

# eggPlant Performance

built by TestPlant

## eggPlant Performance Release Notes

Version 7.0.0

### TOC

eggPlant Performance Release Notes .....	2
<b>Hardware</b> .....	17
<b>Microsoft .NET 4.5</b> .....	18
<b>Supported Operating Systems</b> .....	18
<b>Script Builders</b> .....	18
<b>Hardware</b> .....	18
<b>Microsoft .NET 4.5</b> .....	18
<b>Supported Operating Systems</b> .....	18
<b>Java Runtime</b> .....	19

## eggPlant Performance Release Notes

---

### Features and Improvements (v7.0.0)

- The Test Controller UI has been completely redesigned.
  - A new configurable dashboard has been added so you can choose exactly what information you want to see during test execution.
  - The dashboard configuration persists on a per-test basis so that you can customize the information you want to see on a per-test basis.
  - You can configure the time interval to view on each chart (e.g., last 10 minutes, last 30 minutes, entire test run).
  - Errors/warnings tables are now available that display errors/warnings from a test run in a single location.
  - Improved sorting and filtering of all tables: You can now sort/filter on any column in any table.
  - You can open a chart at any point during the test run and data from the whole test run is available immediately.
- Improved Previous Runs View
  - Much more information available about previous test runs.
  - All charts and tables that are available during the test run are also available for previous runs.
- Test Controller REST API
  - It is now possible to fully automate Test Controller via a REST API.
  - Tests can be started, stopped, and monitored using REST, making for easier integration into CI systems or other processes.
- File Transfer Improvements
  - File transfer from Test Controller to injectors has been improved.
  - File transfer progress is now displayed on a per-injector basis.
  - File transfer can be paused, resumed, or cancelled on a per-injector basis.
  - Additional options for communicating with injectors have been added. It is now possible to choose whether to encrypt communication with injectors, and whether to zip files before transfer.
  - File transfer automatically recovers from connection failures and resumes the file transfer.
- Performance Improvements
  - Scalability has been improved: Large quantities of data generated by large tests are now managed more effectively.
  - New UI is more responsive when handling large quantities of data.
- IPV6 Support
  - Full support for IPV6 during recording, generation, and replay of web scripts.
- Many more bug fixes and improvements.

### Features and Improvements (v6.3.0)

- Round-trip script editing of Web scripts.
  - You can now use Generation Rules without losing the ability to manually edit your script code through the use of pre-defined Regions, located in specific parts of the script.
  - Any code placed within a Region persists between regenerations of the script.
- Improved handling of ASP.NET-based web sites
  - Web pages written using ASP.NET pose a challenge when trying to record and replay scripts for performance testing. In particular, ASP.NET uses hidden fields to transfer and persist data between the server and the client. Hidden fields are likely to be different for each test run, and must be correlated carefully so that the virtual users send correct data back to the server each time.  
The ASP.NET virtual user addresses these concerns by automatically parsing and handling hidden fields on

ASP.NET web pages. It is based on the C# Web virtual user, but comes with additional methods as described in the C# API.

- Generation Rules Assistant
  - The Generation Rules Assistant aids creation of generation rules by analyzing pieces of text.
  - It can then suggest whether a value needs to be correlated or replaced.
- New **Data List Correlation** Generation Rule
  - Allows for the easy extraction of multiple variables from a list, and subsequently logic for selecting from these lists (random, particular item, etc.)
- New **Verify Response Length** Generation Rule
  - Allows easy verification that the length of a response is what we expect, within a tolerance value
- New **Custom Code** Generation Rule
  - Allows custom code to be automatically inserted into the script before/after certain requests are sent to the server.
- Support for background scripts.
  - A Background Script is a script that runs in parallel with the foreground scripts that make up a Virtual User's workflow.
  - New methods have been added to the Java and C# API's to control background scripts: there are methods to create, start, stop, suspend and resume background scripts
  - Background scripts allow easier replication of complicated browser behavior, such as asynchronous polling.
  - A new Generation Rule has been added to make working with background or asynchronous HTTP threads easier. The rule allows one to identify and exclude background requests from the main script. It will create a new empty script where the background requests behavior can be implemented, and insert code into the main script for starting and stopping the background script.
- Generation Rules can now access the current Request, Response and Url objects.
  - Special macros can now be used within code planted by the web generator.
  - The generator will replace the macros with the object that is currently in scope.
- Support for regenerating multiple scripts simultaneously.
- Improved integration with eggPlant Functional
  - Creating a script from an eggPlant Functional suite now prompts the user with a new dialog where one can specify the System-Under-Test connection details, e.g. which type of connection to use (RDP/VNC), user account and password, hostname and IP address, etc.  
The information specified in this dialog is then populated in the resulting ePP script, where new overloads for the Connect() API method make it much easier to initiate a connection to the SUT through ePP.
  - Support has been added for iterations and workflows; it is no longer necessary to define these manually in SenseTalk.
  - All eggPlant Performance data bindings are now accessible from any eggPlant Functional script that is executed by eggPlant Performance.  
The data is placed in a special property list, called performance\_data.
- OCR support has been added to the Citrix Virtual User
  - OCR support is provided by the ABBYY FineReader engine.
  - The eggPlant Performance OCR engine must be installed on all injectors in order to use this functionality.
- Support for tolerance and discrepancy in image matching
  - Image matching improvements are available in both the Citrix and WinDriver virtual users.
- Support for associating and disassociating Elastic IP Addresses with Amazon Cloud injectors
- The web generator now supports multiple live values of a correlated value
  - The variable name planted by a Data Correlation rule now contains the index number of the response it was extracted from.
  - This allows multiple different values to be extracted by the same rule and for all the values to be "live" at the same time.

- The web generator now automatically disposes Request/Response objects when they are no longer needed.
  - Request/Response objects can be large, so we can make more efficient use of available memory by disposing of them as soon as we have finished with them, rather than waiting for the garbage collector to collect them.
  - For C#, the web generator now wraps all Request and Response objects in a using block.
  - For Java, the web generator now wraps all Request and Response objects in a try-with-resource block.
- Data correlation rules are now able to correlate data sent in a redirected response
  - New methods Response.getRedirectedFrom(int depth)/Response.GetRedirectedFrom(int depth) added to the Java/C# web API's, respectively.
  - Calls to these new methods are planted by the web generator if it finds data that needs to be correlated in a response that has been redirected.
- Added the ability to exclude responses containing binary data from Data correlation rules
- Improvements to Generation Rule filters
  - Filter by response code
  - Filter by POST data
- Data correlation rules can now extract data from HTTP response headers, as well as from the body of the response.
- Support for custom error reporting for different HTTP status codes
  - New methods in the API to control whether a specific HTTP status code (e.g. 404 Not Found) results in a warning or an error
  - Also allows custom handling of non-standard HTTP status codes returned by some servers.

## Features and Improvements (v6.2.0)

- Comprehensive support for SOAP Web Services.
  - Generate code for testing a web service from a WSDL.
  - Generated code can be C# or Java.
  - SOAP 1.1 and 1.2 are both supported.
  - Easily generate scripts that target the web service.
  - All serialization/deserialization of XML is handled behind the scenes, so scripts are clean and simple, with no complicated correlations or transformations necessary.
- Improved recording of mobile applications using the Proxy Recorder.
  - The Trusted Root Certificate generated by the Proxy Recorder is now accepted by iOS and Android devices, allowing HTTPS traffic from apps to be recorded.
  - Improved UI guides you through the process of setting up recording of HTTP(S) traffic to/from your mobile device.
- Better support for handling timestamps in web recordings
  - Automatically identify timestamps and generate a timestamp to replace them with.
  - Replacements can be either the current time, a fixed time, or a time generated by custom code.
- Data correlation generation rules now allow you to specify a regular expression to use during extraction.
- Support for Microsoft Visual Studio 2015 and .NET 4.6.
- Improved management of .NET assemblies.
- Improved image scalling in the eggPlant Functional viewer.
- Improved event log filtering in Test Controller.
- Automatic check for new releases of Studio and Analyzer.
- New overloads for the ExtractList() method.
- New GetValue() method added to the Url class.
- Project exports now have the extension .epa (eggPlant Performance Archive), rather than .zip (but both formats are still able to be imported)
- You can exclude specific extensions from a project archive (for example, you could exclude video content in order to reduce the size of the export)

## Features and Improvements (v6.1.0)

- Completely new eggPlant Performance Analyzer component
  - Powerful new chart creation.
  - Ability to filter/split data by injector/script/group.
  - Improved reporting, including PDF reports
  - New comparison charts allow the user to compare multiple test runs and easily identify trends.
  - Key Performance Indicators can be defined and reported on.
  - REST API allows full automation of the analysis and reporting process.
  - Uses a Postgres database to store the analyzed data.

## Features and Improvements (v6.0.0)

- Integration with Cloud Providers
    - Integration with Amazon EC2 and Microsoft Azure allows injector capacity to be added on demand.
    - Create new injector virtual machines in the cloud with the click of a button.
    - When a test run is started, all the required cloud injectors are started automatically.
    - At the end of a test run, the cloud injectors can be automatically shut down or rebooted.
    - Any cloud injectors that are no longer required can be deleted with the click of a button.
  - Injector Families
    - Injectors with the same characteristics are now grouped into **Injector Families** for easy configuration.
  - Script debugging with Visual Studio and Eclipse
    - Individual scripts can be run in an IDE, allowing step by step debugging.
    - For C# scripts, we support debugging in Visual Studio.
    - For Java scripts, we support debugging in Eclipse.
  - Dynamically stop virtual users during a test run
    - You can choose to stop specific virtual users at any time during the test run.
  - Change the pause factor of a group during a test run
    - You can change the pause factor for either the whole test, or for individual groups, at any time during the test run.
  - Automatic correlation of cookie values
    - If a cookie value, such as a session ID, appears in the path or query part of a URL then it will automatically be replaced with the correct value during replay.
  - New style licenses
    - The licensing model has been simplified.
    - There are no longer separate licenses to unlock the machine and to activate virtual users.
    - Instead, there are simply **node locked** licenses that are tied to a particular computer, or **evaluation** licenses that can be used on any computer
  - Shared Data Server scalability improvements
    - Connections to the Shared Data Server from each engine are now managed intelligently by the engine; you no longer have to manually open/close the connections in your scripts.
    - Sharing connections across virtual users massively increases the number of virtual users that can communicate with a single Shared Data Server.
  - Export individual test runs from Test Controller or Analyzer
    - Individual test runs can be exported from Test Controller or Analyzer in the form of a zip file.
  - Removal of support for C++ Web
    - **C++ web scripts are no longer supported.** If you have any C++ web scripts in your workspace then you will be unable to upgrade your workspace to work with version 6.0
-

## Features and Improvements (v5.4.0)

- Web improvements
    - When generating scripts from a Web recording, the powerful **Script Generation Rules wizard** enables you to automatically correlate and parameterize your scripts. This reduces the time needed to get your tests ready to run, because fewer manual script changes will be required.
    - You can generate multiple scripts from the same recording (for example, dividing a single Web recording into Login/Processing/Logout scripts).
    - The **eggPlant Proxy Recorder** now has improved handling of HTTPS traffic and SSL certificates.
  - Citrix improvements
    - The **eggPlant Citrix Recorder** no longer requires an ICA file to connect to a Citrix server, and can automatically log in and launch an application upon connecting.
    - Usability of the **eggPlant Citrix Recorder** to capture images and Citrix events has been improved.
    - Generated Citrix scripts are now more streamlined and require fewer manual script changes.
    - New, simpler API methods have been added to make scripting easier.
    - The **WinSpy** tool image capture functionality is greatly improved.
  - Recording and script generation
    - Recordings now appear in the **Project** tree, so they can be managed more easily.
    - The new **Recording wizard** makes it much easier to record interactions with the System Under Test.
    - The new **Script Generation wizard** makes it much easier to generate scripts from a recording.
  - Project management
    - Recordings, scripts, workflows and tests can all be cloned, renamed, and organised into custom folders.
    - **eggPlant Performance Studio** now saves workspace files in a way optimised for use with **Source Code Management** applications (such as **SVN**, **Mercurial** or **Git**). This provides an easy way for test team members to share and collaborate on the same performance testing project.
    - The management of external libraries and JARs within Custom Virtual User profiles has been improved; the libraries and JARs are copied to a directory within the workspace, so they can be more easily exported and moved with the workspace. Also you can browse to add multiple libraries at once.
  - Improvements to existing functionality
    - The requirement to run **eggPlant Studio** with Administrator privileges has been removed.
    - The **Linux monitor** now displays disk mount points alongside the device names.
    - The **iOS monitor** works with iOS 7 and now iOS 8.
    - Improved support for Unicode.
    - Added support for the HTTP **OPTIONS** method.
    - Many bug fixes and minor improvements.
  - Removal of support
    - There is no longer the option to generate **Web C++** scripts from a Web recording.
    - **Web C++** scripts can still be executed by virtual users in a test, but support for this will be removed in a future release.
    - **eggPlant Performance** no longer supports **Windows XP** or **Windows Server 2003**.
- 

## Features and Improvements (v5.3.0)

- Cross-platform Java injectors
  - New in eggPlant Performance 5.3 is the ability to install eggPlant Performance injector software on **Linux** and **OS X 10.9 (Mavericks)** machines, as well as Windows machines. These cross-platform injectors can run any kind of **Java Virtual User**.
- Mobile device monitoring

- New monitor types have been added to the existing monitoring infrastructure, enabling the monitoring of **iOS** and **Android** devices.
    - **iOS 7.x** devices can be monitored with the use of the **eggMonitor** app for iPhone and iPad, which listens for network connections from the monitor running on the injector machine.
    - **Android** devices can be monitored when connected to an injector via a USB cable, using **adb** (Android Debug Bridge).
  - Viewing live eggPlant Functional virtual user screenshots during a test
    - When running eggPlant Functional virtual users, it is helpful to be able to see the screen display of the System Under Test which the virtual user is interacting with. The new **eggPlant Functional Viewer** can be launched from eggPlant Performance Test Controller while a test is in progress, and enables you to both see a snapshot of the current screen display, and browse through previous snapshots in order to check progress and diagnose visual problems.
  - Import HAR web recording files
    - eggPlant Performance Studio provides the capability to create recordings of web requests as a user interacts with pages in their browser. However if your web recordings already exist in **HAR (HTTP Archive) format**, they can easily be imported into Studio. From there, the host filtering, automatic script generation, and rule-based modifications are available as usual.
  - Improvements to existing functionality
    - New initialisation methods have been added to the *com.facilita.fc.selenium.SeleniumVirtualUserScript* class to make it even easier to get started with different Web Drivers.
    - Eclipse projects are now automatically created when the IDE is launched from eggPlant Performance Studio.
    - Recorded web recordings can be discarded and re-recorded if a mistake is made during recording, or even after a script has been generated from them. This makes it easier to re-record and regenerate scripts if something on the System Under Test changes.
    - The whole eggPlant Performance suite has been internationalized.
    - The user interface looks better on Windows 8 and 8.1
    - Many bug fixes and minor improvements
- 

## Features and Improvements (v5.2.0)

- Monitoring
  - Enhanced monitoring of System Under Test components. New monitor components can run on any eggPlant Performance injector. The monitors access the SUT remotely and collect data without the need to install any software on the SUT.
  - Data collected by the monitors is available in real-time, and is also imported into eggPlant Performance Analyzer for later analysis.
  - The **Windows Performance Monitor** collects data from Windows servers, including counters for SQL Server, ASP.NET, IIS and .NET.
  - The **Linux** monitor collects data from Linux servers, using either ssh or telnet to connect to the SUT and collect data such as CPU and memory usage, disk I/O and network traffic.
  - The **JMX** monitor collects data from java-based servers such as Apache Tomcat, JBOSS or IBM WebSphere.
  - New ability to import **sar** data collected on a linux server directly into eggPlant Performance Analyzer.
- Host filtering
  - New ability to filter the list of hosts in a web recording, for example, to exclude ad-servers from the recording.
- eggPlant Functional licensing
  - New licensing mechanism for eggPlant Functional, allowing easy application of eggPlant Functional licenses to multiple injector machines.
- New software version support
  - Support for Microsoft Visual Studio 2013.
  - Support for Microsoft Windows 8.1

---

## Features and Improvements (v5.1.0)

- You can now dynamically increase the number of virtual users in a test during the test run. This can be done entirely within the Test Controller. You have full control over how many additional virtual users of each type/group to start. When the new users are added to the test, new engines are started for the users and the view in Test Controller will be refreshed to show the status of all virtual users. The new users will use the same settings that are already defined for their corresponding group.
- New preference setting added which allows you to set the number of engines that may be permitted to fail when a test run starts. The default is 0, meaning that if any engine fails to start, the test run will not start.
- Change to existing Web Logging settings - rather than specify particular virtual user index numbers to store web logs for, and due to the ability to add new virtual users to a running test, we decided to simplify this feature. The default setting (100) will record a web log for the 1st virtual user in this group together with the 101st, the 201st, etc., This still allows for full flexibility, but makes the feature more intuitive and easier to use. If new virtual users are added to a running test, this will also apply (web logs will be recorded for new virtual users according to this setting).
- New welcome screen for the workspace and for the project highlights the most common tasks that you may want to perform.
- New Help menu option to display the current Studio log file (you may need to send this to the Support Team in the truly exceptional cases where something does not go 100% smoothly)
- Various bug fixes and improvements

---

## Features and Improvements (v5.0.1)

- A new brand
  - Following the acquisition of Facilita by **TestPlant**, Forecast has been rebranded as **eggPlant Performance**.
  - Improved integration with eggPlant Drive. eggPlant Performance can now start and stop eggPlant Drive instances installed on an injector, automatically transfer eggPlant Functional suites to each injector machine, and automatically capture screenshots in the event of an error in an eggPlant Functional script.
  - A new dialog facilitates easy creation of a performance test based on an existing eggPlant Functional script.
- The Java and C# Web virtual users have now officially been released and are no longer in beta.
- A new dialog makes it easy to create multiple injectors in a single step.
- Various bug fixes and improvements

---

## Features and Improvements (v4.9.1)

- New 64-bit injector allows mixing of 32-bit and 64-bit Virtual User Types in the same workspace/test (for rare cases such as a mix of Base Java VUs which load 64-bit native libraries with Web Java VUs in the same test - [see here for more details](#))
- New Virtual User Group Runtime Ramp-Up option - specify the number of seconds/minutes for the total ramp-up time for the group, and we'll calculate the intervals between starting each virtual user for you. [Diagrams and explanations here](#).
- Forecast Analyzer has been updated due to the new ramp-up option introduced above, we recommend you also update to the latest version of Forecast Analyzer.
- The Selenium Client & WebDriver Language Bindings have been updated to the [latest current version \(v2.32.0\)](#) as of 13/05/2013 (supports Firefox 20)
- Internal core update to support the certificates required for mobile devices
- Various bug fixes and improvements

---

## Features and Improvements (v4.9.0)

- Beta Release of the Web C# Virtual User.
-



## Features and Improvements (v4.8.0)

- Selenium Java Virtual User
    - Introducing the new **Selenium Java Virtual User**, which uses **Selenium WebDriver** to automate web browsers. Selenium WebDriver provides an object-oriented API that provides a common interface for driving all of the major browsers (e.g. Internet Explorer, Firefox, Chrome, iPhone, etc.). It can also drive the **HtmlUnit** GUI-less browser, which is the fastest and most lightweight implementation of WebDriver, and hence is the most appropriate choice for a load testing application.
    - This release packages up the latest Selenium Client & WebDriver Language Bindings (v2.31.0, release date 2013-02-27) which is compatible with Mozilla Firefox (release 19). (Note: version 4.9.1 has an updated version of the bindings that supports Firefox 20).
    - If you need to install a later/earlier version of Firefox, you will need to install a different version of the Language Bindings, read **here for how to configure Studio with a different version of the Language Bindings**
    - Download a **demo**
  - Virtual user types have now been categorised
    - As the list of supported protocols and virtual user types continues to grow in Forecast Studio, we have now categorised the virtual user types and provided some high level details for each type.
    - The categories are:
      - Base
      - GUI
      - Network
      - Web
    - Click on each category type in the Workspace view to see further details.
- 

## Features and Improvements (v4.7.0)

This version contains many new features and a beta pre-release of the new Web Java Virtual User

- Runtimes
  - Support ended for Visual Studio 2005
    - We have now ended support for Visual Studio 2005 (vc8 and cs2 profiles). This runtime and any options or references related to it have been removed. The update process (after archiving your workspace) will remove all references in your workspace.
  - Support added for Visual Studio 2012
    - We have now added support for Visual Studio 2012 (vc11 and cs4\_5 profiles). The update process (after archiving your workspace) will add new references to your workspace.
    - Note that, for Windows XP and Server 2003, when acting as a controller machine - Visual Studio 2012 cannot be installed on these operating systems, so support is not available for compilation of these runtimes. However, Visual Studio 2012 redistributable is supported, so these operating systems can be used as injectors for all runtimes.
  - Default JRE packaged with Studio
    - We have added a default fully compatible 32-bit JDK (Java version 1.7) to the product for runtime of all java tests
- Studio
  - New Enhanced Web Java Virtual User Type
    - Soon to be available in both Java and .NET, both based around a common core. For this beta release, only the Java version is available.
    - Writing scripts in Java allows for a better experience due to improved memory management, superior error reporting (runtime and compilation time), more code libraries are available, easier to program
    - Highly scalable. Supports from one to tens of thousands of virtual users

- Unicode support. Forecast can now handle and display foreign language character sets without using encoded characters
  - Designed to simplify the handling of asynchronous AJAX requests and complex applications
  - An elegant and logical Object Oriented API
  - Fully extensible and integrated with a standard Integrated Development Environment - Eclipse
  - Help
    - The format of the product 'Help' documentation has changed to HTML
    - New basic documentation (also available in Eclipse) for the New Enhanced Web Java Virtual User Type
- 

## Features and Improvements (v4.5.0)

This version contains several new features and is recommended for all users.

- Studio
  - Check for Updates / Get Support ...
    - A new menu option added to the Help menu, to check for the latest updates of all Facilita products has been added to Studio, together with a menu option that will open the Facilita Support Portal in your default browser. You can request a Facilita Support account (email us at [support@testplant.com](mailto:support@testplant.com)) and then (subject to your support and maintenance contract) you can download the latest releases and browse through (and post to) new discussion groups, wiki pages and other documentation. The Facilita Support Portal will be constantly updated with new articles, demos and other features.
  - Touch Source
    - A new menu option added to the Build menu that "touches" all script source files with the current system time. Useful when re-building scripts in an environment that experiences frequent clock rollbacks.
  - Script Exports
    - Scripts can be individually exported into a self-contained archive. Script archives can then be imported into a different project in the same or a different workspace. See the Help for more details. Script archives can be sent to the Facilita Support Team as necessary.
  - Custom Virtual User Exports
    - Custom Virtual Users can be individually exported into a self-contained archive. Virtual user archives can then be imported into a different workspace. See the Help for more details. Custom Virtual User archives can be sent to the Facilita Support Team as necessary.
  - Test Run Results Exports
    - Test Run Results can be individually exported into a self-contained archive. Test Run Results archives can be sent to the Facilita Support Team as necessary.
  - Web Recorder
    - The Web Proxy Recorder now excludes hosts by pattern matching. Previous exclusions were actioned through a mix of check boxes and pattern matches. Any existing host exclusions will be upgraded automatically to the new format.
  - Web Viewer and Runtime Web Log Viewer
    - The two viewers have been merged into a single new viewer. New functionality allows real time web logs from remote injectors to be viewed on the controller machine. New menu options in the viewer make it easier to compare the runtime and recording logs side by side. The number of iterations displayed in the runtime logs is now a configurable property of the test group.
  - Documentation
    - New Citrix Virtual User guide added to the Studio Help
    - New IP Virtual User guide added to the Studio Help
- Test Controller

- Clear Obsolete Transactions
    - Added a button/menu item to the Transactions/Timings screen in Test Controller to clear the transactions/timings table.
  - Analyzer
    - Regular Expression matching for Transactions
      - The user can now group otherwise identical HTTP requests that vary only by the presence of a dynamic string within the path.
    - Improved memory usage
      - Memory usage has been improved making the product more robust and able to cope with extremely large datasets.
- 

## Features and Improvements (v4.4.0)

This version is a major release, recommended for all users

- Studio
  - Web recording:
    - When recording a web script there is a new list of applications that can be launched automatically by the Recorder. This list contains
      - Google Chrome
      - Microsoft Internet Explorer and
      - Mozilla Firefox.
    - If any of these applications are not installed on the controller machine then they cannot be selected from the list
    - When taking a web recording, a snapshot of the browser window is taken every time a transaction name is entered into the Recorder dialog. A thumbnail strip of snapshots has been added to the Recorder dialog. When the recording is viewed in the Web Viewer application, a new tab 'Snapshot' displays the snapshot that was taken at the time of recording and matches the currently viewed transaction
  - Another new feature when recording a web recording is the 'Hosts' tab, which allows the de-selection of hostnames that have been encountered during the recording, and when the web script is generated these hosts will be ignored at runtime. The hostnames are persisted at the project level, so when another recording is taken the hosts that were excluded from the previous recording can also be excluded for the current recording (this is completely configurable from the Hosts tab in the Recorder). Hostnames can be deselected individually and by defining regular expressions
  - Another new tab added to the Recorder is the 'Messages' tab, which displays a log of commands and any messages
  - The 'Stop' (recording) button has been added to the Web Script Recorder dialog but removed from the underlying dialog, to simplify the user interface
  - Transactions can be renamed whilst a script is being recorded
  - Web Script generation:
    - The protocol(s) for a generated web script have now been parameterised, making it easier to change the protocol(s) in each script
    - In generated web scripts we now write `setHeaderValue()` calls for all non-standard headers encountered in the recording, ensuring they will be automatically set at runtime
  - **NetBeans** integration:
    - If NetBeans is installed, it is possible to launch this IDE from within Studio and build the (java) Forecast project within NetBeans

- **dynaTrace** integration (6.9.1, 7.x):
    - If dynaTrace is installed, there will be a new tab on the left-hand side of Studio named 'dynaTrace'. Various dynaTrace configuration options have been added to this new tab. The dynaTrace header can be set on a per virtual user group basis by setting the new option on the 'Web: Logging' tab of the 'Runtime Settings' tab for the virtual user group
  - Test export:
    - This new feature within Forecast Studio allows tests to be exported as single .zip files that contain everything required to run the test. The .zip file can be extracted onto any target machine and run without having to install any other Facilita software.
    - The limitation is that only tests that are defined to run with a single virtual user can be exported
  - New product icons, splash screen and logo
  - Communications
    - The communications between Test Controller/Injectors/Engines have been recoded to provide a more robust and reliable service
    - Error message texts have been improved
    - All files sent between Test Controller/Injectors/Engines are now zipped for transfer thereby decreasing transfer times
    - A new status bar added showing the progress for the retrieval of the log files from the remote injectors
    - The default injector and engine port numbers have changed to ensure that we do not use ports in the dynamically allocated range (of ephemeral ports), this will resolve port conflicts with other running applications and provide smoother test executions. This change complies with (and for further information, see) [this article](#)
    - New registry settings added to optimize the usage of TCP ports for all versions of windows. For a controller machine, run 'Optimize System Settings ...' from the Tools menu in Test Controller. Reboot the injector machine as appropriate to ensure the registry settings are applied.
    - For injector machines, click on forecastOptimalSettings\_x86.reg or forecastOptimalSettings\_x64.reg from within the installation folder. Reboot the injector machine as appropriate to ensure the registry settings are applied
    - When running a large number of engines on an injector, if there are any problems starting all of the engines, increase the Injector 'Connect-back timeout for engines' value to ensure that all engines start correctly
  - Test Controller
    - We have added new 'Aborted' and 'Unknown' states to Test Controller which more clearly define the known statuses of the running virtual users
    - The 'Last Alert' is now displayed at the top of the main window and is updated in real time
  - Analyzer
    - Analyzer will now warn (during database creation) if not all of the expected virtual user log files are available for a test run
  - Miscellaneous
    - Many other issues resolved and many small additional features added
- 

## Features and Improvements (v4.3.0)

This major release has been superseded by 4.4.0, which is recommended for all users

- Studio
  - The Studio graphical user interface has undergone a technology refresh and looks cleaner and crisper than before.
  - When loading a workspace that was created with a previous version of Studio, an upgrader will automatically run and provides an automated backup function.

- The 'Handle Dynamic URLs' web-script generation option has been removed.
  - The workspace and project are now built together, F8 performs a build ALT+F8 performs a full rebuild. The options to build the project and workspace separately have been removed, whereas the ability to open each separately in an IDE has been maintained.
  - Much improved dynamic build output messaging windows, which can be maximised or hidden from view until needed (press SHIFT+ESCAPE or use context menu).
    - Whenever there is a compile-time error, clicking on an entry in these windows will open the script at the failing line in the Scite text editor.
    - New (clickable) build indicators added.
    - New build progress bar added.
  - Web
    - The Internet Explorer Recorder has been removed, as has the old Proxy Recorder. These have been replaced by the new Fiddler Core Proxy Recorder, which provides an option to delete the Internet Explorer cache and files before recording starts.
    - New HTTP 'PUT' and 'DELETE' support.
    - Added support for the WinHTTP socket handler. This handles NTLM authentication better than WinInet. WinHTTP should be used in preference to WinInet for HTTP and HTTPS.
    - The setOptions() method call in web scripts has been replaced with a new pre() method.
  - Other Protocols
    - Custom Virtual User Scripts now have pre() methods (all protocols).
    - New TCP recorder and generator. New scriptable UDP support (generator to be released soon).
  - Test Controller
    - Now reloads changes to the workspace or tests automatically.
  - Miscellaneous
    - Many further minor improvements and enhancements (>200).
- 

## Features and Improvements (v4.2.3)

This version is a minor release, recommended for those users who wish to take advantage of the new Visual Studio 2010 support, there is no additional functionality

- Support for Microsoft Visual Studio 2010 (full and express editions - express editions must be registered): all operating systems above (and including) Windows XP/2003 now fully support the building of workspaces and projects within Visual Studio 2010 (C++ and C#)
  - Optional Injector update will send alerts to the Test Controller if the required .NET 4 frameworks are not installed on the Injector machine
  - Numerous bug clearances
- 

## Features and Improvements (v4.2.2)

This version is a maintenance release, providing numerous minor improvements and enhancements

- setAllowOverlappingTransactions() and getAllowOverlappingTransactions() have been removed from the VirtualUser and VirtualUserScript classes. These methods are no longer valid as the transaction handling logic has changed and we no longer need to make this distinction
- WinDriver support added for Windows Presentation Foundation (WPF)
- Enhancements to Cookie handling in web script generator. Will now accurately track client originated cookies set by client-side scripting as well as those in the cache when recording started
- New VirtualUserScript methods to retrieve the elapsed time since a timing or transaction was started

- Web 'Content check' rule type has been extended such that script code can be inserted before and after the navigation call when text is matched in the page source
  - Numerous bug clearances
- 

### Features and Improvements (v4.2.1)

- Injectors are now self-maintaining. Once an injector has been installed on a machine, it should not require updating. When a test connects to an Injector, all appropriate libraries, data files and engines will be transferred, meaning that when changes are made to the engine code in the main product it is just the controller machine which will require updating, and the new engines will automatically be transferred to the injector machines
  - Numerous minor improvements and enhancements.
- 

### Features and Improvements (v4.2.0)

- Transactions tab has been removed from web script generation options. The generator now auto-detects whether transactions exist.
  - User defined metrics are collected at runtime, and have corresponding charts in Analyzer and Test Controller ('Average', 'Sum' and 'Rate').
  - Web metrics are collected at runtime, and are written to the VU logs - from this additional data new charts have been created, and added to both Analyzer and Test Controller.
  - Using higher resolution multimedia timer, instead of standard win32 timers, to improve the accuracy of timings.
  - New 'filter' feature implemented for the event logs in Analyzer and Test Controller.
  - Product now built against .NET 3.5.
  - Web runtime: more efficient data compression support & direct access to this functionality from scripts.
  - TCP/IP network traffic virtual user support.
  - Runtime supports "Allow overlapping transactions" functionality.
  - Numerous minor improvements and enhancements.
- 

### Features and Improvements (v4.1.1)

#### Version 4.1.1

- Licensing has changed and is different to v4.1.0 and any previous release.
  - Web Service scripts can now be generated from XML source files in Studio.
  - A new .zip file allows for easier and faster installation of injectors (C++ and Web) without the need to install the full Forecast product on the injector machine.
  - A new .msi file allows for easier and faster installation of injectors (all protocols) without the need to install the full Forecast product on the injector machine.
  - Proxy Recorder SSL Certificate Handling - A change has been made in Release 4.1.1 so that the proxy recorder will now create a valid certificate for each new HTTPS site recorded. The certificate is issued by the Facilita Certification Authority. To prevent warnings from your web browser about this certificate you should import the Facilita Root Certificate as a Trusted Root Certificate Authority. Instructions for installing this certificate are shown on the Proxy Recorder Settings dialog, available from the Proxy Recorder wizard "Application Launcher" page. After the Facilita Root Certificate is installed as a trusted root certification authority, then no further browser warnings about certificates will be displayed.  
Also available on the Proxy Recorder Settings dialog is the option to set a path to a client certificate file. This is required if your web server requires a valid certificate to be sent from the client - which is the proxy recorder when recording HTTPS. Note that this client certificate must be in PEM format. The open source tool openssl.exe can be used to convert certificates to PEM format. This tool is delivered in the Forecast installation bin folder.
  - Many fixes and minor improvements, see the Releases section of the website for details of these
-

## Features and Improvements (v4.1.0)

- Licensing has changed. There is no longer a need for a 'runtime.lic' file. See [Forecast Controller Installation and Licensing](#) or the User Guide for more details.
- Web generation rules are now defined using the GUI. Rules can be applied at both the virtual user and Script level. See the User Guide for more details. Old style (.ini) generation rules can still be processed but we recommend that you convert these to the new style.
- Application logs are now automatically flushed. It is no longer necessary to close Studio or Test Controller in order to flush the logs.
- An Analyzer SDK (.NET) that enables programmatic creation of Analyzer databases and views. Note that charting functionality is not yet available from the SDK.
- Framework mode has been removed. Custom VUs are now accessed through the Framework tab and built via the Build menu option.
- New Python APIs allow tests to be run from the console or from user created Python scripts.
- Import and export data with the Shared Data Server in .csv format.
- Workspaces can now only be opened by one user at a time.
- Many improvements to the handling of multiple injectors/engines.
- Studio will no longer start/stop the local injector processes, Test Controller will start any required local injector processes and terminate them when no longer required. The local injector is no longer installed as a service by the installer. This does however mean that manual installation of the service on remote injector machines is required. Do this by opening a command prompt and navigating to the %FORECAST4HOME%\bin\ folder . Here type the following to install the injector service;

```
fcinjector -i
```

If for any reason you want to remove the injector service use the -u switch. The Forecast installer is used to uninstall the release when Forecast has been removed using Add/Remove programs. This will not remove the injector service. Therefore when upgrading a remote injector from V3 or V4.0 it is unlikely that a manual install of the service will be required and likewise when upgrading a remote V4.1 injector which has previously received a service installation from the command line.

- Numerous minor improvements and enhancements.

---

## Migrating from Version 3 to Version 4

- A project upgrader is distributed with V4 to enable upgrading of projects from V3 to V4.
- Manual edits will be required to some existing scripts as a result of migration.
  - Data Streams no longer exist within Version 4. Remove any #include "DataDictionaryStream.h" statements to ensure successful compilation.
- Data Streams and Data Tables have merged into one data source; namely Data Table. To achieve Data Stream functionality assign a Data Table at the VU Group level.
- Data Dictionary data source files can only be assigned at the Test level.

---

## Migrating from Version 4.1.0 to Version 4.1.1 (if upgrading from an earlier release, read the appropriate section(s) above)

Licensing has changed, it will be necessary to perform the following steps

1. De-register v4.1.0 (run <facilitate\_installation\_dir>\bin\fcstudio unregister and follow the instructions provided by the dialogs
2. Uninstall v4.1.0 (*Control Panel->Add/remove programs*)

3. Install v4.1.1 (a new hardware fingerprint will be generated) and forward the new hardware fingerprint to [support@testplant.com](mailto:support@testplant.com)
  4. Install the new licenses issued by Facilita, instructions for this will be provided along with the new licenses
- 

Migrating from Version 4.1.1 to Version 4.2.1 (if upgrading from an earlier release, read the appropriate section(s) above)

All injector machines should be upgraded to the new self-maintaining injectors (see above for instructions on upgrading injector machines)

---

Upgrading from Version 4.X to Version 4.2.3 (if upgrading from an earlier release, read the appropriate section(s) above)

Workspaces must be updated to include the necessary supporting files for Visual Studio 2010. You should always archive your existing workspaces before upgrading to a new version of Forecast. In v4.2.3, whenever a new workspace is opened in Studio, an upgrade check will run automatically and issue a prompt, requesting whether to upgrade the workspace

The prompt will also remind you that this operation is one-way, so once a workspace has been upgraded to v4.2.3, it cannot be reverted back to a previous version. Upon encountering this prompt, you can either proceed or you can cancel the upgrade process, and your workspace will not be upgraded, allowing you to first archive your workspace and then begin the upgrade operation again.

If you wish to run scripts compiled with Visual Studio 2010, you must either update each injector machine with the latest v4.2.3 injector release, or install the **Visual Studio 2010 runtime**.

---

Upgrading from Version 4.2.3 to Version 4.3.0 (if upgrading from an earlier release, read the appropriate section(s) above)

When a workspace that was created with a version previous to v4.3.0 is opened, an automatic upgrader will run, and an automated backup process will run. An archive of your workspace is necessary as changes will be made that are not backward-compatible. Once the upgrader has completed the workspace will be opened. Whenever an upgrade to Studio is applied, please perform a full rebuild of your workspace and projects.

---

Upgrading from Version 4.3.0 to Version 4.4.0 (if upgrading from an earlier release, read the appropriate section(s) above)

To upgrade to the latest version of Forecast Studio, all Injector machines must also be upgraded.

The TCP/IP Virtual User has been superseded by the new IP virtual user - if you have any custom virtual users or any scripts based on the existing TCP/IP Virtual User, please contact the Facilita support team before upgrading.

---

Upgrading from Version 4.4.0 to Version 4.5.0 (if upgrading from an earlier release, read the appropriate section(s) above)

For controller machines, if upgrades are necessary the automated upgrader will run and prompt you to archive your workspace before it makes any changes

---

Upgrading from Version 4.5.0 to Version 4.7.0 (if upgrading from an earlier release, read the appropriate section(s) above)

The automated upgrader must run as there are many changes - removed support for Visual Studio 2005, added support for Visual Studio 2012, the installer will add the new Web Java Virtual User type

---

Upgrading from Version 4.7.0 to Version 4.8.0 (if upgrading from an earlier release, read the appropriate section(s) above)

No changes required. The installer will add the new Selenium Virtual User type

---



Upgrading from Version 4.8.0 to Version 4.9.0 (if upgrading from an earlier release, read the appropriate section(s) above)

No changes required. the installer will add the new Web C# Virtual User type

---

Upgrading from Version 4.9.0 to Version 4.9.1 (if upgrading from an earlier release, read the appropriate section(s) above)

All users should upgrade to the latest version of Forecast Analyzer

---

Upgrading from Version 4.9.x to Version 5.x (if upgrading from an earlier release, read the appropriate section(s) above)

No changes required.

---

## Known Issues and Limitations

- On Windows machines, after installation, you may need to change the permissions level of the 'forecast4 Injector Listener' service so that it will run under a local administrator account. By default the service will be setup to use the Local system account. If you receive errors such as the following when trying to execute a test you will need to change permissions level for the service.

### **C++ Engine:**

External data: Cannot load library C:\fcCache\%FORECAST4HOME%\bin\fc\_vc10.dll

### **Java Engine:**

pipe has ended. NoClassDefFoundError

### **Steps for changing Injector service permissions**

1. Select 'Run' from the Start menu.
2. Enter services.msc.
3. From the list of services, find 'forecast4 Injector Listener', right-click and select Properties.
4. Select the 'Log on' tab.
5. Check 'this account' and enter a local administrator username and password.
6. 'OK' the changes.

Alternatively the injector process can be started from a command line:

```
%FORECAST4HOME%\bin\fcinjector -e -p <port_number>
```

- Version 4 now executes project binary files from a 'File Transfer Repository'. By default this is <ForecastInstallationDrive>\fcCache

For this reason, the user account which Forecast is installed under will need to have write access to the root of <ForecastInstallationDrive> (default is C:).

- The lightweight web page viewer used for the web log in RunMonitor and the web recording in Studio does not adequately display web pages that utilize style sheets.
- The OpenSSL default timeout is currently set to four minutes, and cannot be altered in this version.

## Prerequisites

### eggPlant Performance Controller

#### Hardware

- Recommended minimum 2GHz processor, 2Gb RAM, 200Mb free for the installation. You will also need additional disk space for the test run results data.

#### Microsoft .NET 4.5

- To run Studio, Test Controller, Analyzer, or the C# engine, Microsoft .NET 4.5 must be installed (<http://www.microsoft.com/en-us/download/details.aspx?id=30653>)
- Additionally, Microsoft .NET Framework 4 Client Profile and Microsoft .NET Framework 4 Extended must both be installed (available via Microsoft Update, and also from [http://msdn.microsoft.com/en-us/library/5a4x27ek.aspx#standalone\\_redistributable\\_packages](http://msdn.microsoft.com/en-us/library/5a4x27ek.aspx#standalone_redistributable_packages))

#### Supported Operating Systems

- Windows Vista SP1 (64-bit)
- Windows 7 SP1 (64-bit)
- Windows 8.1 (64-bit)
- Windows 10 (64-bit)
- Windows Server 2008 (64-bit)
- Windows Server 2012 (64-bit)

#### Script Builders

Virtual User scripts are available in a number of languages dependent upon the VU type.

- For C++ and C# virtual users, any one of the following are suitable (latest version preferred):
  - Microsoft Visual Studio .NET 2008
  - Microsoft Visual Express Edition 2008 (freely available from Microsoft)
  - Microsoft Visual Studio .NET 2010
  - Microsoft Visual Express Edition 2010 (freely available from Microsoft, but must be registered)
  - Microsoft Visual Studio .NET 2012
  - Microsoft Visual Express Edition 2012 (freely available from Microsoft, but must be registered)
  - Microsoft Visual Studio .NET 2013
  - Microsoft Visual Express Edition 2013 (freely available from Microsoft, but must be registered)
- For Java Virtual Users
  - 32-bit Java JDK 1.5 or later (we supply a Java 1.8 JDK with Studio), *we also supply a new Eclipse plugin for integration with Eclipse*

#### eggPlant Performance Injectors

##### Hardware

The system requirements needed to support a load test are dependent upon a number of factors. The specifications listed below will comfortably support a typical 1000 Web virtual user test.

- Recommended minimum 2GHz processor, 2Gb RAM, 100Mb free for the installation. You will also need additional disk space for the test run results data.

#### Microsoft .NET 4.5

- To run the C# engine, Microsoft .NET 4.5 redistributable must be installed; it is installed automatically when the injector is installed.
- Additionally, Microsoft .NET Framework 4 Client Profile and Extended Profile must be installed (available via Microsoft Update, and also from [http://msdn.microsoft.com/en-us/library/5a4x27ek.aspx#standalone\\_redistributable\\_packages](http://msdn.microsoft.com/en-us/library/5a4x27ek.aspx#standalone_redistributable_packages))

#### Supported Operating Systems

- Windows Vista SP1 (32-bit and 64-bit)
- Windows 7 SP1 (32-bit and 64-bit)
- Windows 8.1 (32-bit and 64-bit - all versions except RT)
- Windows 10 (32-bit and 64-bit)

- Windows Server 2008 (32-bit and 64-bit)
- Windows Server 2012 (32-bit and 64-bit)

#### Java Runtime

- The Java Runtime on the injector should be compatible with the runtime on the Controller. We recommend that you use the packaged version.
- The Java Runtime must be 32-bit as 64-bit JRE's are not supported.

## eggPlant Performance Controller Installation

### Installation

- On the Windows PC that will act as the Controller, log in as a user with local administrator privileges and execute either the `eggPlantPerformance<version>.msi` or the `eggPlantPerformanceInjector<version>.msi`.  
You will need at least 500Mb of free disk space on the installation drive if installing Studio plus injector, or just 50Mb for the standalone injector (though more space will be required for running tests as the engines and various libraries will be transmitted to the injector at runtime). You must also have disk space available for recordings and test run data. This data can range from a few kilobytes to several gigabytes, depending on the test run options selected, the level of logging, the duration of your test runs, and the number of simulated virtual users.
- Start eggPlant Performance Studio.
- A dialog will be displayed showing the current machine's Host ID
- Email this Host ID to [support@testplant.com](mailto:support@testplant.com) and request your license. A valid license will be emailed back to you by the support team.
- Click the **Add a license** button, then enter the user name key you have been given. Click **OK**.

### Uninstallation

The eggPlant Performance uninstall program does not remove your workspace or project data. If you do not want to save any of your workspaces, projects, or test results, you must manually delete them.

1. Close all eggPlant Performance component instances.
2. Uninstall via Windows Control Panel, Add or Remove Programs.

### Re-installation on the same Controller machine

If you intend to update your installation with a new release, you must carry out the following steps (note that injector-only machines should not require updating):

1. Perform uninstallation.
2. Install the latest version.

## eggPlant Performance Injector Installation

These instructions cover installation of Windows Injectors.

### Installation (via .msi)

*On any machine that supports it, User Account Control (UAC) must be switched off before performing any of the following.*

On the Windows PC that will act as the Injector, log in as a user with local administrator privileges and execute either the `eggPlantPerformance<version>.msi` or the cut down `eggPlantPerformanceInjector<version>.msi` downloaded from the web or on CD.

You will need at least 2Mb of free disk space on the installation drive. (More space is required for running tests as the engines and various libraries are transmitted to the injector at runtime.) You must also have disk space available for recordings and test run data. This data can range from a few kilobytes to several gigabytes, depending on the test run options selected, the level of logging, the duration of your test runs, and the number of simulated virtual users.

Injectors do not require separate licensing. If you have installed using the `eggPlantPerformance<version>.msi`, you will not be able to run any of the licensable components such as Studio or Test Controller.

To optimize the machine settings, click on `forecastOptimalSettings_x86.reg` or `forecastOptimalSettings_x64.reg` found in the installation folder.

### Uninstallation (.msi)

The eggPlant Performance uninstall program does not remove your workspace or project data. If you do not want to save any of your workspaces, projects, or test results, you must manually delete them.

1. Close all eggPlant Performance component instances.
2. Uninstall via Windows Control Panel, Add or Remove Programs.

### Reinstallation

If you intend to update your installation with a new release, you must carry out the following steps.

1. Perform uninstallation.
2. Install the latest version.

### Contact TestPlant

Please email all problems or issues to [support@testplant.com](mailto:support@testplant.com).

For support issues relating to eggPlant Performance workspaces and projects, it can be useful if you email the project contents to TestPlant support. Select **File > Export Workspace/Projects** from within Studio to create a zip archive that can be emailed to [support@testplant.com](mailto:support@testplant.com).

**This page intentionally left blank to ensure new chapters start on right (odd number) pages.**