unlimited VSs
  - Dynamic: VSs in simulation, unlimited OpenText Service Virtualization Server nodes, 3 VS types
- Includes limited capacity of OpenText Service Virtualization Lab virtual services

## System requirements

Supported Operating Systems:

- Windows: Windows 8.1, 10, 11, Windows Server 2012/R2, 2016, 2019, 2022
- Linux® (OpenText Service Virtualization Server only): Red Hat® Enterprise Linux 7.0–7.3, 8.0–8.5, Oracle® Linux 7.3, 8.2 and CentOS 7, 8.2
- Database: MS SQL 2008, 2012, 2014, 2016, 2019, 2022, Azure SQL, Oracle 11g/ 12c, PostgreSQL 9.x/12.x
- OpenText Service Virtualization Lab Server: OpenJDK 8u112, 8u361, Oracle® Java 1.8.111
- Containers: Docker® (Windows and Linux), VMware
- OpenText Service Virtualization Configurator: Java 11



OpenText Service Virtualization as part of the application delivery ecosystem integrated to ALM, functional, and performance testing tools



**Connect with MFGS, Inc.,**  
the exclusive master supplier  
of OpenText (legacy Micro  
Focus) products to the DOD  
and IC.



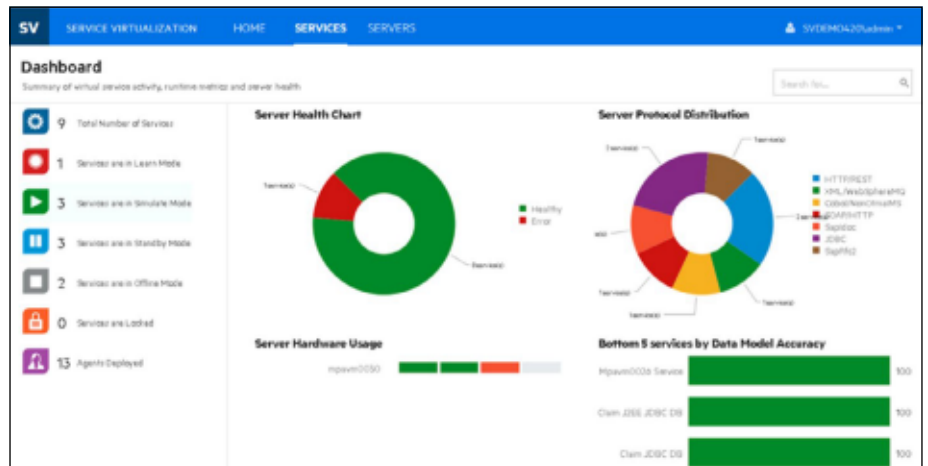
Learn more at [mfgsinc.com](https://mfgsinc.com)

## Resources

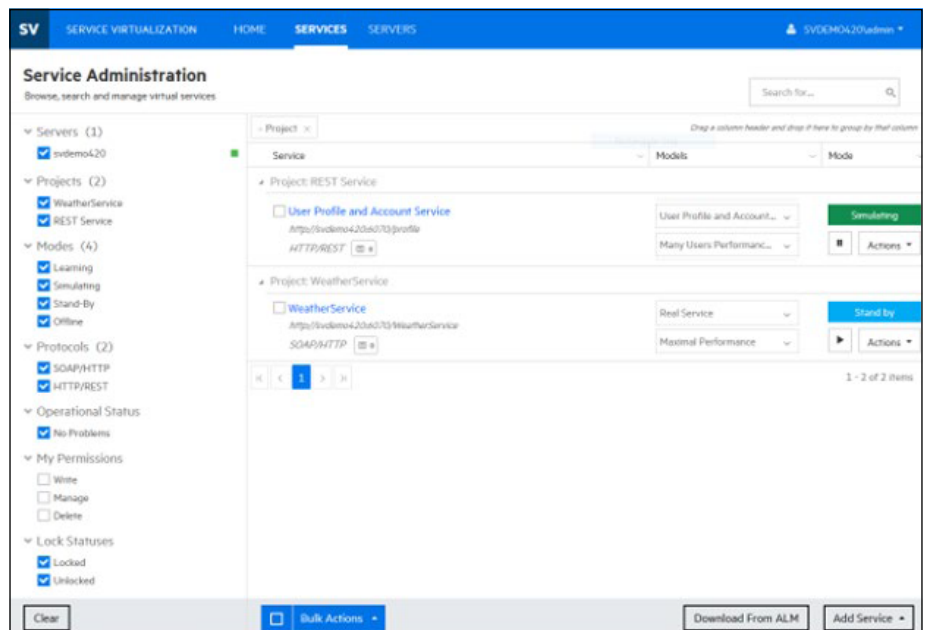
[Visit OpenText Service  
Virtualization web page ›](#)

[Visit DevOps Cloud  
web page ›](#)

[Join the DevOps  
Cloud Community ›](#)



OpenText Service Virtualization supports a variety of technologies run and managed in a scalable server accessible over Web UI and API



Provides remote management, deployment, and control of virtual services, selection of data and performance models, simulation, access to metrics, and more